



**HAL**  
open science

# Architecture générique de fusion par approche Top-Down : application à la localisation d'un robot mobile

Maroua Ladhari

► **To cite this version:**

Maroua Ladhari. Architecture générique de fusion par approche Top-Down : application à la localisation d'un robot mobile. Automatique / Robotique. Université Clermont Auvergne [2017-2020], 2020. Français. NNT : 2020CLFAC052 . tel-03180407

**HAL Id: tel-03180407**

**<https://theses.hal.science/tel-03180407v1>**

Submitted on 25 Mar 2021

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

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

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

Thèse présentée par  
**Maroua LADHARI**

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

---

**Architecture générique de fusion par approche Top-Down :  
application à la localisation d'un robot mobile**

---

Préparée à l'Institut Pascal et  
Soutenue publiquement le 15 décembre 2020 devant le jury :

Mme. Samia BOUCHAFA	Professeure des universités	Université d'Evry	Rapporteur
Mme. Véronique CHERFAOUI	Professeure des universités	UTC Compiègne	Rapporteur
M. Michel DEVY	Directeur de recherche	LAAS-CNRS Toulouse	Examineur
M. Romuald AUFRÈRE	Maitre de Conférences	Université Clermont Auvergne	Examineur
M. Christophe DEBAIN	Chargé de Recherches	INRAE Clermont-Ferrand	Examineur
M. Roland CHAPUIS	Professeur des universités	Université Clermont Auvergne	Directeur de thèse

This work has been sponsored by the French government research programme Investissements d'avenir through the RobotEx Equipment of Excellence (ANR-10-EQPX-44) and the LabEx IMobS3 (ANR-10-LABX-16-01), by the European Union through the Programme Regional Competitiveness and Employment 2007-2013, European Regional Development Fund (ERDF – Auvergne region) and by «Région Auvergne»

# Remerciements

En tout premier lieu, je tiens à remercier Monsieur Michel Devy pour m'avoir fait l'honneur de présider mon jury de thèse et pour son rôle en tant qu'examineur. Je tiens à remercier également Mesdames Véronique Cherfaoui et Samia Bouchafa pour leur travail de rapporteuses et pour leur intérêt pour mon travail et leurs questions.

Je tiens à remercier tout particulièrement mon directeur de thèse Roland Chapuis qui m'a encadrée tout au long de ces quatre années. Il m'a guidée, orientée, aidée et soutenue durant l'intégralité de ma thèse. Je le remercie pour sa bonne humeur presque permanente.

Je remercie également mes encadrants Romuald Aufrère et Christophe Debain qui m'ont encadrée et épaulé tout au long de cette thèse. Sans leurs conseils et leur soutien quotidien, cette thèse n'aurait certainement pas pu voir le jour.

J'en profite également pour remercier tous les gens de l'Institut Pascal ayant permis mon intégration dans leurs locaux. Je remercie particulièrement Laurent Malaterre pour nos nombreuses discussions passionnées et passionnantes.

Je remercie tous ceux qui ont partagé mon bureau à un moment ou à un autre. Je ne vais pas citer les noms par peur d'en oublier.

Je remercie mes parents, ma soeur et mon frère pour leur soutien tout au long de cette thèse. Peu importe la distance qui nous sépare, ils étaient toujours aussi proches.

Finalement, je saisis cette occasion pour exprimer également mes vifs remerciements à toute personne ayant contribué de près ou de loin à la réalisation de cette thèse.

# Résumé

La problématique qui va être abordée dans cette thèse est la localisation d'un robot mobile. Ce dernier, équipé de capteurs bas-coût, cherche à exploiter le maximum d'informations possibles pour répondre à un objectif fixé au préalable. Un problème de fusion de données sera traité d'une manière à ce qu'à chaque situation, le robot saura quelle information utiliser pour se localiser d'une manière continue. Les données que nous allons traiter seront de différents types.

Dans nos travaux, deux propriétés de localisation sont désirées : la précision et la confiance. Pour pouvoir le contrôler, le robot doit connaître sa position d'une manière précise et intègre. En effet, la précision désigne le degré d'incertitude métrique lié à la position estimée. Elle est retournée par un filtre de fusion. Si en plus, le degré de certitude d'être dans cette zone d'incertitude est grand, la confiance dans l'estimation sera élevée et cette estimation sera donc considérée comme intègre. Ces deux propriétés sont généralement liées. C'est pourquoi, elles sont souvent représentées ensemble pour caractériser l'estimation retournée de la pose du robot. Dans ce travail nous rechercherons à optimiser simultanément ces deux propriétés.

Pour tirer profit des différentes techniques existantes pour une estimation optimale de la pose du robot, nous proposons une approche descendante basée sur l'exploitation d'une carte environnementale définie dans un référentiel absolu. Cette approche utilise une sélection a priori des meilleures mesures informatives parmi toutes les sources de mesure possibles. La sélection se fait selon un objectif donné (de précision et de confiance), l'état actuel du robot et l'apport informationnel des données.

Comme les données sont bruitées, imprécises et peuvent également être ambiguës et peu fiables, la prise en compte de ces limites est nécessaire afin de fournir une évaluation de la pose du robot aussi précise et fiable que possible. Pour cela, une focalisation spatiale et un réseau bayésien sont utilisés pour réduire les risques de mauvaises détections. Si malgré tout de mauvaises détections subsistent, elles seront gérées par un processus de retour qui réagit de manière efficace en fonction des objectifs souhaités.

Les principales contributions de ce travail sont d'une part la conception d'une architecture de localisation multi-sensorielle générique et modulaire de haut niveau avec un mode opératoire descendant. Nous avons utilisé la notion de triplet perceptif qui représente un ensemble amer, capteur, détecteur pour désigner chaque module perceptif. À chaque instant, une étape de prédiction et une autre de mise à jour sont exécutées. Pour l'étape de mise à jour, le système sélectionne le triplet le plus pertinent (d'un point de vue précision et confiance) selon un critère informationnel. L'objectif étant d'assurer une localisation intègre et précise, notre algorithme a été écrit de manière à ce que l'on puisse gérer les aspects ambiguës.

D'autre part, l'algorithme développé permet de localiser un robot dans une carte de l'environnement. Pour cela, une prise en compte des possibilités de mauvaises détections suite aux phénomènes d'ambiguïté a été considérée par le moyen d'un processus de retour en arrière. En effet, ce dernier permet d'une part de corriger une mauvaise détection et d'autre part d'améliorer l'estimation retournée de la pose pour répondre à un objectif souhaité.

**Mots clefs :** Localisation sur une carte, architecture multi-sensorielle générique de haut niveau, approche Top-Down, filtrage bayésien, Réseau bayésien.

# Abstract

The issue that will be addressed in this thesis is the localization of a mobile robot. Equipped with low-cost sensors, the robot aims to exploit the maximum possible amount of information to meet an objective set beforehand. A data fusion problem will be treated in a way that at each situation, the robot will select which information to use to locate itself in a continuous way. The data we will process will be of different types.

In our work, two properties of localization are desired : accuracy and confidence. In order to be controlled, the robot must know its position in a precise and reliable way. Indeed, accuracy refers to the degree of uncertainty related to the estimated position. It is returned by a fusion filter. If, in addition, the degree of certainty of being in this uncertainty zone is important, we will have a good confidence contribution and the estimate will be considered as reliable. These two properties are generally related. This is why they are often represented together to characterize the returned estimate of the robot position. In this work, our objective is to simultaneously optimize these two properties.

To take advantage of the different existing techniques for an optimal estimation of the robot position, we propose a top-down approach based on the exploitation of environmental map environmental map defined in an absolute reference frame. This approach uses an a priori selection of the best informative measurements among all possible measurement sources. The selection is made according to a given objective (of accuracy and confidence), the current robot state and the data informational contribution.

As the data is noisy, imprecise and may also be ambiguous and unreliable, the consideration of these limitations is necessary in order to provide the most accurate and reliable robot position estimation. For this, spatial focusing and a Bayesian network are used to reduce the risk of misdetection. However, in case of ambiguities, these misdetections may occur. A backwards process has been developed in order to react efficiently to these situations and thus achieve the set objectives.

The main contributions of this work are on one side the development of a high-level generic and modular multi sensory localization architecture with a top-down process. We used a concept of perceptual triplet which is the set of landmark, sensor and detector to designate each perceptual module. At each time, a prediction and an update steps are performed. For the update step, the system selects the most relevant triplet (in terms of accuracy and confidence) according to an informational criterion. In order to ensure an accurate and reliable localization, our algorithm has been written in such a way that ambiguity aspects can be managed.

On the other side, the developed algorithm allows to locate a robot in an environment map. For this purpose, the possibility of bad detections due to ambiguity phenomena has been taken into account in the backward process. Indeed, this process allows on the one hand to correct a bad detection and on the other hand to improve the returned position estimation to meet a desired objective.

**Keywords :** map localization, high-level generic multi-sensory architecture, Top-Down approach, fusion filter, Bayesian network.

# Table des matières

<b>I</b>	<b>Introduction</b>	<b>14</b>
I.1	Un bref historique sur le développement des robots mobiles	15
I.2	Les robots mobiles	18
I.3	Objectif et cadre de la thèse	18
I.4	Organisation du mémoire	19
I.5	Contributions	19
<b>II</b>	<b>État de l’art sur les approches de localisation des robots mobiles</b>	<b>21</b>
II.1	Introduction	22
II.2	Les modèles cinématiques des robots	22
II.2.1	Le modèle cinématique de type char	22
II.2.2	Le modèle cinématique de type tricycle	24
II.2.3	Le modèle cinématique de type Ackermann	24
II.3	La localisation relative	25
II.3.1	Localisation odométrique et inertielle	25
II.3.2	Odométrie optique	28
II.3.3	Localisation de type SLAM	30
II.3.4	Bilan	33
II.4	La localisation absolue	34
II.4.1	Types d’amers	34
II.4.2	Exploitation des amers	36
II.4.3	Méthodes de localisation à partir d’amers	36
II.4.4	Systèmes GNSS	38
II.5	Bilan : localisation relative et localisation absolue	39
II.6	La fusion d’informations pour la localisation	39
II.6.1	Définition	40
II.6.2	Classification des techniques de fusion d’informations	40
II.6.3	Stratégies de fusion ascendante et descendante	44
II.7	Conclusion	46
<b>III</b>	<b>Localisation sur une carte existante</b>	<b>47</b>
III.1	Introduction	48
III.2	Map-Matching et fusion multi-sensorielle	48
III.2.1	Approches géométriques	49
III.2.2	Approches topologiques	50
III.2.3	Approches avancées	51
III.3	Récapitulatif	56
III.4	Choix et conséquence	56
III.4.1	Choix de la carte	57
III.4.2	Choix de l’approche	58
III.5	Bilan	59
III.6	Conclusion	59

<b>IV</b>	<b>Description de l'architecture de fusion proposée</b>	<b>61</b>
IV.1	Introduction	62
IV.2	Algorithme	62
IV.2.1	Principes généraux	62
IV.2.2	Notion de triplet perceptif	63
IV.2.3	Objectifs	64
IV.2.4	Objectif de précision	64
IV.2.5	Objectif de confiance	66
IV.2.6	Critère unifiant de précision et confiance	68
IV.3	Critère de sélection d'un triplet	68
IV.3.1	Disponibilité du triplet	68
IV.3.2	Gain en précision escompté	69
IV.3.3	Gain en confiance	70
IV.3.4	Pertinence globale	71
IV.3.5	Bilan	73
IV.4	Principe de focalisation et détection	73
IV.5	Mise à jour	74
IV.6	Processus de retour en arrière	75
IV.6.1	Principe	75
IV.6.2	Algorithme récursif développé	75
IV.6.3	Exemple	77
IV.6.4	Bilan	78
IV.7	Conclusion	79
<b>V</b>	<b>Description des triplets perceptifs</b>	<b>80</b>
V.1	Introduction	81
V.2	Architecture multi-sensorielle générique	81
V.3	Triplet Lidar-Poteau	82
V.3.1	Notations	82
V.3.2	Analyse de pertinence d'un triplet lidar-poteau	83
V.3.3	Mise à jour d'un triplet lidar-poteau	89
V.4	Triplet Lidar-Mur	90
V.4.1	Notations	90
V.4.2	Analyse de pertinence d'un triplet lidar-mur	91
V.4.3	Mise à jour d'un triplet lidar-mur	99
V.5	Triplet UWB	100
V.5.1	Notations	100
V.5.2	Analyse de pertinence d'un triplet UWB	100
V.5.3	Mise à jour d'un triplet UWB	102
V.6	Triplet GPS	104
V.6.1	Notations	104
V.6.2	Analyse de pertinence d'un triplet GPS	105
V.6.3	Mise à jour d'un triplet GPS	109
V.7	Conclusion	112
<b>VI</b>	<b>Résultats architecture de fusion pour la localisation</b>	<b>113</b>
VI.1	Introduction	114
VI.2	Conditions de simulation pour les tests à 3 et 4 murs	114
VI.3	Résultats avec 3 murs	115
VI.4	Résultats avec 4 murs dont une ambiguïté	118
VI.5	Résultats avec 15 murs	124
VI.5.1	Cas "simple" avec 15 murs	124

VI.5.2 Tests quantitatifs et qualitatifs . . . . .	127
VI.6 Résultats sur le site PAVIN . . . . .	130
VI.6.1 Résultats bruts sur PAVIN . . . . .	130
VI.6.2 Tests quantitatifs et qualitatifs . . . . .	132
VI.7 Prise en compte d'un récepteur GNSS . . . . .	135
VI.7.1 Scénario 1 : grande incertitude initiale et GPS fiable . . . . .	135
VI.7.2 Scénario 2 : grande incertitude initiale et GPS peu fiable . . . . .	136
VI.7.3 Scénario 3 : incertitude initiale moyenne et GPS peu fiable . . . . .	136
VI.7.4 Scénario 4 : incertitude initiale bonne mais peu fiable et GPS fiable . . . . .	137
VI.7.5 Bilan sur l'ajout du triplet GNSS . . . . .	137
VI.8 Prise en compte de balises UWB . . . . .	138
VI.8.1 Situation canonique une balise et un mur avec mise à jour de type Kalman . . . . .	138
VI.8.2 Situation avec une balise et 15 murs et mise à jour de type Kalman . . . . .	138
VI.8.3 Illustration des effets de la linéarisation Kalman . . . . .	138
VI.8.4 Simulation UWB, lidar-mur, GPS avec pose initiale imprécise . . . . .	140
VI.8.5 Simulation UWB, lidar-mur, GPS avec pose initiale précise mais fausse . . . . .	140
VI.8.6 Bilan sur l'ajout du triplet UWB . . . . .	144
VI.9 Conclusion . . . . .	145
<b>VII Conclusion et perspectives</b>	<b>146</b>
VII.1 Conclusion . . . . .	147
VII.2 Perspectives . . . . .	148
VII.2.1 Autres types de triplets . . . . .	148
VII.2.2 Prise en compte du temps d'arrivée de la donnée . . . . .	148
VII.2.3 Correction et enrichissement de la carte . . . . .	148
VII.2.4 Triplet de construction de carte . . . . .	149
<b>VIII Annexes</b>	<b>150</b>
VIII.1 Introduction à la Théorie de l'Information . . . . .	150
VIII.1.1 Incertitude et information . . . . .	150
VIII.1.2 Entropie d'une source . . . . .	151
VIII.1.3 Entropie différentielle . . . . .	152
VIII.1.4 Information mutuelle . . . . .	153
VIII.2 Analyse géométrique sur les segments du plan . . . . .	153
VIII.2.1 Modélisation polaire d'un segment . . . . .	153
VIII.2.2 Distance d'un point à un segment . . . . .	154
VIII.2.3 Point d'intersection d'un segment avec un autre . . . . .	155
VIII.2.4 Intersection d'une droite avec un segment . . . . .	156
VIII.3 Produit de gaussiennes . . . . .	157

# Table des figures

FIGURE I.1:	Un modèle réduit de l' <i>automobile</i> de Léonard (source : (Goulu, 2008)) . . . . .	15
FIGURE I.2:	Mécanisme de la torpille Whitehead, publié en 1891 (source : (wikipedia, 2020b))	15
FIGURE I.3:	L'un des véhicules autonomes les plus documentés est le Stanford Cart, exploité comme véhicule de recherche pour les missions lunaires puis comme un suiveur de ligne blanche en 1966, jusqu'à son dernier rôle actif en 1980 comme véhicule de recherche autonome (source : (cyberneticzoo, 2016)). . . . .	16
FIGURE I.4:	Le MQ – 1 Predator est un drone de l'armée de l'air des <i>Etats – Unis</i> d'Amérique avec une longue autonomie. (source : (wikipedia, 2020a)) . . . . .	17
FIGURE I.5:	La Tesla Model S en pilote automatique (source : (kidioui, 2015)) . . . . .	17
FIGURE II.1:	Modèle cinématique de type char. Vecteur de configuration $[x_k, y_k, \theta_k]$ . . . . .	23
FIGURE II.2:	Modèle cinématique de type tricycle. Vecteur de configuration $[x_k, y_k, \theta_k, \delta_r]$ . . . . .	24
FIGURE II.3:	Modèle cinématique de type voiture . . . . .	25
FIGURE II.4:	Modèle cinématique de type bicyclette . . . . .	25
FIGURE II.5:	Illustration de la dérive d'un robot (source : (Guyonneau, 2013)) . . . . .	26
FIGURE II.6:	Résultats tirés de (Mennillo et al., 2017). À gauche, un exemple de reconstruction de trajectoires. Dans cette séquence, le véhicule d'acquisition suit un autre véhicule en mouvement. La trajectoire bleue est celle du véhicule d'acquisition, tandis que les trajectoires rouges correspondent aux points rouges du véhicule en mouvement visibles sur les deux vues de droite. À droite, Reconstruction de l'environnement rigide généré par le module visuel SLAM et sa trajectoire correspondante. . . . .	33
FIGURE II.7:	Principe de la localisation à partir d'amers en utilisant une approche Bottom-Up	36
FIGURE II.8:	Niveaux de fusion selon les niveaux d'abstraction . . . . .	43
FIGURE III.1:	Exemple d'un problème de sélection de route où la position estimée est très proche d'une route qui est différente de la route de circulation du véhicule. (source : (El Badaoui El Najjar, 2003)) . . . . .	49
FIGURE III.2:	La localisation de type point à point est basée sur un critère de calcul de la distance entre le point estimé et le nœud le plus proche. . . . .	50
FIGURE III.3:	Un exemple de carte topologique 2D tiré de (Dufourd, 2005). a) Représentation métrique de l'environnement associée à des lieux topologiques reliés entre eux par des relations d'adjacence. b) Représentation purement topologique correspondante. . . . .	50
FIGURE III.4:	Map-matching dans un croisement (source : (Gruyer et al., 2014)) . . . . .	52
FIGURE IV.1:	Algorithme de localisation . . . . .	63
FIGURE IV.2:	Apport de précision suivant une direction donnée pour deux matrices de covariance initiales différentes. L'ellipse en pointillé noire représente la covariance initiale a priori avant mise à jour. $A_1$ et $A_2$ sont deux amers de la carte de positions connues et les ellipses en traits pleins sont les covariances estimées après détection. . . . .	65

FIGURE IV.3:	Illustration de la différence d'apport de la précision entre deux triplets. En trait pointillé noir : la covariance initiale. En traits pleins : les covariances estimées après détection ; en bleu en utilisant le triplet $T_1$ et en rouge en utilisant le triplet $T_2$ . . . . .	66
FIGURE IV.4:	Illustration du principe de sélection selon l'apport de confiance. L'ellipse en pointillé noire représente l'incertitude autour de la pose estimée du robot. Les points A, B et C représentent des amers. Les autres ellipses en couleur présentent la zone de recherche pour chaque amer. . . . .	67
FIGURE IV.5:	Illustration du principe d'observabilité des triplets. . . . .	69
FIGURE IV.6:	Illustration de l'intégrité d'une estimation. La zone d'incertitude, représentée par une ellipse, est centrée autour de la position estimée. Dans la figure de gauche, la vraie pose du robot appartient à la zone d'incertitude estimée. L'estimation est donc intègre. Cependant celle de droite est non intègre puisque la zone d'incertitude n'englobe pas la pose réelle. On notera que cette notion implique un grand nombre d'estimation : une seule hors de la zone étant finalement possible bien que moins probable. . . . .	70
FIGURE IV.7:	Réseau bayésien utilisé dans le triplet lidar-mur prenant en compte le nœud $d_b$ qui sera utilisé pour le calcul de la probabilité $P(d_b)$ . . . . .	72
FIGURE IV.8:	Illustration du principe de focalisation spatiale. La position réelle du robot est $X_k$ et celle estimée est $\hat{X}_k$ avec une zone d'incertitude associée donnée par l'ellipse bleue. La zone de recherche ROI est donnée par l'ellipse en noir. Étant donnée l'incertitude de la pose estimée, cette zone ne sera pas centrée sur l'amer. . . . .	74
FIGURE IV.9:	Illustration du principe du retour en arrière . . . . .	77
FIGURE IV.10:	Algorithme de localisation avec un processus récursif . . . . .	78
FIGURE V.1:	Représentation du robot, du lidar et de la zone de recherche lidar . . . . .	83
FIGURE V.2:	Réseau bayésien utilisé dans ce triplet lidar-poteau . . . . .	86
FIGURE V.3:	Notation utilisées lorsque le capteur n'est pas centré sur le robot (phénomène de "bras de levier") . . . . .	87
FIGURE V.4:	Représentation du robot, du lidar, de la zone de recherche lidar et du segment AB représentant un mur . . . . .	90
FIGURE V.5:	Différents cas possibles d'observabilité d'un segment. La zone hachurée correspond à l'ouverture lidar. . . . .	91
FIGURE V.6:	calcul des angles $\theta_{di}$ et $\theta_{gi}$ . . . . .	92
FIGURE V.7:	ROI d'un segment . . . . .	94
FIGURE V.8:	Exemple de ROI avec les sigma points. La position du robot présente une incertitude $\sigma_x = \sigma_y = 1m$ et $\sigma_\theta = 20^\circ$ . . . . .	94
FIGURE V.9:	Critère de détection d'un segment : si $\rho_g$ et $\rho_d$ sont tous deux inférieurs à $\rho_{min}$ ou supérieurs à $\rho_{max}$ alors le segment ne sera pas détecté. . . . .	95
FIGURE V.10:	Exemples de fonctionnement de la procédure de détection de segments : les segments verts sont considérés non détectables et les segments rouges le sont. Le segment bleu est celui sur lequel est centrée la ROI . . . . .	96
FIGURE V.11:	Réseau bayésien utilisé dans ce triplet lidar-mur . . . . .	97
FIGURE V.12:	Paramètres de mesure d'un segment "mur" vu d'un lidar placé en $X_{sw}$ . . . . .	97
FIGURE V.13:	Illustration de la mise à jour par filtrage de Kalman dans le cas d'une mesure de type <i>range-only</i> . La balise, est située à une distance de 26m. L'estimation initiale est l'ellipse bleue et l'estimation de Kalman est l'ellipse verte. Celle-ci est correcte sur le point de linéarisation mais s'éloigne de la forme naturelle de l'anneau. . . . .	103

FIGURE V.14:	Illustration du calcul de l'incertitude recalculée avec l'approche proposée pour diverses configurations. L'ellipse rouge montre l'estimation réalisée. Les sigma points initiaux sont les croix bleues et points $Q_k$ (voir texte) sont les astérisques en rouge. . . . .	104
FIGURE V.15:	Détermination de $P_k^+$ pour un triplet GPS . . . . .	106
FIGURE V.16:	<b>Hypothèse 1</b> : l'estimation finale (cercle rouge) est superposée à l'estimation initiale (cercle noir). . . . .	110
FIGURE V.17:	<b>Hypothèse 2</b> : l'estimation finale (cercle rouge) est superposée à la mesure GPS (cercle bleu). . . . .	110
FIGURE V.18:	<b>Hypothèse 3</b> : l'estimation finale (cercle rouge) est donnée par un filtrage de Kalman en l'état précédent (cercle noir) et la mesure $z_k$ (cercle bleu) . . . . .	111
FIGURE VI.1:	Configuration initiale avec 3 murs . . . . .	115
FIGURE VI.2:	Configuration initiale avec 3 murs dans l'espace capteur . . . . .	116
FIGURE VI.3:	Configuration avec 3 murs à la profondeur 1 . . . . .	116
FIGURE VI.4:	Situation dans l'espace capteur après 1 détection . . . . .	116
FIGURE VI.5:	Configuration avec 3 murs à la profondeur 2 . . . . .	117
FIGURE VI.6:	Situation dans l'espace capteur après 2 détections . . . . .	117
FIGURE VI.7:	Configuration finale avec 3 murs . . . . .	118
FIGURE VI.8:	Configuration initiale avec 4 murs . . . . .	118
FIGURE VI.9:	Configuration initiale avec 4 murs dans l'espace capteur . . . . .	119
FIGURE VI.10:	Configuration avec 4 murs à la profondeur 1 avec ambiguïté . . . . .	120
FIGURE VI.11:	Situation dans l'espace capteur après 1 détection avec ambiguïté . . . . .	120
FIGURE VI.12:	Configuration avec 4 murs à la profondeur 2 avec ambiguïté . . . . .	120
FIGURE VI.13:	Situation dans l'espace capteur après 2 détections avec ambiguïté . . . . .	121
FIGURE VI.14:	Configuration avec 4 murs à la profondeur 0 après remise en cause . . . . .	121
FIGURE VI.15:	Nouvelle configuration avec 4 murs dans l'espace capteur après remise en cause . . . . .	122
FIGURE VI.16:	Configuration avec 4 murs à la profondeur 1 après remise en cause . . . . .	122
FIGURE VI.17:	Situation dans l'espace capteur après 1 détection après remise en cause . . . . .	123
FIGURE VI.18:	Configuration avec 4 murs à la profondeur 2 après remise en cause . . . . .	123
FIGURE VI.19:	Situation dans l'espace capteur après 2 détection après remise en cause . . . . .	123
FIGURE VI.20:	Configuration finale avec 4 murs . . . . .	124
FIGURE VI.21:	Configuration initiale avec 15 murs . . . . .	125
FIGURE VI.22:	Configuration initiale avec 15 murs dans l'espace capteur . . . . .	125
FIGURE VI.23:	Configuration avec 15 murs à la profondeur 1 . . . . .	126
FIGURE VI.24:	Configuration avec 15 murs dans l'espace capteur à la profondeur 1 . . . . .	126
FIGURE VI.25:	Configuration avec 15 murs à la profondeur 2 . . . . .	126
FIGURE VI.26:	Configuration avec 15 murs dans l'espace capteur à la profondeur 2 . . . . .	127
FIGURE VI.27:	Configuration avec 15 murs à la profondeur 4 . . . . .	127
FIGURE VI.28:	Configuration finale avec 15 murs . . . . .	128
FIGURE VI.29:	Résultat de la reconnaissance sur 1000 situations . . . . .	128
FIGURE VI.30:	Nombre d'itérations sur 1000 situations . . . . .	129
FIGURE VI.31:	Probabilité de confiance sur 996 situations . . . . .	129
FIGURE VI.32:	Distance de Mahalanobis sur 996 situations . . . . .	130
FIGURE VI.33:	Cartographie de PAVIN . . . . .	131
FIGURE VI.34:	Configuration initiale avec les murs de PAVIN (Situation 1) . . . . .	131
FIGURE VI.35:	Résultat final avec les murs de PAVIN (Situation 1) . . . . .	132
FIGURE VI.36:	Configuration initiale avec les murs de PAVIN (Situation 2) . . . . .	132
FIGURE VI.37:	Résultat final avec les murs de PAVIN (Situation 2) . . . . .	133
FIGURE VI.38:	Résultat de la reconnaissance sur 200 situations . . . . .	133
FIGURE VI.39:	Nombre d'itérations sur 200 situations . . . . .	134
FIGURE VI.40:	Probabilité de confiance sur 192 situations . . . . .	134

FIGURE VI.41:	Distance de Mahalanobis sur 192 situations . . . . .	135
FIGURE VI.42:	Exemple de situation où l’algorithme n’atteint pas les objectifs . . . . .	135
FIGURE VI.43:	Sur la gauche, en cyan, la position réelle du robot, en vert la position initiale estimée du robot et en rouge l’ellipse d’incertitude à un sigma autour de la position estimée. Le système choisit le triplet “mur 9” dès le début du processus. Seuls des triplets murs sont présents ici. La reconnaissance de la pose s’opèrera ensuite en 16 itérations de détections dans ce cas. La confiance finale sera de 0.9979 . . . . .	136
FIGURE VI.44:	Choix du triplet GPS de par son apport informationnel comparativement aux autres triplets. La zone d’incertitude autour de la mesure GPS est donnée par le cercle bleu. La reconnaissance de la pose s’opèrera ensuite en 7 itérations de détections ici. La confiance finale sera quasiment 100%. . . . .	136
FIGURE VI.45:	Choix du triplet GPS (malgré sa confiance de seulement 50%) de par son apport informationnel comparativement aux autres triplets. La reconnaissance de la pose s’opèrera ensuite en 7 itérations de détections ici. La confiance finale sera de quasiment 100% . . . . .	137
FIGURE VI.46:	Choix du triplet mur 9 dans cette configuration où l’incertitude initiale plus faible rend plus pertinente la sélection d’un triplet mur. La reconnaissance de la pose s’opèrera ensuite en 37 itérations de détections ici. La confiance finale sera de 100% . . . . .	137
FIGURE VI.47:	Choix du triplet GPS de par son apport informationnel. La fusion ne sera pas faite avec l’état précédent du fait que l’hypothèse initiale est de faible probabilité. La reconnaissance de la pose s’opèrera ensuite en 17 itérations de détections ici. La confiance finale sera de 100% . . . . .	138
FIGURE VI.48:	Simulation avec un mur et une balise (numéro 200) positionnée en (20, 10). La précision de mesure est supposée de 10cm. La pose initiale (carré vert) est (-10,0,0) et l’écart-type de son incertitude est (30m, 30m, 20°) : ellipse rouge. La position réelle est en (0,0,0) : carré cyan. Le système sélectionne tout d’abord le mur (figure -a-) puis la balise (figure -b-). La figure -c- montre la convergence finale. . . . .	139
FIGURE VI.49:	Simulation avec 15 murs et une balise. Le système sélectionne tout d’abord le mur 5 (figure -a-), la balise arrive en seconde place (figure -a-). Après la mise à jour (figure -b-), l’ordre de pertinence des triplets a changé, ce sera encore un mur qui sera choisi. La balise ne sera choisie qu’à la 9ème itération. . . . .	140
FIGURE VI.50:	Simulation avec 2 balises et une <i>mise à jour de type Kalman</i> . La pose réelle est donnée encore par le rectangle cyan en (10m, 15m, 0°). La pose estimée est en (0,0,0) : carré vert. Le première mesure (balise 200) contraint l’estimation initiale (figure -a-) à une ellipse fine et allongée qui ne contient pas la vraie pose (figure -b-). La seconde mesure est faite par la balise 201 qui met à jour cet l’état de manière précise mais malheureusement erronée (figure -c-). . . . .	141
FIGURE VI.51:	Simulation avec 2 balises et une <b>mise à jour de type UKF adaptée</b> . La pose réelle est donnée encore par le rectangle cyan en (10m, 15m, 0°). La pose estimée est en (0,0,0) : carré vert. Le première mesure (balise 200) contraint l’estimation initiale (figure -a-) à une ellipse qui n’englobe pas encore la vraie pose à un écart-type mais qui en est proche (figure -b-). La seconde mesure est faite par la balise 201 qui met à jour cet l’état de manière moins précise que précédemment mais qui est cette fois juste (figure -c-) . . . . .	142

FIGURE VI.52:	Simulation avec 2 balises, 3 mur et un récepteur GPS. La pose initiale est très imprécise (figure -a-). Le récepteur GPS est choisi logiquement en premier (de par son observabilité) et conduit à une mise à jour d'écart-type de l'ordre de 3m (celui du récepteur GPS). La balise 201 est ensuite sélectionnée pour conduire à l'estimation montrée en figure -c-. On voit dans cette figure que ce sera la balise 200 qui sera la plus pertinente ensuite. Logiquement, le mur 2 est choisi ensuite (figure -d-), ce qui conduira à atteindre le critère. . . . .	143
FIGURE VI.53:	Simulation avec 2 balises, 3 mur et un récepteur GPS. La pose initiale est assez précise <b>mais fausse</b> avec une confiance de 50% (figure -a-). Le récepteur GPS est choisi logiquement en premier et conduit à une mise à jour d'écart-type de l'ordre de 3m mais avec un choix de l'hypothèse de fusion 2 (voir V.6.2.2) qui conduit à ne pas réaliser la fusion avec l'état à priori mais à choisir la mesure GPS pour réinitialiser cet état (celui du récepteur GPS). Ensuite ce sera un triplet mur qui sera privilégié par l'algorithme (figure -c-) et la reconnaissance sera réalisée en 4 itérations sans retour arrière. . . . .	144
FIGURE VIII.1:	Évolution de l'entropie $H(A)$ en fonction de $P(a) = 1 - P(b)$ (source : (Liu et al., 2016)) . . . . .	152
FIGURE VIII.2:	Représentation $(\rho, \theta)$ d'un segment $AB$ . . . . .	154
FIGURE VIII.3:	Position du problème : on recherche la distance entre le point $P$ et le segment . . . . .	154
FIGURE VIII.4:	Point d'intersection de deux segments $[AB]$ et $[CD]$ . . . . .	155
FIGURE VIII.5:	Intersection d'une droite passant par l'origine $O$ avec un segment $AB$ . . . . .	156

# Liste des tableaux

II.1	Avantages et inconvénients de l'odométrie visuelle et de l'odométrie lidar inspirés de (Fre- mont, 2009) . . . . .	29
II.2	Méthodes de SLAM par filtrage (Innowtech, 2019) . . . . .	32
II.3	Table comparative des différentes technologies de localisation basées sur des balises actives tirée de (Zebra, 2014) . . . . .	36
II.4	Comparatif localisation relative - localisation absolue (source : (Chausse et al., 2006)) . . . . .	39
II.5	Table de fusion d'informations selon le modèle de classification Dasarathy . . . . .	42
II.6	Table comparative entre les deux approches Bottom-Up et Top-Down . . . . .	46
III.1	Table comparative entre les travaux dans la littérature et l'approche que nous proposons . . . . .	59
IV.1	Table des probabilités conditionnelles $P(obs O_k^-)$ . . . . .	71
IV.2	Table des probabilités conditionnelles $P(d_b obs)$ . . . . .	72

# Chapitre I

## Introduction

### Sommaire

---

I.1	Un bref historique sur le développement des robots mobiles . . . . .	15
I.2	Les robots mobiles . . . . .	18
I.3	Objectif et cadre de la thèse . . . . .	18
I.4	Organisation du mémoire . . . . .	19
I.5	Contributions . . . . .	19

---

## I.1 Un bref historique sur le développement des robots mobiles

Les robots mobiles ont une longue histoire. Dans cette partie, nous nous sommes intéressés à l'histoire des robots autonomes pour citer une série des grandes avancées dans les domaines terrestre, maritime et aérien qui ont contribué à fonder les bases des voitures du futur :

**Le chariot de Léonard de Vinci autour de 1493** : Léonard de Vinci esquisse un chariot qui pouvait se déplacer à l'aide de cames sans être poussé ou tiré (figure I.1). Le chariot est a priori conçu pour se déplacer le long d'une trajectoire prédéterminée. Ce dispositif, qui est un précurseur lointain de la voiture, est parfois considéré comme le premier robot au monde.



FIGURE I.1 – Un modèle réduit de l'*automobile* de Léonard (source : ([Goulu, 2008](#)))

**Whitehead Torpedo 1868** : alors que les armes de guerre sont apparues au milieu des années 1700, l'invention par Robert Whitehead d'une torpille (figure I.2) capable de se propulser sous l'eau a révolutionné le domaine des flottes navales du monde entier. La torpille Whitehead pouvait se déplacer à plusieurs centaines de mètres sous l'eau en maintenant une profondeur constante. Le guidage des torpilles allait ensuite évoluer de façon spectaculaire et mener à une large gamme d'armes, d'avions et d'autres dispositifs autonomes.

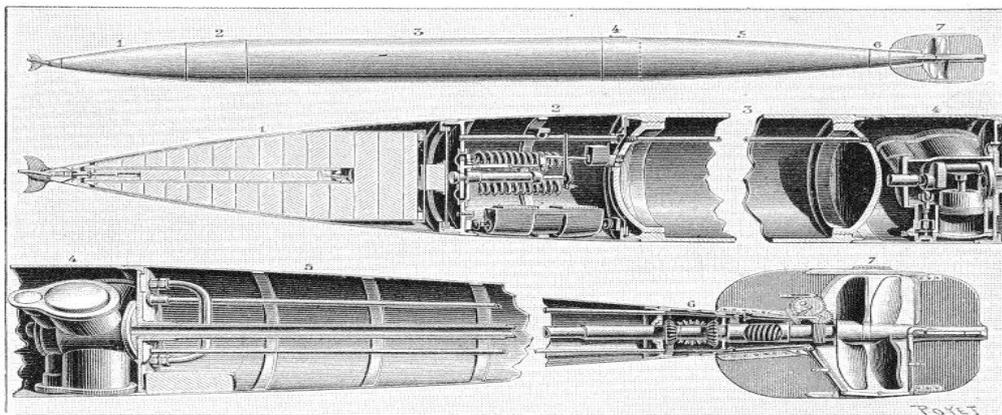


Fig. 1. — Torpille automobile Whitehead. — 1. Magasin. — 2. Chambre à secret. — 3. Réservoir d'air comprimé. — 4. Chambre des moteurs à air comprimé. — 5. Floqueur ou chambre de flottaison. — 6. Mécanisme de commande de rotation des hélices. — 7. Hélices et gouvernails.

FIGURE I.2 – Mécanisme de la torpille Whitehead, publié en 1891 (source : ([wikipedia, 2020b](#)))

**Pilote automatique de l'avion Mechanical-Mike 1933** : les trajets à grandes distances ont forcé le développement de systèmes de pilote automatique. Mechanical-Mike était un prototype de pilote automatique conçu par la société américaine d'équipements électroniques Sperry Gyroscope Co., et utilisé par Wiley Post lors d'un vol autour du monde de 13000 milles en 1933. Il intégrait deux gyroscopes et fonctionnait à l'air comprimé et à la pression hydraulique. C'était un pilote automatique "trois axes", c'est-à-dire qu'un gyroscope détectait les changements de tangage et de roulis (inclinaison) de l'avion, et l'autre gyroscope détectait les changements de cap (lacet). Les gyroscopes font aujourd'hui partie intégrante des technologies intégrées dans un robot mobile.

**Régulateur de vitesse Teetor 1945** : on raconte qu'un ingénieur, excédé par le mouvement de balancement qu'il ressentait en voiture avec son homme de loi au volant, qu'il a développé l'un des premiers systèmes de régulateur de vitesse pour adoucir la conduite, en utilisant un accélérateur mécanique qui pouvait régler la vitesse du véhicule. L'invention a été commercialisée en 1958.

**Stanford Cart 1961** : Rosen et al. au centre d'intelligence artificielle du SRI (Institut de recherche de Stanford) ont développé Shakey (figure I.3) le premier robot mobile au monde, à la fin des années 60. L'idée de base était de concevoir un rover lunaire télécommandé. Shakey était équipé de divers capteurs et utilisait des algorithmes pour la perception, la planification et la commande. Il n'est pas surprenant qu'il ait eu une influence considérable sur la robotique actuelle. Le but était d'étudier l'application de l'intelligence artificielle de l'époque sur des systèmes dans des environnements complexes. Le chariot a finalement été équipé de caméras et programmé pour détecter et suivre de manière autonome une ligne blanche continue au sol. Aujourd'hui, les technologies qui lui ont succédé et qui utilisent des caméras restent un élément essentiel des véhicules autonomes.

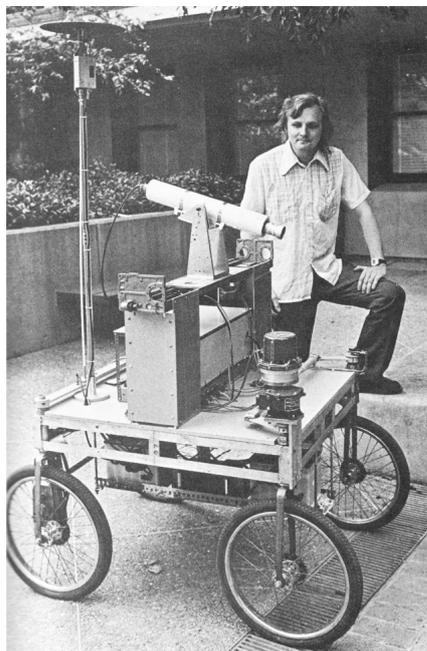


FIGURE I.3 – L'un des véhicules autonomes les plus documentés est le Stanford Cart, exploité comme véhicule de recherche pour les missions lunaires puis comme un suiveur de ligne blanche en 1966, jusqu'à son dernier rôle actif en 1980 comme véhicule de recherche autonome (source : ([cyberneticzoo, 2016](#))).

**Tsukuba Mechanical Engineering 1977** : aussi révolutionnaire qu'il ait été le Stanford Cart, la société japonaise Tsukuba Mechanical a produit un véhicule de tourisme autonome capable de reconnaître les marquages au sol d'une route tout en roulant à près de 30 *km/h*, grâce à deux caméras montées sur le véhicule.

**VaMoRs 1987** : une autre avancée importante dans la technologie autonome est donnée par le professeur allemand Ernst Dickmanns, qui a équipé une voiture d'une série de caméras et de 60 modules de microprocesseurs pour détecter des objets sur la route, devant et derrière le véhicule. La principale innovation de Dickmanns était la "vision dynamique", qui permettait au système d'imagerie, de filtrer les bruits et de se focaliser uniquement sur les objets pertinents. Aujourd'hui, ce type d'imagerie est crucial pour aider les véhicules autonomes à identifier les dangers potentiels et leur emplacement. Le système VaMoRs a permis de naviguer à des vitesses allant jusqu'à 60 miles par heure.

**General Atomics MQ 1 Predator 1995** : une autre catégorie de dispositifs autonomes a été conçue pour

voyager complètement seul. C'est le cas des drones, dont le plus remarquable est le Predator de General Atomics (figure I.4), un avion sans pilote qui, depuis 25 ans, survole les points chauds du globe pendant 14 heures d'affilée. Les drones ne sont pas seulement des véhicules militaires. En effet, le Predator est doté de technologies adaptées aux robots mobiles, notamment un radar qui peut voir à travers la fumée ou les nuages et des caméras thermiques qui permettent de voyager la nuit.



FIGURE I.4 – Le MQ – 1 Predator est un drone de l'armée de l'air des *Etats – Unis* d'Amérique avec une longue autonomie. (source : ([wikipedia](#), 2020a))

**Les défis de la DARPA 2004-2013** : le département américain de la recherche et de la défense DARPA, a financé une série de défis qui a fait avancer les technologies autonomes. En 2004, un concours a été organisé pour mettre au défi les véhicules de naviguer d'une manière autonome sur 150 miles de route dans le désert. Bien qu'aucune voiture n'ait terminé le parcours, les défis ultérieurs ont vu les capacités augmenter de façon spectaculaire. Le défi de 2007 s'est déroulé dans un environnement urbain de 60 miles de long, avec quatre voitures complétant le parcours dans le temps imparti de six heures.

**Pilote automatique Tesla 2015** : l'aspect le plus fascinant de la fonction "Pilote automatique" semi-autonome de Tesla, introduite fin 2015, qui permettait un contrôle mains libres pour la conduite sur autoroute, est qu'elle a été livrée sous la forme d'une mise à jour logicielle unique du modèle S (figure I.5). Équipée de fonctionnalités de conduite variées : un radar à l'avant, 12 capteurs ultrasons, une caméra orientée vers l'avant, un système électrique d'aide au freinage haute précision contrôlé numériquement, un GPS et des cartes numériques, cette fonction rend la voiture complètement autonome dans certaines situations.



FIGURE I.5 – La Tesla Model S en pilote automatique (source : ([kidioui](#), 2015))

## I.2 Les robots mobiles

D'une manière générale, les robots sont des machines intelligentes capables d'exécuter des tâches dans le monde réel sans contrôle humain explicite pendant de longues périodes (Nattharith, 2010).

Les robots mobiles sont des machines programmables et autocontrôlées, capables d'accomplir des tâches tout en naviguant dans un environnement avec ou sans obstacles, en fonction des informations détectées de son propre état interne et de l'environnement externe à l'aide de capteurs embarqués. En robotique mobile, un problème de navigation serait probablement défini comme la réponse à la problématique suivante : étant donné un certain espace métrique (une certaine région dans un environnement) et un ensemble de points fixes (l'emplacement de certains points de repère, connus ou inconnus au préalable), déterminer un chemin approprié d'un point de départ à un point d'arrivée en utilisant une représentation donnée de l'environnement. Pour trouver une trajectoire optimale ou approximativement optimale sans collision et pour réaliser un mouvement de sécurité, le robot peut suivre différents scénarios pour atteindre son point d'arrivée.

Le problème complet peut être résolu en réalisant un ensemble de sous-objectifs. À cette fin, Leonard et Durrant-Whyte (Leonard et al., 1992) divisent le problème de la navigation des robots mobiles en trois sous-problèmes : Où suis-je? (c'est-à-dire le problème de localisation), Où vais-je? (c'est-à-dire le problème de d'interprétation de la scène), et Comment y arriver? (c'est-à-dire le problème de planification du trajet). La navigation est une caractéristique importante des robots mobiles. La localisation dans un environnement est la toute première étape pour y parvenir.

La localisation est le processus qui consiste à déterminer la position et l'orientation d'un robot mobile par rapport à son environnement. C'est l'une des compétences les plus fondamentales requises par un robot autonome. La position du robot dépend du capteur utilisé. Les capteurs couramment utilisés par les robots mobiles comprennent les capteurs proprioceptifs : l'odomètre, le gyroscope ou le gyromètre<sup>1</sup>, le magnétomètre, les accéléromètres, etc. et les capteurs extéroceptifs : les caméras, le télémètre laser, l'infrarouge ultrasonique, le radar, etc.

## I.3 Objectif et cadre de la thèse

Le sujet s'inscrit dans le cadre du défi 1 du LabEx IMobS3. L'action IRMA-AVN a pour objectif de rationaliser les diverses approches exploitables en vue de la navigation fiable de robots mobiles dans un cadre de transport ou de service et dans des environnements complexes. Une des grandes problématiques de cette action est la localisation précise et intègre des robots. En effet, s'il semble simple de piloter un robot pour un humain, l'interprétation de la scène en vue de la localisation reste délicate pour un robot. Pour contourner cet écueil, il est possible d'exploiter des cartes de données facilitant cette interprétation. Ainsi, le problème réside dans l'exploitation de ressources multiples en plus de cette carte afin d'assurer une bonne localisation du robot sur celle-ci.

L'objectif de la thèse sera donc :

- la localisation absolue du robot dans des situations diverses grâce aux capteurs embarqués et en exploitant ces données de carte afin de permettre une localisation optimale tant du point de vue de la précision que de l'intégrité. Les informations d'incertitude et de fiabilité seront systématiquement associées à l'estimation de la pose.
- la conception d'une architecture générique, modulaire et multi-sensorielle qui exploite au mieux les données contextuelles de l'environnement ainsi que les capteurs embarqués pour gérer simultanément les aspects imprécision et ambiguïté qui peuvent dégrader le bon fonctionnement du système de localisation.

---

1. le gyromètre fournit la vitesse angulaire alors que le gyroscope fournit l'angle (position angulaire)

## I.4 Organisation du mémoire

Partant de l'existant et des différentes technologies utilisées, nous adressons dans un premier temps un état de l'art des méthodes utilisées dans le domaine de la localisation de robots mobiles. Les méthodes de localisation que nous présenterons seront réparties en deux grandes parties : la localisation relative et la localisation absolue. Le bilan de ces deux méthodes conclut sur la pertinence des techniques de fusion.

La deuxième partie du premier chapitre se focalise sur les différentes architectures et modèles de fusion existants pour combiner et intégrer des données contextuelles. Des approches dites ascendantes et descendantes, que nous allons détailler dans ce chapitre, sont utilisées à cet effet. La question de la localisation qui est posée dans cette thèse est abordée comme un problème de fusion de sources d'informations de natures variées qu'elles soient symboliques comme les informations cartographiques 2D ou numériques comme les informations délivrées par des capteurs (GPS, télémètre, caméra,...).

Le deuxième chapitre s'intéresse aux méthodes de localisation sur une carte. Une comparaison entre les différentes approches les plus utilisées pour la localisation d'un robot sur une carte sera exposée afin de révéler les avantages et les inconvénients de chacune d'elles. Un bilan comparatif sera enfin présenté.

Le troisième chapitre détaille notre approche de localisation dirigée par l'objectif fondée principalement sur les théories bayésiennes. Pour cela, nous utilisons la notion de triplet perceptif que nous avons repris dans nos travaux. Le problème de la localisation d'un robot sur une carte est traité dans ce chapitre comme l'estimation de la position d'un robot étant donnés les mesures bruitées fournies par les capteurs et leurs modèles d'observations. L'estimation assurée sera caractérisée par deux critères principaux : la précision de la localisation qui sera donnée par une zone d'incertitude autour de la pose estimée et la confiance qui représente la probabilité d'être réellement dans cette zone. Le premier critère s'appuie sur une estimation traditionnelle bayésienne (de type filtre de Kalman) tandis que le deuxième est donné par un réseau bayésien dans le but d'exploiter au mieux les avantages de chacune des sources d'informations, tout en essayant de pallier leurs limitations individuelles. Le réseau bayésien n'est pas seulement utilisé pour estimer la confiance mais aussi dans le but de gérer des situations ambiguës.

La deuxième partie de ce chapitre donne une présentation du processus de retour en arrière. En effet, la similitude des amers dans la carte de l'environnement ainsi que l'imprécision de la pose estimée sont deux facteurs qui mènent à une mauvaise association entre les données détectées et les amers de la carte ce qui conduit à une mauvaise estimation de la pose du robot. Le processus récursif mis en place sera en mesure de corriger les mauvaises associations et l'amélioration de la qualité de l'estimation.

Le quatrième chapitre est dédié à la présentation de l'implémentation technique des triplets perceptifs. En effet, l'idée générale de notre algorithme de fusion est d'utiliser d'une manière optimale et descendante l'ensemble des capteurs dont dispose le robot. Le système principal est une architecture modulaire haut-niveau disposant d'un ensemble de triplets perceptifs. Dans nos travaux, nous introduisons différents types de triplets qui seront détaillés dans ce chapitre de manière systématique afin de montrer la généralité de l'approche.

Le dernier chapitre expose une présentation des expérimentations et des résultats obtenus. L'objectif de ce chapitre est de montrer d'une part l'aspect générique et multisensoriel de notre approche et d'autre part l'intérêt du processus de retour et son avantage à assurer une estimation optimale de la pose initiale du robot dans un milieu présentant des ambiguïtés.

Enfin, une conclusion générale et des perspectives finalisent le document.

## I.5 Contributions

Les travaux menés dans cette thèse s'inscrivent dans le contexte de l'exploitation des données de la scène de navigation pour la localisation de robots autonomes dans un milieu urbain. L'objectif est de concevoir une

architecture générique, modulaire et multi-sensorielle qui exploite au mieux les données contextuelles de l'environnement ainsi que les capteurs embarqués pour assurer une estimation optimale de la position du robot. La localisation que nous cherchons à assurer doit gérer simultanément les aspects imprécision et ambiguïté qui peuvent dégrader le bon fonctionnement du système de localisation. L'originalité de ces travaux repose sur les principes de :

- ne pas fusionner à l'aveugle les données capteurs mais, contrairement aux approches ascendantes, nous cherchons à les sélectionner selon leur apport potentiel. En fait, les capteurs et les détecteurs ne remontent pas d'information en continu comme le ferait une approche exclusivement ascendante mais seulement si en amont ils ont été jugés utiles. Il en résulte qu'une seule donnée qui est jugée la plus pertinente est utilisée lors de l'estimation de la pose du robot. Du fait que cette donnée est sélectionnée, il s'en suit un processus de focalisation qui permet de réduire les potentialités de mauvaise détection ou d'outliers. Une première contribution consiste à utiliser un formalisme informationnel pour calculer l'apport de chaque donnée contextuelle pour atteindre l'objectif fixé.
- prendre en compte explicitement les possibilités de mauvaise détection ou d'ambiguïté des données via un processus de retour en arrière. En effet, ce dernier permet, d'une part, de corriger les mauvaises détections en cas d'une mauvaise association de données suite aux phénomènes d'ambiguïtés. D'autre part, il permet d'améliorer l'estimation retournée de la pose pour répondre à un objectif souhaité.
- dans le cas d'une mesure GPS, gérer la confiance à chaque instant permet de sélectionner une hypothèse de fusion pertinente.
- prendre en compte des mesures de type range-only avec des capteurs de type UWB. Pour cela, nous avons utilisé un filtre de type UKF afin d'éviter les convergences dues aux non linéarités caractéristiques de ce type de capteur.
- exploiter différents types de capteur dans le but de développer une approche de fusion modulaire et multi-sensorielle permettant, selon un objectif donné, de sélectionner les informations à fusionner en fonction de leur apport informationnel potentiel pour assurer une estimation précise et fiable de la pose du robot.

# Chapitre II

## État de l'art sur les approches de localisation des robots mobiles

### Sommaire

---

II.1	Introduction	22
II.2	Les modèles cinématiques des robots	22
II.2.1	Le modèle cinématique de type char	22
II.2.2	Le modèle cinématique de type tricycle	24
II.2.3	Le modèle cinématique de type Ackermann	24
II.3	La localisation relative	25
II.3.1	Localisation odométrique et inertielle	25
II.3.2	Odométrie optique	28
II.3.3	Localisation de type SLAM	30
II.3.4	Bilan	33
II.4	La localisation absolue	34
II.4.1	Types d'amers	34
II.4.2	Exploitation des amers	36
II.4.3	Méthodes de localisation à partir d'amers	36
II.4.4	Systèmes GNSS	38
II.5	Bilan : localisation relative et localisation absolue	39
II.6	La fusion d'informations pour la localisation	39
II.6.1	Définition	40
II.6.2	Classification des techniques de fusion d'informations	40
II.6.3	Stratégies de fusion ascendante et descendante	44
II.7	Conclusion	46

---

## II.1 Introduction

En général, lorsqu'il s'agit d'une application relevant des problèmes d'estimation dynamique de l'état d'un système, nous sommes amenés à faire appel à plusieurs capteurs, systèmes et informations souvent hétérogènes. Cela est vrai pour un système de localisation d'un robot mobile où l'objectif est d'estimer la pose de celui-ci. À travers ce chapitre, nous essayons de faire le point sur les approches de localisation des robots mobiles.

Pour commencer, nous présentons les grandes approches de localisation qui existent dans l'état de l'art à savoir les méthodes de localisation relative et absolue. Le bilan comparatif de ces méthodes montre l'intérêt de procéder à un système de fusion pour exploiter au mieux les différentes sources d'informations. Ces dernières sont très variées et fournissent des données différentes. Une étude sur les manières d'exploitation de ces données sera exposée dans la deuxième partie de ce chapitre.

Dans la littérature, différentes techniques sont utilisées pour localiser un robot mobile. Elles sont classées en deux catégories. Une première classe définit la localisation relative. Comme le nom l'indique, elle fournit une position par rapport à une autre, appelée position de référence ou simplement référentiel. Dans la deuxième classe se trouvent les systèmes de localisation absolue comme les systèmes GNSS (pour Global Navigation Satellites System) qui donnent la position du récepteur dans un référentiel considéré comme absolu comme la Terre. Avant d'aborder plus précisément ces deux grandes catégories d'approches pour localiser un robot mobile, nous nous intéressons d'abord aux modèles de mouvement des robots. Ensuite une étude des deux méthodes de localisation relative et absolue est présentée. Enfin, une étude sur la fusion d'informations pour la localisation est faite

## II.2 Les modèles cinématiques des robots

Ce travail de thèse se positionne autour de la problématique de la localisation d'un robot mobile. Pour cela, différentes informations relatives au comportement du robot sont nécessaires comme sa cinématique. À cet égard, nous allons nous intéresser dans cette partie aux modèles cinématiques d'un robot.

Un robot mobile est une machine capable de se mouvoir dans un environnement donné. Contrairement aux robots fixes manipulateurs de l'industrie, les robots mobiles sont capables de se déplacer. Dans notre cas, nous nous intéressons aux robots mobiles à roues évoluant dans un milieu localement plan. Plusieurs classes de robots existent selon le nombre et la position des roues (Bayle, 2008) (Guechi, 2010). Néanmoins, nous ne nous intéressons qu'aux modèles cinématiques des robots, les modèles dynamiques sont pertinents lorsqu'on évolue dans des environnements non plans, éventuellement glissants, et lorsque la prise en compte des vitesses et accélérations est nécessaire.

Avant de présenter les modèles cinématiques des robots à roues, nous rappelons les hypothèses suivantes pour vérifier la condition de roulement sans glissement :

- Le contact entre la roue et le sol est ponctuel
- Les roues sont indéformables, de rayon fixe  $r$
- Le sol est parfaitement plan

### II.2.1 Le modèle cinématique de type char

Ce modèle (figure II.1) autrement appelé modèle différentiel, est un modèle où le robot est équipé de deux roues fixes montées sur un même axe mais commandées indépendamment l'une de l'autre, de façon à pouvoir diriger le robot. En considérant un déplacement sans glissement du robot, ce modèle géométrique donne la

transformation qui permet de passer des vitesses angulaires des roues  $\omega_g$  et  $\omega_d$ , nous calculons la vitesse longitudinale, la vitesse latérale du robot selon l'axe  $(x, y)$  et la vitesse de rotation du robot qui seront données par l'équation II.1

- En premier lieu, en fonction des vitesses angulaires des roues  $\omega_g$  et  $\omega_d$ , nous calculons la vitesse longitudinale, la vitesse latérale du robot selon l'axe  $(x, y)$  et la vitesse de rotation du robot qui seront données par l'équation II.1

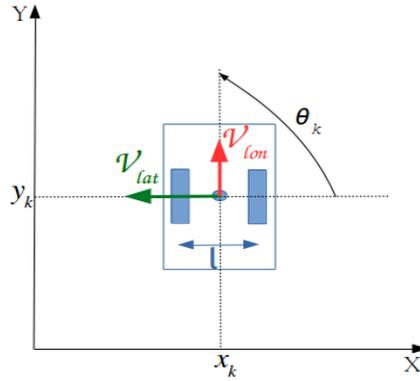


FIGURE II.1 – Modèle cinématique de type char. Vecteur de configuration  $[x_k, y_k, \theta_k]$

$$\begin{cases} v_{lon} &= \frac{r(\omega_d + \omega_g)}{2} \\ v_{lat} &= 0 \\ \dot{\theta} &= \frac{r(\omega_d - \omega_g)}{l} \end{cases} \quad (\text{II.1})$$

où  $r$  est le rayon des roues qui est supposé être identique pour les deux roues et  $l$  est la distance séparant les deux roues. La vitesse selon l'axe latéral du robot  $v_{lat}$  est considérée nulle du fait que le déplacement est supposé sans glissement.

- Les deux vitesses latérale et longitudinale du robot selon l'axe  $(x, y)$  sont données par l'équation II.2 :

$$\begin{cases} v_{lon} &= \dot{x} \cos(\theta) + \dot{y} \sin(\theta) \\ v_{lat} &= -\dot{x} \sin(\theta) + \dot{y} \cos(\theta) \end{cases} \quad (\text{II.2})$$

- Finalement, les déplacements élémentaires du robot sont obtenus en intégrant dans le temps les vitesses longitudinale, latérale et de rotation et en tenant compte des équations II.1 et II.2. Ils sont donnés par l'équation II.3 :

$$\begin{cases} x_k &= x_{k-1} + d_s \cos(\theta_{k-1}) \\ y_k &= y_{k-1} + d_s \sin(\theta_{k-1}) \\ \theta_k &= \theta_{k-1} + d_\theta \end{cases} \quad (\text{II.3})$$

avec :

$$— d_s = \frac{r(\omega_d + \omega_g)}{2} d_k \text{ et } d_\theta = \frac{r(\omega_d - \omega_g)}{l} d_k$$

—  $d_k$  est le temps écoulé entre les instants  $k - 1$  et  $k$ .

## II.2.2 Le modèle cinématique de type tricycle

Ce modèle, figure II.2, caractérise un robot constitué de deux roues arrières fixes et montées sur le même axe comme dans le cas du chariot mais qui ne sont plus motorisées et d'une roue centrée orientable placée sur l'axe longitudinal du robot. La propulsion et la direction du tricycle sont assurées exclusivement par la roue avant sous les hypothèses de roulement pur sans glissement. L'état de configuration du robot est alors composé de  $[x_k, y_k, \theta_k, \delta_r]$  avec  $\delta_r$  l'angle de braquage de la roue avant. Une formulation du modèle cinématique est donnée par l'équation II.4.

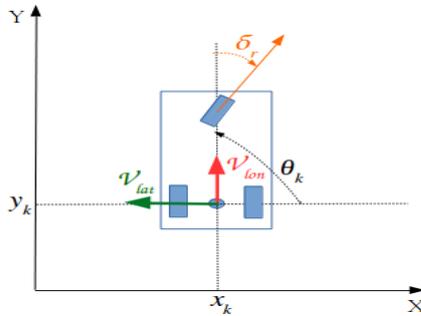


FIGURE II.2 – Modèle cinématique de type tricycle. Vecteur de configuration  $[x_k, y_k, \theta_k, \delta_r]$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\delta}_r \end{pmatrix} = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ \tan(\frac{\delta_r}{l}) \\ 0 \end{pmatrix} v_{lon} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} v_r \quad (\text{II.4})$$

où  $\dot{\delta}_r$  est la vitesse de braquage de la roue directrice,  $v_{lon}$  la vitesse longitudinale des roues du même axe et  $v_r$  est la vitesse de l'angle de direction.

En intégrant dans le temps  $\dot{x}$ ,  $\dot{y}$  et  $\dot{\theta}$  et en admettant que l'angle de braquage est connu à tout instant (l'estimation de  $\dot{\delta}_r$  est par conséquent inutile), le déplacement du robot entre les instants  $k$  et  $k - 1$  sera donné par l'équation II.5.

$$\begin{cases} x_k = x_{k-1} + v_{lon} \Delta_t \cos(\theta_{k-1}) \\ y_k = y_{k-1} + v_{lon} \Delta_t \sin(\theta_{k-1}) \\ \theta_k = \theta_{k-1} + \tan(\frac{\delta_r}{l}) \end{cases} \quad (\text{II.5})$$

## II.2.3 Le modèle cinématique de type Ackermann

Aussi appelé modèle de type bicyclette (figure II.4), c'est un modèle à deux roues. C'est une simplification du modèle classique de véhicule routier (à quatre roues) de la figure II.3. Ceci implique que les deux roues de chaque essieu peuvent être confondues au centre de l'essieu pour créer une roue équivalente, comme représenté par la figure II.4. En fait, les deux roues du modèle bicyclette sont placées sur l'axe médian du robot.

L'une de ces deux roues est orientable d'un angle  $\delta$  de telle sorte que le CIR (pour Centre Instantané de Rotation) reste inchangé. Ce dernier est le point d'intersection des lignes joignant les axes de rotation des roues. C'est un point de vitesse nulle autour duquel le robot tourne de façon instantanée. Pour passer du modèle voiture (figure II.3) au modèle bicyclette (figure II.4) l'angle de braquage  $\delta$  donné par la roue avant peut être fourni selon deux manières :

- si les deux roues de l'essieu avant ont les mêmes angles de braquage, alors dans ce cas :

$$\delta = \delta_i = \delta_e$$

- dans le cas contraire (les deux roues avant n'ont pas les mêmes angles) et en considérant la condition de l'épure de Jeantaud donnée par l'équation II.6 :

$$\delta = \frac{\delta_i + \delta_e}{2}$$

$$\cot \delta_e - \cot \delta_i = \frac{L}{D} \quad (\text{II.6})$$

avec  $\delta_e \leq \delta_i$ ,  $\delta_e$  l'angle de braquage de la roue gauche,  $\delta_i$  l'angle de braquage de la roue droite,  $L$  la distance séparant les deux roues de l'essieu arrière et  $D$  la distance séparant les deux essieux avant et arrière.

À un instant  $t$  donné et avec un  $\delta$  faible, les contraintes cinématiques du modèle Ackermann s'écrivent selon l'équation II.7.

$$\begin{cases} x_{k+1} = x_k + d_s \cos(\theta_k + \frac{d\theta}{2}) \\ y_{k+1} = y_k + d_s \sin(\theta_k + \frac{d\theta}{2}) \\ \theta_{k+1} = \theta_k + \frac{d_s}{L} \sin \delta \end{cases} \quad (\text{II.7})$$

avec :  $d\theta = \frac{d_s}{L} \sin \delta$  et  $d_s$  le déplacement élémentaire entre les instants  $k$  et  $k + 1$

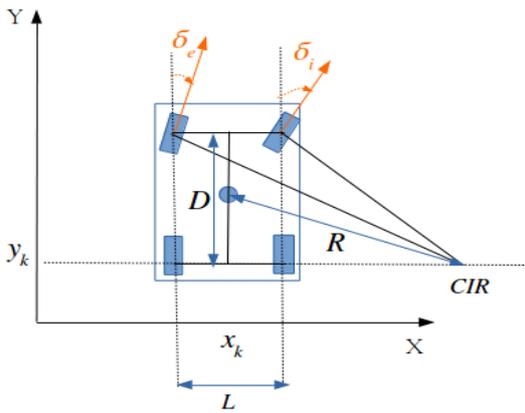


FIGURE II.3 – Modèle cinématique de type voiture

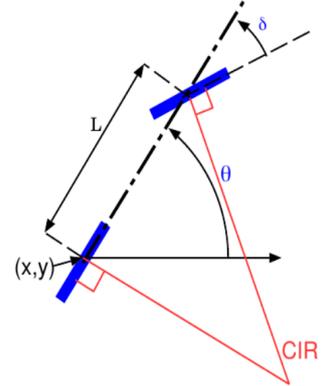


FIGURE II.4 – Modèle cinématique de type bicyclette

## II.3 La localisation relative

Ce type de localisation est fortement basé sur l'utilisation des capteurs proprioceptifs qui fournissent des mesures propres au robot. Parmi les capteurs utilisés dans ce type de localisation nous trouvons l'odomètre, la centrale inertielle, les gyromètres et les accéléromètres.

### II.3.1 Localisation odométrique et inertielle

La localisation par odométrie est très répandue en robotique mobile de par sa facilité de mise en œuvre ainsi que son faible coût (Chong and Kleeman, 1997) (Agarwal et al., 2019) (Dao et al., 2003). L'odomètre se caractérise par une cadence d'acquisition très élevée ce qui le rend de haute disponibilité. Un odomètre est équipé d'encodeurs. Le nombre des encodeurs utilisés ainsi que leur résolution déterminent la qualité de la mesure retournée. Cette technique permet d'estimer la position d'un robot en mouvement. Elle permet de calculer le déplacement du robot en calculant le nombre de tours effectués par chaque roue pendant une durée

déterminée. Si, en plus, nous considérons un temps d'échantillonnage faible, nous pourrions déduire les vitesses linéaires et angulaires (Bonnifait, 1997).

Le mouvement global du robot est reconstitué à partir de la mesure individuelle des déplacements des roues. À chaque instant, la position courante du robot est calculée en partant d'une position initiale et en intégrant les déplacements effectués. Pour calculer la position du robot à partir des mesures odométriques, il est nécessaire de disposer d'un modèle décrivant le déplacement du robot. Pour cela nous pouvons nous référer à la partie §II.2. Selon les modèles cinématiques montrés dans cette partie, nous constatons que la pose du robot est obtenue à partir de la vitesse. Et comme nous intégrons une vitesse pour en déduire une position, nous intégrons aussi l'erreur, ce qui veut dire que l'erreur s'accumule en fonction du temps. La figure II.5 illustre le phénomène de dérive de l'odométrie d'un robot en déplacement. Plus le robot évolue dans le temps, plus les erreurs liées à l'odométrie augmentent.

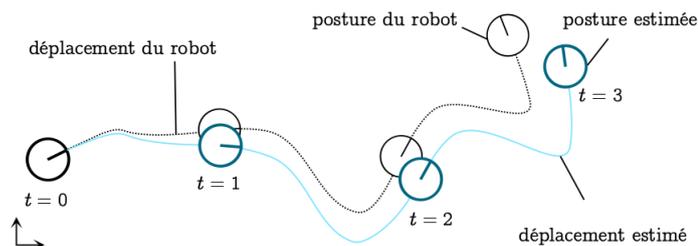


FIGURE II.5 – Illustration de la dérive d'un robot (source : (Guyonneau, 2013))

L'accumulation des erreurs suite à des hypothèses faites et des erreurs de mesure (mouvement sans glissement, mauvaise connaissance du diamètre des roues qui varie au cours du temps, résolution des codeurs, fréquence d'échantillonnage inexacte, etc.) rend l'utilisation de l'odométrie problématique. Pour plus de détails nous renvoyons le lecteur à (Rives and Devy, 2001) et (Laneurit, 2006) où un modèle de l'erreur sur la mesure d'odométrie est fourni.

En résumé, nous trouvons les propriétés suivantes pour les systèmes odométriques (Abuhadrous, 2005) :

— avantages :

- Le prix : peu coûteux. Leur exploitation pour la navigation ne représente pas de charge ou de coût supplémentaires
- Fréquence d'acquisition élevée
- Simple : l'odométrie est basée sur des équations simples, donc informatiquement parlant les calculs sont peu coûteux

— inconvénients :

- Une imprécision croissante à cause des dérives dues aux patinages des roues, aux glissements et aux bruits capteurs.
- Information en 2D : l'odométrie retourne la distance parcourue par le robot sur le sol, si la route est inclinée, la distance mesurée sera erronée.
- Pas d'information absolue : nous avons besoin d'un capteur externe de type absolu pour avoir un positionnement absolu.

Une autre solution proprioceptive est donnée par les capteurs inertiels. L'utilisation de ces systèmes qui sont basés sur des gyromètres et des accéléromètres, comme la centrale inertielle (IMU pour Inertial Measurement Unit) et le système de navigation inertielle (INS pour Inertial Navigation System) fournit des mesures sur la position et l'orientation du robot à partir des vitesses angulaires et les accélérations linéaires. Cette utilisation est précise à court terme mais elle reste très sensible à la gravité et manque de précision à long terme. C'est le cas des capteurs bas coût. Toutefois il existe certaines centrales qui sont très précises (les IMU Landins de chez XBlue par exemple) qui sont équipées de gyrolaser mais ces capteurs sont très chers et encombrants, et réservés le plus souvent au domaines militaire, aéronautique ou spatial.

Pour réaliser le positionnement dans un système de référence, il est alors nécessaire de connaître l'orientation spatiale des composantes accélérométriques. Ainsi les systèmes INS sont composés de trois accéléromètres orientés à l'aide de trois gyromètres.

Les IMU fournissent une résolution plus précise du positionnement et des fréquences de mise à jour plus rapides, qui sont des fonctionnalités requises par de nombreuses applications embarquées. En prime, un traitement important avec un logiciel est toujours nécessaire pour filtrer les données des capteurs IMU, pour fusionner les données provenant des divers types de capteurs à l'intérieur de l'IMU et pour calculer une position à partir de ces données fusionnées à l'aide d'une double intégration. Le positionnement est alors soumis à de fortes dérives liées à l'intégration des bruits de mesures et à un mauvais calibrage du facteur d'échelle (ou encore la non linéarité du facteur d'échelle). Dans (Abuhadrous, 2005) et (Laneurit, 2006), une modélisation des erreurs des capteurs inertiels est présentée ainsi que l'influence de ces erreurs sur la localisation.

De même, nous trouvons les propriétés suivantes pour un système inertiel (Cappelle, 2008) :

— avantages :

- Bonne précision à court terme
- Bonne disponibilité : système autonome insensible aux conditions de roulement
- Fréquence d'échantillonnage élevée : en général entre 100 à 150 Hz
- Solution complète : information de position, vitesse, accélération et orientation
- Aucun lien mécanique avec les parties mobiles du robot

— inconvénients :

- Une imprécision croissante à cause des effets de biais et de la marche aléatoire des accéléromètres et des gyromètres.
- Un système inertiel est sensible à la gravité.
- Pas d'information absolue : un système inertiel nécessite un capteur extéroceptif pour être initialisé.

Afin d'exploiter les avantages de chaque technique, certains travaux couplent ces deux types de capteur. En effet, l'odométrie est généralement couplée à d'autres capteurs (par exemple un gyromètre, une centrale inertielle) (Cho et al., 2011) (Agarwal et al., 2019) (Youssef and El-Sheimy, 2019) puisqu'elle ne permet de donner que les positions relatives du robot et que la position initiale doit être donnée par d'autres moyens. Les données utilisées pour prédire la pose du robot sont les déplacements relatifs des roues donnés par un odomètre ainsi que l'angle de braquage donné par un gyromètre. Dans (Ibraheem, 2010) et (Aynaud et al., 2014b) un modèle qui fait la fusion des données odomètre et gyromètre, en considérant un modèle de dérive sur les mesures, a été utilisé pour une estimation plus fiable de la position du robot.

Après avoir présenté les deux principales techniques de localisation relative que sont l'odométrie et les systèmes inertiels, nous citons dans la partie §II.3.2 d'autres méthodes d'odométrie telles que l'odométrie

optique. Cette dernière a pour objectif d'estimer l'ego mouvement du robot en utilisant des données images obtenues à partir d'une ou plusieurs caméras embarquées et/ou de données télémétriques lidar.

## II.3.2 Odométrie optique

### II.3.2.1 Odométrie visuelle

Toujours dans le cadre de la localisation relative (le repère de référence est lié à la position initiale du robot), nous citons la technique de localisation par odométrie visuelle qui considère un ensemble d'images prises à des instants successifs et déduit la pose du robot par rapport au déplacement de la caméra. La technique se base sur la mise en correspondance de pixels des images acquises et nécessite que les images acquises successivement aient un recouvrement. Le principe est très proche de celui de l'odométrie. En effet, la position globale est estimée par composition d'estimations de déplacements élémentaires, dont les erreurs se cumulent au cours du temps. En revanche, contrairement à l'odométrie proprioceptive présentée dans §II.3.1, la technique d'odométrie visuelle est considérée comme une technique extéroceptive qui utilise des mesures de l'environnement extérieur du robot présentant ainsi des propriétés complémentaires aux techniques odométriques et inertielles standards. Le principe est illustré dans (Datondji et al., 2015) où l'auteur présente une méthode d'odométrie visuelle pour la navigation autonome à partir d'un capteur de vision omnidirectionnelle. L'objectif consiste à utiliser l'information de vision pour estimer le déplacement entre deux images. Ce déplacement est réalisé à partir d'appariements de points d'intérêt et de l'estimation de la géométrie épipolaire (estimation de la matrice essentielle en particulier).

En effet, deux axes de recherche ont été menés au fil du temps : l'un monoculaire (Gonzalez et al., 2011) (He et al., 2020) (Sabry et al., 2019) et l'autre stéréo (Zhang et al., 2014) (Abdelghani et al., 2018). Dans (Scaramuzza and Fraundorfer, 2011), les auteurs présentent dans la première partie de leur publication un état de l'art sur la VO (pour Visual Odometry), ensuite des améliorations ont été ajoutées dans (Fraundorfer and Scaramuzza, 2012) pour une estimation plus robuste. Des travaux similaires peuvent être consultés dans (Larnaout, 2014) (Martins, 2017). Deux principaux phénomènes de dérive de l'odométrie visuelle existent. Le premier fait partie des limites de l'odométrie monoculaire où l'inconvénient est lié à l'estimation du facteur d'échelle. En effet, la reconstruction est réalisée à un facteur d'échelle près qui est théoriquement constant et qui induit une reconstruction incohérente de la carte et limite la taille de l'environnement reconstruit. Le deuxième inconvénient de l'odométrie visuelle est lié à l'association temporelle des points d'une image à une autre. Les mouvements brusques, les occultations ainsi que les mauvaises conditions météorologiques peuvent engendrer des incohérences lors du suivi des points et l'estimation du déplacement (Le Barz, 2015).

Évidemment la dérive peut être réduite par l'utilisation d'autres capteurs. Afin de pallier les limites de l'odométrie visuelle, les travaux présentés dans (Melbouci, 2017), se sont centrés sur l'intégration d'informations supplémentaires afin de mieux contraindre la trajectoire de la caméra et la reconstruction 3D. Les informations intégrées sont l'information de profondeur fournie par un capteur 3D (e.g. Microsoft Kinect) et des informations géométriques sur la structure de la scène. Par ailleurs, une méthode de visuo-inertiel (VINS) (fusion de l'odométrie visuelle et INS) présentée dans (Nam and Gon-Woo, 2020) consiste à éliminer les données aberrantes dans un contexte multi-hypothèses. D'autres méthodes existent pour résoudre le problème de la dérive de l'odométrie visuelle par une optimisation sur les dernières poses estimées des caméras avec un algorithme d'ajustement de faisceaux (en anglais Bundle Adjustment) (Mouragnon et al., 2006) (Gong et al., 2015). En outre, la technique de fermeture de boucle (en anglais Loop-Closure) permet de corriger les défauts de la carte créée et de la maintenir spatialement cohérente (Birem, 2015) (Caramazana et al., 2016).

### II.3.2.2 Odométrie lidar

D'autres techniques basées sur l'utilisation du capteur lidar au lieu d'une caméra existent dans la littérature. Ces techniques d'odométrie par lidar sont connues en anglais sous le nom de visual-lidar odometry ou de lidar odometry. Par rapport aux systèmes de vision, les mesures de profondeur sont directement disponibles sans passer par les étapes d'extraction de points caractéristiques et de reconstruction 3D à partir des mesures images.

Ces techniques exploitent l'ensemble de points des scans lidar obtenus à deux instants d'acquisition successifs. En utilisant souvent un algorithme d'association de données ou de recalage de type ICP (pour Iterative Closest Point) (Besl and McKay, 1992) et ses variantes (Rusinkiewicz and Levoy, 2001), ce type d'algorithme prend comme entrée deux ensembles de primitives (points, segments, surfaces) 2D ou 3D, et détermine la transformation optimale de rotation et/ou de translation qui peut exister entre les deux (Fremont, 2009).

Néanmoins, contrairement aux méthodes existantes qui utilisent la correspondance de caractéristiques ou la recherche du plus proche voisin pour calculer le coût entre deux nuages de points, (Kim and Choi, 2017) propose une autre méthode d'odométrie lidar. L'algorithme n'utilise qu'un lidar multi nappes rotatif sans aucun autre capteur. La méthode proposée calcule le coût directement en utilisant la projection cylindrique au lieu de la correspondance des caractéristiques et de la recherche du plus proche voisin. La technique proposée permet aussi de réduire les effets des objets en mouvement et des occultations. Les travaux dans (Li and Wang, 2020), présentent la méthode Deep Matching lidar Odometry (DMLO), un nouveau cadre basé sur l'apprentissage qui permet d'appliquer la méthode d'appariement des caractéristiques à la tâche d'odométrie lidar. En outre et contrairement à de nombreuses méthodes récentes basées sur l'apprentissage, DMLO utilise explicitement des contraintes géométriques. Plus précisément, DMLO décompose l'estimation de la pose à 6 degrés de liberté en deux parties : un réseau d'appariement basé sur l'apprentissage qui fournit des correspondances précises entre deux balayages et une estimation de transformation rigide avec une solution proche par décomposition de valeur singulière (SVD pour Singular Value Decomposition). D'autres travaux d'odométrie lidar peuvent être consultés dans (Sun et al., 2018) (Thomas et al., 2019) (Li et al., 2019) (Cho et al., 2019) (Liang et al., 2020).

### II.3.2.3 Récapitulatif

Au vu des caractéristiques de chaque technique, il apparaît une forte complémentarité entre l'odométrie visuelle et l'odométrie lidar. Il est par conséquent judicieux de les utiliser conjointement afin d'améliorer les performances. Nous rappelons dans la table II.1 les éléments comparatifs concernant ces deux techniques. Dans (Zhang and Singh, 2015), un cadre général permettant de combiner l'odométrie visuelle et l'odométrie lidar est présenté.

Pour résumer, l'odométrie optique se base sur l'utilisation de deux types de capteurs que sont les caméras et les lidars. En effet, ces deux capteurs présentent une certaine forme de polyvalence facilitant ainsi la perception de l'environnement et pouvant être exploités pour des fins de localisation du robot. Dans la partie suivante, nous présentons une autre technique de localisation : la localisation par la technique de SLAM.

	<b>Odométrie visuelle</b>	<b>Odométrie lidar</b>
<b>Résolution</b>	résolution spatiale importante	résolution spatiale limitée
<b>Portée de détection</b>	limitée	importante
<b>Sensibilité du capteur</b>	fortement dépendant de la luminosité ambiante et des conditions météo	sensible aux conditions météorologiques défavorables (neige, pluie, brouillard)
<b>Importance de la texture</b>	cruciale pour avoir de bons résultats	sensible aux spécularités ou objets très noirs
<b>Interprétation haut niveau</b>	importante	limitée
<b>Traitement 3D</b>	extraction et triangulation	techniques d'analyse de données (segmentation, reconnaissance de formes, classification)

TABLE II.1 – Avantages et inconvénients de l'odométrie visuelle et de l'odométrie lidar inspirés de (Fremont, 2009)

### II.3.3 Localisation de type SLAM

Le terme SLAM, traduit par cartographie et localisation simultanées, regroupe différentes techniques, qui ont toutes le même objectif : naviguer à partir d'une position inconnue dans un environnement inconnu en construisant la structure de l'environnement. La carte construite est utilisée simultanément pour s'y localiser et pour la mettre à jour suite à chaque nouvelle observation réalisée. Cette technique, initiée dès le début des années 90, regroupe des méthodes de localisation basées sur des cartes construites de manière incrémentale. Les lecteurs intéressés pourront d'ailleurs se tourner vers (Durrant-Whyte and Bailey, 2006) et (Bailey and Durrant-Whyte, 2006) pour un examen historique approfondi des 20 premières années du problème SLAM (entre 1986 – 2004) et vers (Khairuddin et al., 2015) et (Agunbiade and Zuva, 2018) pour les dernières avancées dans la période suivante entre (2004 – 2018).

De nombreux capteurs (caméra, télémètre laser, radar, odomètre, etc.) peuvent être employés seuls (Lu et al., 2020) (visual monocular SLAM) (Słowak and Kaniewski, 2020) (SLAM LiDAR) (Holder et al., 2019) (SLAM Radar) ou bien combinés (Chen et al., 2020) (Chan et al., 2018) (fusion du SLAM Lidar et du SLAM visuel), (Gong et al., 2019) (fusion du SLAM visuel et GNSS).

Étant donnée la richesse de ce sujet dans la littérature, cette partie sera divisée en deux. Nous commencerons par une analyse sur des solutions SLAM basées sur des techniques de filtrage (cf. II.3.3.1) avant d'introduire les méthodes de SLAM par minimisation globale (cf. II.3.3.2). Cette division est due au fait que ces deux approches s'opposent dans le principe : alors que les filtres dans la première approche s'orientent vers la minimisation des erreurs quadratiques, la minimisation globale est plutôt proche à un estimateur de maximum de vraisemblance.

#### II.3.3.1 Approches de SLAM par filtrage

Les filtres ont été initialement très employés dans le cadre du SLAM. Nous pouvons trouver des méthodes de SLAM par filtre de Kalman et ses dérivés (EKF et UKF) ou de FastSLAM utilisant les filtres particulaires. Ces filtres sont utilisés dans un cadre probabiliste de SLAM utilisant des cartes construites d'une manière incrémentale vu qu'ils offrent les meilleurs résultats et permettent de minimiser les incertitudes de la pose estimée du robot ainsi que celles de la carte.

Les premiers travaux dans le domaine du SLAM consistent à utiliser des cartes construites d'une manière causale (incrémentale). Dans les travaux de (Smith et al., 1990), le concept d'une carte stochastique a été introduit dans un cadre de SLAM utilisant le filtre de Kalman étendu. Ce dernier permet, à partir de mesures bruitées et des modèles d'observation et d'évolution non linéaires, de fournir conjointement et d'une manière récursive l'estimation de l'état avec une incertitude associée donnée sous forme d'une matrice de variance-covariance. L'incertitude associée est représentée par une Pdf (fonction de densité probabiliste). Suite aux observations effectuées et les nouvelles informations obtenues, la carte est corrigée et mise à jour.

Un exemple de SLAM basé sur un filtre de Kalman dans un environnement intérieur est montré dans (Słowak and Kaniewski, 2020). Le système mis en place utilise un filtre de Kalman ainsi que des données obtenues par un LiDAR 2D. L'approche applique l'algorithme ICP pour calculer la localisation et utilise des techniques de regroupement et de reconnaissance de formes pour construire la carte de l'environnement. Le filtre de Kalman suppose néanmoins que toutes les mesures sont décorréliées et que le bruit suit une distribution gaussienne et surtout des modélisations linéaires, ce qui est rarement le cas. Cela mène à une diminution des performances des algorithmes SLAM surtout du point de vue de la précision de l'estimation. Une solution dans (Zhang et al., 2020b) est proposée en utilisant la méthode de Bayes variationnelle et en modélisant le bruit d'observation comme une distribution de Wishart inverse avec une moyenne harmonique. En outre, une solution par intégration est utilisée pour résoudre le problème du système non linéaire et de la divergence du filtrage. D'autres recherches similaires sont présentées dans (Cai and Zhong, 2015) et (Zhang et al., 2019b) .

Dans (Saman and Lotfy, 2016), une implémentation software et hardware de la technique de SLAM basée sur un EKF est présentée. Les données utilisées sont des données réelles obtenues dans un environnement

intérieur en utilisant un télémètre laser et une paire de roues codeuses. Le filtre EKF étant particulièrement sensible aux mauvaises associations surtout avec un nombre conséquent d'amers, une solution de filtrage de Kalman étendu itéré est proposée dans (Han-hai et al., 2017). Il s'agit d'un SLAM basé sur un EKF utilisé pour faire correspondre les données obtenues par l'IMU et le capteur laser avec les caractéristiques environnementales. L'idée consiste à appliquer l'algorithme du K plus proches voisins pour faire l'association des données. Toutefois, la complexité de l'EKF est quadratique en fonction du nombre d'amers de la carte ce qui rend son utilisation difficile dans des grands environnements. Des solutions dans la littérature existent pour réduire cette complexité telles que (Abouzahir, 2017) : Atlas Framework (Bosse et al., 2003), EKF compressé (Lerma, 2011) (CEKF), filtre d'information étendu (SEIF) (Thrun, 2002b) ou la méthode proposée par (Paz et al., 2008).

Afin de compenser les faibles performances de l'EKF dans le cas de systèmes hautement non linéaires, des solutions de SLAM basées sur un filtre de Kalman sans parfum (UKF) existent dans la littérature. L'idée principale est d'éviter le calcul des jacobiniennes qui rend les estimées inconsistantes. Ce principe est illustré dans Ammann and Mayo (2018) dans le cadre d'un SLAM monoculaire. Cette méthode permet de saisir toutes les corrélations entre les caractéristiques de la carte et les erreurs d'estimations des variables d'état sans qu'il soit nécessaire de calculer des matrices jacobiniennes. Un exemple d'un UKF-SLAM est donné dans (Nemoto et al., 2018) où le modèle dynamique d'un robot quadrupède rampant, qui prend en compte les frottements offerts aux pattes de support, présente des fortes non-linéarités. D'autres travaux sur l'UKF-SLAM peuvent être consultés comme (Gharib and Esmaili, 2019) (Joukhadar et al., 2019).

Le dernier type de SLAM basé sur une méthode par filtrage que nous allons présenter dans cette partie est le FastSLAM utilisant un filtre particulaire (PF). Ce dernier est souvent utilisé lorsque le système à estimer ne suit pas une distribution gaussienne suite à la non linéarité des fonctions. En fait, l'état est échantillonné en un ensemble de particules qui suivent sa densité de probabilité. Chaque particule décrit une trajectoire complète du robot et possède une carte. Chaque carte est composée d'un ensemble d'amers dont la position est estimée par un simple filtre de Kalman (un KF ou un EKF). La complexité de cet algorithme est logarithmique,  $O(M \log(N))$ , où  $M$  est le nombre de particules utilisées et  $N$  est le nombre d'amers de la carte. Cette approche est appliquée à un SLAM par vision comme montré dans (CUI et al., 2016) (Qiu et al., 2012) ou à un SLAM par lidar comme nous le trouvons dans (Shamsudin et al., 2018) (Lin et al., 2019) (Lei et al., 2020). Le problème majeur de cette approche reste la détermination du nombre de particules nécessaire pour estimer avec précision l'état du robot. En effet, un nombre élevé de particules entraîne un temps de calcul important et requiert une grande mémoire tandis que l'utilisation de quelques particules fait diminuer la consistance des résultats.

Pour résumer, bien que cette méthode offre la possibilité d'utiliser différents types de capteur, elle présente toutefois des limites face aux grands environnements. Elle convient plutôt aux environnements à petites tailles. Néanmoins, des solutions à ce problème existent comme l'utilisation de sous-cartes (submapping) (Wang et al., 2018) ou bien la suppression d'amers (Choudhary et al., 2015). Chaque méthode de filtrage a ses avantages et ses inconvénients et diffère des autres du point de vue des hypothèses établies vis-à-vis de la nature du système (système linéaire ou non linéaire) et du bruit associé ainsi que du point de vue de la précision, de la rapidité et de la demande en mémoire. La table II.2, donne un comparatif entre les différentes méthodes de SLAM par filtrage.

### II.3.3.2 Minimisation globale par ajustement de faisceaux

Dans cette partie, nous considérons les algorithmes de type vision-SLAM. Ce type est communément utilisé dans le domaine de la robotique. Il adresse sensiblement le même problème que la technique de structuration à partir du mouvement (structure from motion SfM) qui est couramment utilisée dans le domaine de la vision par ordinateur. La différence majeure entre les deux approches est au niveau de la reconstruction 3D. Alors que le SLAM construit la carte de l'environnement d'une manière incrémentale, la technique SfM nécessite parfois l'intégralité des images ou bien le flux vidéo à traiter. Selon la communauté, il existe deux classes d'outils pour traiter le problème de vision-SLAM : les outils statistiques (par filtrage) pour le SLAM traités dans la partie

Type		Kalman Filter	Extended Kalman Filter	Uncented Kalman Filter	Filtre Particulaire
Exigences	Système	Linéarité du modèle d'évolution du système	Aptitude de traitement des systèmes non linéaires		
	Bruit	Bruit gaussien (suit une loi normale)			Bruit non gaussien (forme quelconque)
Principe		Calcul basé sur l'état actuel et la loi de commande du système			Calcul basé sur un test aléatoire de tous les états futurs possibles (Particules)
Précision		+	++	+++	++++
Rapidité		++++	++++	++	+
Demande de mémoire		--	--	---	-----

TABLE II.2 – Méthodes de SLAM par filtrage (Innowtech, 2019)

précédente (cf. II.2) et les outils géométriques. Parmi ces derniers, la méthode la plus précise et la plus utilisée est l'ajustement de faisceaux (Bundle Adjustment BA) qui permet d'optimiser simultanément les paramètres de la pose de la caméra ainsi que la pose 3D des points appartenant à la carte construite de l'environnement.

Le principe de ces approches consiste à extraire des amers (des primitives) à partir des images, ensuite mettre en correspondance ces amers par un processus d'appariement, et finalement faire une optimisation non linéaire par un ajustement de faisceaux afin de minimiser l'erreur de projection et calculer la géométrie 3D. Un exemple de SLAM visuel est présenté dans (Mennillo et al., 2017). L'idée de ces travaux consiste à réutiliser des systèmes multi-caméras déjà montés sur les véhicules. Cette méthode est incrémentale et permet la reconstruction de points mobiles éparés, grâce notamment à plusieurs contraintes géométriques de segmentation des points reconstruits ainsi que de leur trajectoire. Une reconstruction de séquence complète et sa trajectoire associée, générée par le module SLAM visuel, sont présentées à la figure II.6. Une étude plus détaillée de cette recherche peut être consultée dans (Mennillo, 2019).

L'approche de SLAM visuel permet d'assurer une localisation précise du robot. Néanmoins, la carte construite peut présenter des imprécisions. Nous distinguons deux limites liées à cette approche.

Une première limite est due aux dérives sur de longues trajectoires. En pratique, les images du flux vidéo qui sont utilisées pour extraire les primitives sont bruitées. Ceci peut être dû aux mauvaises calibrations des caméras ou bien à cause des conditions météorologiques (pluie, éblouissement des caméras, changement du niveau d'éclairage de la scène, etc.). De plus, le processus SLAM est un processus incrémental dans la mesure où l'estimation des paramètres de l'état actuel s'appuie sur l'historique des étapes précédentes. Au fur et à mesure, les erreurs s'ajoutent et entraînent par conséquent une reconstruction incohérente. Ce problème est fortement lié à la construction des environnements de grande taille.

Une deuxième limite est liée au facteur d'échelle. En effet, pour le cas d'application avec une seule caméra et sans connaissance a priori sur la géométrie de la scène, l'algorithme est incapable d'estimer le déplacement. Ce dernier est donné par un facteur d'échelle près, théoriquement constant sur l'ensemble de la séquence, mais qui dérive réellement au cours du temps. Néanmoins, des solutions au phénomène de la dérive du facteur d'échelle existent qui consistent et peuvent être consultées dans (Melbouci et al., 2015), (Frost et al., 2018) et (Liu et al., 2018).

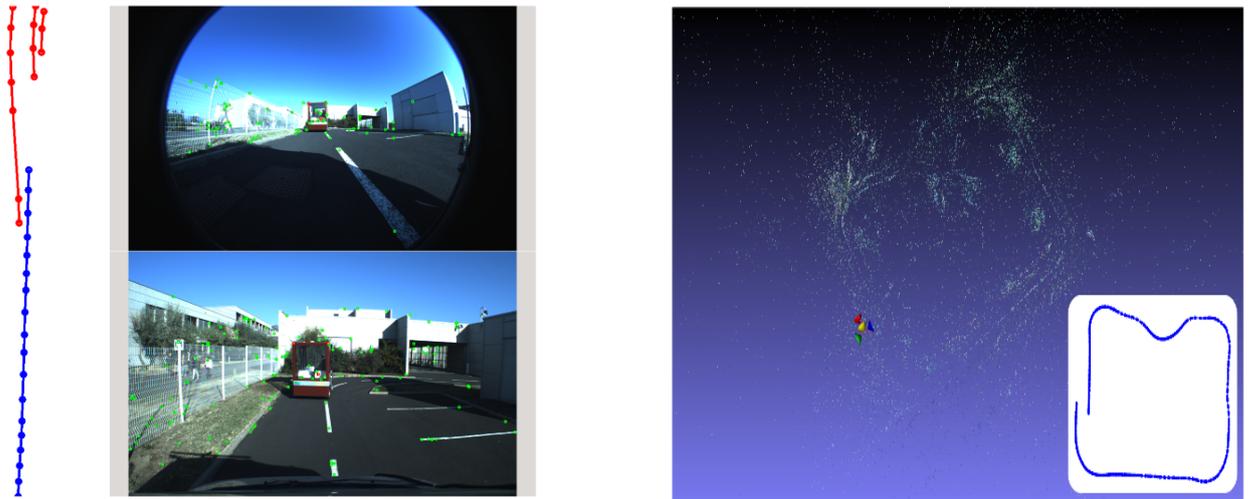


FIGURE II.6 – Résultats tirés de (Mennillo et al., 2017). À gauche, un exemple de reconstruction de trajectoires. Dans cette séquence, le véhicule d’acquisition suit un autre véhicule en mouvement. La trajectoire bleue est celle du véhicule d’acquisition, tandis que les trajectoires rouges correspondent aux points rouges du véhicule en mouvement visibles sur les deux vues de droite. À droite, Reconstruction de l’environnement rigide généré par le module visuel SLAM et sa trajectoire correspondante.

Pour résumer, cette approche de SLAM utilisant des algorithmes d’ajustement par moindres carrés non linéaires présente aussi des limites face aux environnements de grande taille. Un intérêt particulier consiste à assurer une fermeture de boucle cohérente et des optimisations locales et/ou globales pour une économie des ressources. Au delà de la grande quantité d’informations gérée par cette approche, le temps de calcul et la consommation mémoire sont plus élevés comparés aux approches de SLAM basées sur des techniques de filtrage.

### II.3.3.3 Synthèse

Les algorithmes de SLAM basés sur des approches probabilistes semblent être plus utilisés par rapport aux algorithmes basés sur la structuration à partir du mouvement. Ces derniers sont plus récents et commencent à être de plus en plus utilisés dans plusieurs applications robotiques.

La revue de l’état de l’art faite dans cette partie semble confirmer la tendance de cette approche à gérer une quantité très importante d’informations généralement d’une manière ascendante. Par contre, un aspect qui existe et qui reste inhérent à tout algorithme de SLAM c’est la notion de dérive. En effet, la carte générée est une carte locale (non géo-référencée) qui dérive selon la longueur de la trajectoire parcourue. En outre, une grande problématique est liée à l’amélioration de la précision de la carte afin d’assurer une estimation robuste de l’état du robot. Néanmoins, la confiance de la reconstruction dans sa globalité importe peu. Seule la précision locale est nécessaire. Toutefois, une reconstruction non cohérente pourra fausser l’estimation des paramètres d’état du robot et retourner une estimation non consistante. Une autre problématique liée à cette approche est la construction des scènes dynamiques. À chaque fois que la région cartographiée change, une nouvelle carte doit être construite à partir d’autres observations plus récentes.

### II.3.4 Bilan

Dans cette partie nous avons présenté des techniques de localisation relative qui intègrent des données et estiment des vitesses ou des déplacements élémentaires. En effet, la position du robot est calculée par intégration dans le temps ou par cumul de déplacements estimés.

Nous avons commencé par la localisation basée sur les mesures odométriques. Ce type de localisation est un système peu coûteux, rapide et précis à court terme mais reste sujette à des sources d'erreurs qui dégradent son bon fonctionnement.

Ensuite, nous avons traité la localisation par système inertiel. Ce dernier a pour avantage de fournir une information plus complète y compris la position, la vitesse, l'accélération, l'orientation et la rotation du robot. Néanmoins, comme l'odométrie, ce système présente des imprécisions croissantes à long terme et reste sensible à la gravité.

Après, nous avons présenté les systèmes de localisation optique composé des systèmes basés sur des techniques d'estimation visuelle du mouvement où les informations utilisées sont des images ainsi que des systèmes utilisant des informations de formes issues des capteurs de profondeur lidars.

Finalement, nous avons présenté deux approches de SLAM : l'approche par filtrage et l'approche de minimisation globale par ajustement de faisceaux. Ces deux approches semblent séduisantes dans la mesure où elles permettent d'exploiter différents types de capteurs et d'assurer une localisation du robot en offrant en même temps une carte de l'environnement. Cependant, bien que des dérives et des incohérences sont ajoutées lors de la construction de la carte et de l'estimation de la position, ces approches ne se soucient souvent que de la précision de la pose estimée et rarement une métrique décrivant la fiabilité de l'estimation est donnée.

En résumé, les techniques de localisation relative présentent l'avantage de ne pas avoir besoin des informations a priori pour l'estimation du déplacement. Par contre, l'inconvénient majeur est que les erreurs d'estimation s'accumulent dans le temps et créent une dérive de trajectoire. Dans ce type de localisation, la position initiale est le seul élément de référence. Par conséquent, il n'y a aucun moyen de constater cette dérive. Néanmoins, la fusion et la combinaison de capteurs peuvent apporter des améliorations importantes bien évidemment en fonction des caractéristiques des capteurs et de leurs précision (Jones and Soatto, 2011) (Ben-Afia et al., 2014). Afin de diminuer l'influence des dérives dans ce type de positionnement, il est nécessaire de procéder à un recalage régulier par un système de positionnement absolu.

## II.4 La localisation absolue

Ce type de localisation est une technique qui permet de localiser un robot dans son milieu d'évolution (environnement intérieur ou extérieur) en utilisant un repère lié à l'environnement. Ces méthodes utilisent des capteurs extéroceptifs (caméra, télémètre laser, GPS, radar, etc.) qui perçoivent l'environnement de navigation du robot. Les capteurs extéroceptifs retournent la position du robot en observant des références dans l'environnement, appelés amers, dont la position est connue dans un référentiel global attaché à l'environnement.

Les méthodes de localisation absolue peuvent être classées en plusieurs catégories selon l'utilisation d'amers, de satellites ou bien de cartes. Les deux premières catégories seront traitées dans cette sous section alors que la localisation absolue sur une carte sera traitée dans le [chapitre III](#).

### II.4.1 Types d'amers

L'utilisation des amers dans le domaine de la robotique mobile est inspirée des techniques de l'Antiquité des marins naviguant à vue et se repérant en utilisant des documents qui décrivent les côtes abordables en signalant principalement les amers. Remarquable par sa position et ses dimensions, l'amer guidait les navires pour se repérer, rentrer au port et les tenir écartés des dangers.

Le terme amer est ensuite utilisé en robotique mobile pour désigner les points de repère utilisés par un robot pour se repérer dans son environnement de navigation et calculer sa position. Un bon amer doit être très remarquable et reconnu sans ambiguïté.

Les amers observés peuvent être classés en deux catégories : les amers naturels d’une part et les amers artificiels d’autre part.

— Les amers naturels sont des éléments *déjà présents* dans l’environnement et sont exploités pour assurer une fonction de navigation. Un phare, un château d’eau, un clocher, les panneaux de circulation, les bords de la route, etc. peuvent constituer des amers de ce type. Généralement, la tâche de perception d’amers est assurée par des capteurs télémétriques et/ou de vision. Pour cela, nous distinguons :

- des amers télémétriques de haute sémantique comme les murs, les poteaux (Lerasle, 2008) (Delobel et al., 2015) (Kenmogne, 2019) les marquages au sol, les panneaux de signalisation et les bords de route (Qu et al., 2015) (Somawirata et al., 2019) (de Almeida et al., 2020)
- des amers visuels de basse sémantique comme des primitives usuelles comme les points de Harris (Wei et al., 2012) (Wang and Hayes, 2017)

L’intérêt principal de la localisation basée sur des amers naturels est de ne pas avoir besoin d’aménager l’environnement.

— Les amers artificiels (désignés indifféremment sous le nom de *balises*) sont des objets ou des marqueurs placés *spécifiquement* dans l’environnement pour un objectif de navigation. Ce type d’amers conduit souvent à des mesures plus fiables et plus robustes par rapport à la localisation basée sur des amers naturels. Néanmoins, un aménagement préalable de l’environnement est nécessaire. Ce type d’amer est souvent appelé balise (anchor en anglais). Pour ce type d’amers artificiels nous distinguons :

- les balises passives : ce sont des balises placées dans l’environnement qui ne transmettent pas de signaux. La tâche principale consiste dans un premier temps à détecter la balise pour ensuite calculer la pose du robot. Les radars, les télémètres laser, les ultra sons ou encore les systèmes de stéréovision sont exploités pour détecter des balises passives.
- les balises actives : à la différence des balises passives, les balises actives émettent des signaux sonores, lumineux ou encore électromagnétiques, que le robot, équipé d’un équipement récepteur adéquat, peut recevoir. La localisation absolue basée sur des amers actifs était synonyme jusqu’à il y a quelques années de la localisation par GPS (cf §II.4.4). Néanmoins, différentes technologies faisant partie des systèmes de localisation par détection d’amers actifs sont de nos jours en plein essor comme la technologie :
  - \* UWB : le principe de la technique est une trilatération entre un émetteur et un récepteur retournant une mesure de distance évaluée par une mesure du temps de propagation du signal.
  - \* RFID : c’est une technique d’identification par radiofréquence des données numériques encodées sur une étiquette RFID ou une étiquette à puce par le biais d’un lecteur d’ondes radio.
  - \* Bluetooth : c’est une norme de communication permettant l’échange bidirectionnel de données à très courte distance en utilisant des ondes radio UHF (Wikipédia, 2020). Cette technologie est utilisée pour des fins de localisation des robots mobiles comme par exemple dans (Cuadra-Troncoso et al., 2015).
  - \* WiFi : le principe de la technique est la triangulation. Un moteur RTLS (pour Real Time Location System) retourne la position calculée d’un émetteur, se trouvant dans une zone de couverture de trois bornes WiFi, à un opérateur récepteur.

Des travaux de localisation d’un robot mobile s’appuyant sur ces dernières technologies peuvent être consultés respectivement dans : (Benjamin et al., 2015), (Pierre et al., 2018a), (Khosyi’in et al., 2019) et (Yin and Yin, 2018). La table II.3 donne une comparaison entre différentes technologies de localisation basées sur des balises actives.

Technologie	Portée	Précision	Déploiement	Coût
<b>RFID</b> Étiquettes bon marché	Portée de 1 à 8 m et jusqu'à 90m en champ libre	Précision moyenne (+/- 7%)	Déploiement difficile	Coût moyen
<b>Wi-Fi</b> Présent partout	Portée de 25m en extérieur et de 35m en intérieur	Faible précision (de l'ordre de 10m)	Facilité de déploiement moyenne	Coût moyen
<b>Bluetooth</b> Facilité de déploiement	Portée de 10 à 15m	Bonne précision (de l'ordre de 30cm)	Déploiement facile	Coût faible
<b>UWB</b> Haute précision	Portée de 200m	Précision accrue (de l'ordre de 10 cm)	Déploiement difficile	Coût élevé

TABLE II.3 – Table comparative des différentes technologies de localisation basées sur des balises actives tirée de (Zebra, 2014)

## II.4.2 Exploitation des amers

La localisation absolue basée sur l'exploitation d'amers ou balises consiste à utiliser les mesures retournées par les capteurs embarqués sur le robot en observant des amers placés dans l'environnement et en ayant une estimation approximative initiale de la pose du robot. La position de ces amers est supposée connue par le robot avec une certaine précision. L'objectif sera ensuite de reconnaître les amers observés à partir des données acquises et d'en déduire la pose du robot. La détection des amers, la mise en correspondance et l'estimation de la pose vont dépendre de la précision de l'estimation initiale de la pose du robot ainsi que de l'incertitude autour de la position acquise des amers. Le principe général ascendant de cette localisation sur amers est résumé dans la figure II.7.

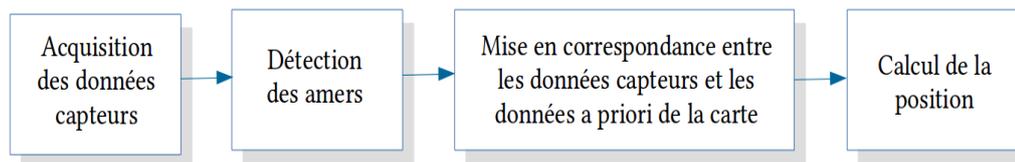


FIGURE II.7 – Principe de la localisation à partir d'amers en utilisant une approche Bottom-Up

## II.4.3 Méthodes de localisation à partir d'amers

Dans cette partie, nous présentons quelques travaux de localisation exploitant des amers. Dans un premier temps, considérons les techniques de localisation par amers naturels. Nous rappelons que les amers naturels sont des amers qui existent déjà dans l'environnement sans qu'ils soient placés spécifiquement pour assurer un objectif de localisation. Dans cette partie, nous traitons la localisation exploitant des amers naturels sans utilisation d'une carte. La localisation sur une carte sera abordée dans le chapitre III.

À titre d'exemple, les auteurs dans (Faisal et al., 2016) proposent un système de localisation pour améliorer la localisation relative des robots mobiles. Le système de localisation proposé utilise le filtre de Kalman étendu afin de résoudre les problèmes de dérive que nous avons énoncés dans la partie §II.3. Le système proposé exécute le cycle du filtre de Kalman étendu, en détectant les murs de l'environnement de navigation. Les murs sont considérés comme des amers naturels, de positions connues, qui permettent de corriger les erreurs de position du robot (améliorer la précision de l'estimation de la pose du robot). Des travaux similaires dans un milieu intérieur de détection de murs d'un couloir pour la localisation d'un robot sont présentés dans (Padhy et al., 2018).

Dans un cadre de localisation absolue d'un robot mobile dans un milieu extérieur, une méthode basée sur la détection des panneaux de signalisation à l'aide d'une caméra embarquée est présentée dans (Hong et al., 2017). Cette recherche s'intéresse à traiter le problème de la correspondance ambiguë entre les amers et leurs caractéristiques, qui se manifeste suite à la présence de plusieurs amers (panneaux) identiques. Des tests à hypothèses multiples sont adoptés pour résoudre ce problème et minimiser les mauvaises associations. Une recherche similaire basée sur la détection des panneaux de signalisation pour affiner la trajectoire d'un robot mobile est disponible dans (Verentsov et al., 2017).

Une autre méthode de localisation basée amers est présentée dans (Kasmi et al., 2019). Les auteurs présentent une méthode permettant de détecter le numéro de la voie de circulation d'un véhicule par vision monoculaire embarquée en utilisant des informations issues d'OpenStreetMap (OSM). L'idée est d'introduire un réseau bayésien modulaire (BN) pour déduire la position de la voie de circulation à partir de multiples sources d'informations imprécises. Des recherches similaires pour des fins de localisation en identifiant la voie de circulation et du marquage au sol peuvent être consultées respectivement dans (Wu and Ranganathan, 2013) et (Choi et al., 2020).

Dans un deuxième temps, considérons la localisation en utilisant des amers artificiels avec des balises passives.

Les travaux dans (Lee et al., 2015) présentent une approche pour la localisation multi-mobiles par vision dans un environnement intérieur en utilisant un capteur de vision bas coût. La caméra embarquée sur le robot et orientée vers le bas permet d'identifier un certain nombre d'objets sur la base de codes de couleur. Ces objets forment des points de repère colorés qui fournissent aux robots mobiles des informations utiles pour la localisation dans la vue de l'image. Ces informations seront ensuite transformées en coordonnées du monde réel et échangées entre les robots par le biais d'un émetteur-récepteur RF connecté au système de vision pour permettre une communication sans fil entre les différents robots. Une recherche similaire basée sur la détection de QR codes placés dans l'environnement de navigation du robot est fournie dans (Yang et al., 2020). Un deuxième exemple peut être trouvé dans (Betke and Gurvits, 1997) où une méthode pour localiser un robot mobile dans un environnement avec des amers fixes est présentée. La recherche repose sur l'identification de ces points de repère et la mesure de leurs orientations les uns par rapport aux autres.

Le principe de la localisation basée sur des amers artificiels actifs peut être aussi illustré dans (Zhou et al., 2008) où un algorithme de localisation des balises concentriques est proposé. Quatre balises sont placées aux quatre coins de la région cible carrée et émettent des signaux de localisation à différents niveaux de puissance, avec l'information de leur position et la portée estimée de la balise. Les signaux sont reçus par les nœuds dans cette région et leurs positions sont calculées en conséquence. Une recherche similaire dans le cadre de la localisation d'un convoi de robots est présentée dans (Pierre et al., 2018b) en utilisant des mesures de distance uniquement effectuées par des capteurs ultra-large bande (UWB pour ultra wideband) peu coûteux. Une mesure balise à balise est effectuée à l'aide de capteurs UWB et l'estimation de la position est calculée par la balise cible. L'algorithme de localisation est basé sur un filtre SCIF (pour Split Covariance Intersection Filter) qui permet de traiter le phénomène de la corrélation entre les estimations de pose des balises.

Dans (Vallicrosa et al., 2014), l'auteur présente une solution au problème de la localisation des balises en utilisant un filtre SOG (pour Sum Of Gaussian) ainsi qu'une méthode de localisation active, qui est basée sur la minimisation de l'incertitude de la position de la balise. La méthode est appliquée pour localiser de manière autonome un panneau sous-marin. Dans le même contexte de la navigation et de la localisation des robots sous-marins, une méthode de localisation à l'aide d'une seule balise acoustique, en supposant que la profondeur de la balise soit connue a été introduite dans (Becker et al., 2012).

Dans cette catégorie de localisation basée sur des amers artificiels actifs, nous pouvons considérer que les satellites de géolocalisation sont des amers actifs. Par conséquent les systèmes GNSS sont des systèmes de localisation absolue exploitant des amers. Étant donné la richesse des travaux exploitant ces systèmes, ils seront

introduits séparément dans le prochain paragraphe §II.4.4.

## II.4.4 Systèmes GNSS

Un système de positionnement par satellites GNSS (pour Géolocalisation et Navigation par un Système de Satellites) est un ensemble de trois parties : une partie spatiale (constellation de satellites), une partie de contrôle (des stations au sol situées à travers le monde) et une partie utilisateur (un récepteur qui reçoit les informations des satellites). Un système GNSS permet de fournir à un utilisateur sa position en coordonnées géographiques en trois dimensions (longitude, latitude, hauteur ellipsoïdale), sa vitesse de déplacement et la date/heure.

Différents systèmes GNSS qui permettent la navigation et la localisation existent. Le plus connu et celui qui est opérationnel depuis le plus longtemps est le système américain GPS (pour Global Positioning System). D'autres versions existent comme les versions européenne Galileo, russe GLONASS et chinoise Beidou.

Le calcul du géopositionnement obéit au principe de trilatération en calculant la distance entre l'antenne réceptrice et un certain nombre de satellites dont la position est connue (Wassila Leila et al., 2015). Cette distance est obtenue grâce à sa relation de proportionnalité avec le temps de parcours du signal émis par chaque satellite.

La précision dépend de très nombreux facteurs dont les facteurs atmosphérique et la qualité du récepteur. La précision de la localisation absolue fournie par un GPS bas coût peut descendre jusqu'à plusieurs mètres. Cette précision n'est pas satisfaisante pour des applications de conduite autonome. Des solutions plus onéreuses sont néanmoins disponibles comme le GPS RTK (pour Real Time Kinematic) qui offre une précision centimétrique. Une étude sur différents types de systèmes GPS peut être consultée dans (Yozevitch and Ben-Moshe, 2019). En outre différents problèmes sont associés aux systèmes GNSS à savoir :

- le problème des multi trajets dans des environnements cloisonnés où les signaux émis par les satellites sont réfléchis par des bâtiments au lieu d'être reçus directement par le récepteur. Ce phénomène est problématique dans la mesure où la position fournie n'est pas intègre (la position réelle du robot n'est pas dans la zone d'incertitude de la mesure obtenue). Ce point sera évoqué plus en détails dans la partie §V.6.3 du chapitre §IV.
- le problème de canyon urbain en milieux urbains avec de grands immeubles où le signal GPS est perdu à cause d'un masquage des satellites.

Néanmoins, des solutions palliatives existent afin de corriger ces problèmes comme la technologie RAIM (pour Receiver Autonomous Integrity Monitoring) qui utilise la redondance d'information pour lever l'ambiguïté (Milner and Pervan, 2020). Dans (Zair, 2016), l'auteur propose un algorithme de détection de mesures de pseudo-distances aberrantes exploitant la modélisation a contrario. Une autre méthode est donnée dans (Hong et al., 2018) où les auteurs présentent une méthode pour rejeter les signaux GPS aberrants sur la base de l'estimation de la forme d'onde de la signature. Une autre solution présentée dans (Zhang and Masoud, 2020) permet d'apprendre une stratégie optimale pour apporter des corrections sur les observations GPS brutes. Le modèle proposé utilise un mécanisme de compensation efficace basé sur la confiance. D'autres méthodes d'amélioration de la mesure GPS peuvent être consultées dans (Pollard and Gingras, 2012) (Encarnacion and Tiglao, 2016) (Moreau et al., 2017).

Plusieurs recherches dans le domaine de la localisation d'un robot mobile basées sur l'utilisation des systèmes GNSS existent. Afin de pallier le problème de la non disponibilité de l'information GPS, ces recherches ont souvent recours à la fusion des données. L'intérêt de la fusion des données sera abordé plus en détails dans la partie §II.6. En effet, l'utilisation d'autres capteurs permet d'assurer la continuité de la tâche de localisation

lorsque le signal GPS n'est pas disponible. Un exemple donné par (Cai et al., 2019) traite la navigation d'un robot mobile dans un environnement extérieur. Le principe est basé sur la fusion des données de l'IMU, du GPS différentiel et de l'odométrie visuelle à l'aide d'un filtre de Kalman étendu. D'autres travaux sur l'utilisation d'autres capteurs avec les données GNSS peuvent être consultés dans (Al Khatib et al., 2015) (Kaviani et al., 2016) (Yousuf and Kadri, 2016).

## II.5 Bilan : localisation relative et localisation absolue

Dans cette section, nous avons présenté les deux principales méthodes de localisation d'un robot mobile dans son environnement que sont la localisation relative et la localisation absolue.

La localisation relative dite localisation à l'estime, permet de déterminer la position et l'orientation d'un robot en utilisant ses déplacements successifs depuis une configuration de départ connue. Les capteurs utilisés sont des capteurs proprioceptifs mesurant de manière rapide et fréquente des grandeurs propres au fonctionnement interne du robot. Cependant, le principal inconvénient est l'accumulation de la dérive au cours du temps.

L'avantage des méthodes de localisation absolue par rapport aux méthodes de localisation relative est l'absence du phénomène de dérive que présentent ces dernières. Ce type de localisation est basé sur l'utilisation de capteurs extéroceptifs et des amers de l'environnement selon deux sortes de stratégies : la première est la localisation basée sur des amers naturels et la deuxième est la localisation basée sur des amers artificiels. Néanmoins, la localisation absolue dépend directement de la visibilité et de l'observabilité des amers par le système de perception. En plus, un aménagement de l'environnement est souvent nécessaire (pour le deuxième stratégie) ce qui implique un investissement lourd en équipement.

Ces deux méthodes de localisation mettent toutes en jeu différents types de capteurs et exploitent des informations variées. Ces dernières sont souvent imparfaites (entachées d'erreurs) et présentent des incertitudes ainsi que des ambiguïtés. La table comparative II.4 montre la complémentarité entre les deux méthodes. Ainsi, pour tirer profit de ces deux méthodes, l'objectif principal sera d'exploiter au mieux les avantages que présente chaque méthode en adoptant une méthode basée sur la fusion d'informations. Dans cet ordre d'idée, les données proprioceptives et extéroceptives qui sont collectées par les capteurs du robot doivent être traitées et combinées pour en extraire des informations plus complètes et de meilleure qualité. En effet, les algorithmes de fusion ont pour mission d'exploiter au mieux la richesse et la complémentarité des informations pour répondre à un objectif donné. Nous nous intéresserons dans la partie suivante à l'introduction de la fusion d'informations.

Localisation relative	Localisation absolue
(-) Accumulation d'erreurs avec la distance	(+) Erreurs indépendantes de la distance
(-) Repère attaché à la position initiale du véhicule	(+) Repère rattaché à l'environnement
(+) Autonomie vis-à-vis de l'environnement	(-) Aménagement parfois nécessaire
(+) Fonctionnement à fréquence élevée	(-) Fonctionnement à fréquence très faible
	(-) Perte d'informations possible

TABLE II.4 – Comparatif localisation relative - localisation absolue (source : (Chausse et al., 2006))

## II.6 La fusion d'informations pour la localisation

Historiquement, la fusion d'informations a été appliquée dans le domaine de la robotique ainsi qu'aux applications militaires, spatiales et avioniques, (Azimirad et al., 2015)(Sabatini, 2016). À ce jour, plusieurs domaines de recherche l'ont trouvée utile. En effet, la fusion d'informations est utilisée dans le domaine de l'intelligence artificielle (Wang et al., 2019b), la vision par ordinateur (Kirtay et al., 2020), la télédétection (Schmitt and Zhu, 2016) et le positionnement (Kubelka et al., 2015) (Gao et al., 2019). Les techniques et algorithmes utilisés peuvent être classés en fonction du type des informations, de l'objectif de l'application ou du

fondement mathématique.

Un grand enjeu abordé dans nos travaux est la fusion d'informations. Le choix d'une architecture particulière et d'un algorithme donné doit être justifié par l'objet de l'application. En effet, nous pouvons distinguer différentes classifications des architectures de fusion. Mais avant cela, nous nous attachons à donner une définition de la fusion d'informations puis nous présentons une classification des différentes architectures. Un point particulier sur les approches de traitement de l'information que sont l'approche Bottom-Up et l'approche Top-Down sera discuté.

### II.6.1 Définition

Plusieurs définitions ont été données à la fusion d'informations dans la littérature. Nous ne présentons ici que celles que nous considérons les plus proches de notre domaine d'application.

- **Définition 1 :** *"La fusion est un processus multi-niveaux et à facettes multiples ayant pour objet la détection automatique, l'association, la corrélation, l'estimation et la combinaison d'informations de sources singulières et plurielles"* selon le groupe de travail le Joint Directors of Laboratories (JDL) (White, 1991) qui s'est créé en 1986 et qui s'est intéressé à la définition de la terminologie de la fusion de données.
- **Définition 2 :** *"La fusion d'informations consiste à combiner des informations issues de plusieurs sources afin d'améliorer la prise de décision"* selon (Bloch, 2003).
- **Définition 3 :** *"La fusion d'informations consiste à regrouper des informations issues de plusieurs sources d'informations et à exploiter l'information regroupée"* selon le groupe de travail européen FUSION.

Dans toutes ces définitions nous distinguons plusieurs notions qui ont été employées autour de la fusion. Nous constatons que l'information peut être issue de différentes sources comme elle peut être donnée par une même source à des instants distincts mais d'un autre point de vue comme par exemple des primitives extraites d'images hétérogènes prises à des instants distincts et d'un point de vue différent.

Enfin l'objet de la fusion dépend du cadre de l'application. Il est défini dans ces définitions par les termes associer, combiner, regrouper et estimer. Nos travaux se focalisent sur ce dernier volet (Hao and Sun, 2020) (Peng et al., 2020) (Zhang et al., 2020a). En effet, nous pouvons distinguer trois objectifs liés à un système de fusion. Pour commencer, la fusion d'informations peut être utilisée pour l'estimation d'un paramètre obtenu en combinant des données issues de différentes sources. Un système de fusion est également exploité pour des applications de reconnaissance (Vasquez et al., 2018) (Wang et al., 2019a) (Jia et al., 2019). Enfin, c'est aussi employé pour l'association de données ayant pour objectif de trouver parmi différentes données, celle émise par la même source (Yan et al., 2016) (Dos Santos et al., 2019) (Hong et al., 2017). Un exemple pour cet objectif est la poursuite de cibles (Isobe et al., 2018) (Kyriakides, 2019).

La difficulté pour un système de fusion sera donc de gérer un grand volume de données avec un traitement adéquat de ces données (suppression d'une donnée redondante ou d'une donnée aberrante). La définition que nous retenons pour nos travaux est : *"l'opération consistant à combiner de l'information pour l'aide à la prise de décision. Cette combinaison est loin d'être réduite à une simple union de l'information issue de différentes sources"*.

### II.6.2 Classification des techniques de fusion d'informations

La fusion de données est un domaine multidisciplinaire qui implique plusieurs thématiques, et il est difficile d'établir une classification claire et stricte. Les méthodes et techniques employées peuvent être réparties selon

les critères suivants :

- selon les relations entre les sources de données en entrée du système de fusion. Ces relations peuvent être définies comme :

(a) complémentaires (ou redondantes) : l'utilisation de plusieurs sources d'informations différentes (ou identiques) permet d'ajouter de la complémentarité aux informations fournies et plus de précision que pour une solution où chaque capteur est utilisé seul. Dans (Delobel et al., 2015), les auteurs présentent les améliorations apportées à la précision de l'estimation de la pose d'un robot mobile en passant de l'utilisation d'un lidar à l'utilisation de cinq lidar.

(b) compétitives : afin de minimiser l'incertitude et les erreurs sur la localisation, une manière de faire est d'utiliser deux capteurs différents pour réaliser la même fonction. Par exemple fusionner les données GPS et les données odométrie permet de réduire les effets des données incertaines et erronées. Un exemple de fusion de données en fonction des propriétés des données GPS et de l'odométrie visuelle est cité dans (Jiang et al., 2018).

(c) coopératives : l'utilisation de plusieurs sources d'informations permet d'avoir une information indisponible avec chaque source prise individuellement. Par exemple, une combinaison des données de deux caméras placées une à l'avant et l'autre à l'arrière du robot permet de fournir une vision plus générale sur la scène de navigation (Matsumoto et al., 2011). Une méthode de détection de robots utilisant deux caméras ayant des distances focales différentes est donnée dans (Dinh et al., 2020).

- selon les types de données d'entrée/sortie et leurs natures comme proposé par Dasarthy (Dasarthy, 1997). Le modèle de fusion Dasarthy (DFD) (pour Data-Feature-Decision) est classé en fonction des types de données/informations qui sont traités et des types qui résultent du processus. Son flux de données est caractérisé par un processus d'entrée/sortie (I/O). Les trois principaux niveaux d'abstraction de la fusion des données sont identifiés comme suit :

(a) données (Data) : ou plus précisément les données capteurs

(b) caractéristiques (Feature) : ou informations de niveau intermédiaire

(c) décisions (Decision) : symboles ou valeurs non compatibles

Dans le modèle de fusion de Dasarthy, il y a cinq catégories possibles de fusion, illustrées dans le tableau II.5 comme types d'entrée/sortie considérés.

La principale contribution de la classification de Dasarthy est la spécification du niveau d'abstraction soit en entrée soit en sortie, fournissant ainsi un cadre pour classer les différentes méthodes ou techniques d'un système de fusion. Afin de positionner nos travaux avec ce type de classification, le système de fusion présenté dans nos travaux sera un système de type decision in-decision out (DEI-DEO) où différentes informations issues de l'utilisation de différents capteurs embarqués sont exploitées.

- D'autres classifications possibles de fusion de données basées sur les niveaux d'abstraction des informations employées sont utilisées. Parmi ces niveaux d'abstraction nous trouvons :

(a) la fusion bas niveau : les données brutes (signaux directement issus des capteurs) sont directement fournies en entrée du système de fusion (Scharwächter and Franke, 2015).

Mode	Exemple	Références
Data in-data out (DAI-DAO)	Fusion de données multi-spectrales	(Yokoya et al., 2012) (Duriković and Kunovsky, 2019)
Feature in-feature out (FEI-FEO)	Fusion de données image et non-image	(Liu et al., 2013) (Qayum and Razzaq, 2019)
Decision in-decision out (DEI-DEO)	Lorsque les capteurs ne sont pas compatibles	(Agarwal et al., 2019) (Semenas and Bausys, 2018)
Data in-feature out (DAI-FEO)	Extraction de formes	(Özcan and Ünsalan, 2020) (Zhao et al., 2018)
Feature in-decision out (FEI-DEO)	Reconnaissance des objets	(Loghmani et al., 2019) (Knyaz, 2019)
Data in-decision out (DAI-DEO)	Reconnaissance de formes	(Kulkarni et al., 2019) (DENCEUX, 1998)

TABLE II.5 – Table de fusion d’informations selon le modèle de classification Dasarathy

(b) la fusion de niveau moyen : également connue sous le nom de niveau de caractéristiques ou d’attributs ; des informations extraites après des phases de traitement des mesures sont fusionnées (Wang et al., 2017).

(c) la fusion de haut niveau : ce niveau également appelé fusion de décision, prend comme sources d’entrée des représentations symboliques (ou des décisions) et les fusionne pour obtenir une décision globale à partir des décisions d’entrée (Laudy and Museux, 2019).

(d) la fusion à niveaux multiples : ce niveau concerne les données fournies à partir de différents niveaux d’abstraction (c’est-à-dire lorsque par exemple une mesure est combinée avec une caractéristique pour obtenir une décision) (Snidaro and Visentini, 2016) (Xu and Chen, 2018).

La figure II.8 illustre les différents niveaux de fusion pour les données issues de deux sources  $S_1$  et  $S_2$ . L’extraction de l’information décisionnelle est effectuée de bas en haut dans les quatre schémas, qui se différencient par les différents niveaux de fusion (Lamalle, 2012).

- selon le type d’architecture : en effet, une des principales questions qui se pose lors de la conception d’un système de fusion de données est de savoir où le processus de fusion d’informations est effectué. Sur la base de ce critère, nous distinguons les types d’architectures suivants :

(a) architecture centralisée : dans cette architecture, le nœud de fusion réside dans le processeur central qui reçoit les informations de toutes les sources d’entrée. Par conséquent, tous les processus de fusion sont exécutés dans un processeur central qui utilise les mesures brutes fournies par les sources. Dans ce schéma, les sources n’obtiennent que les mesures des observations et les transmettent à un processeur central, où le processus de fusion d’informations est effectué. La grande quantité de bande passante nécessaire pour envoyer des données brutes par le réseau ainsi que les délais de transfert des informations et le problème de synchronisations sont les inconvénients majeurs de ce type d’architecture.

(b) architecture décentralisée : Elle est composée d’un réseau de nœuds dont chaque nœud a ses propres capacités de traitement sans y avoir de point unique de fusion d’informations. Par consé-

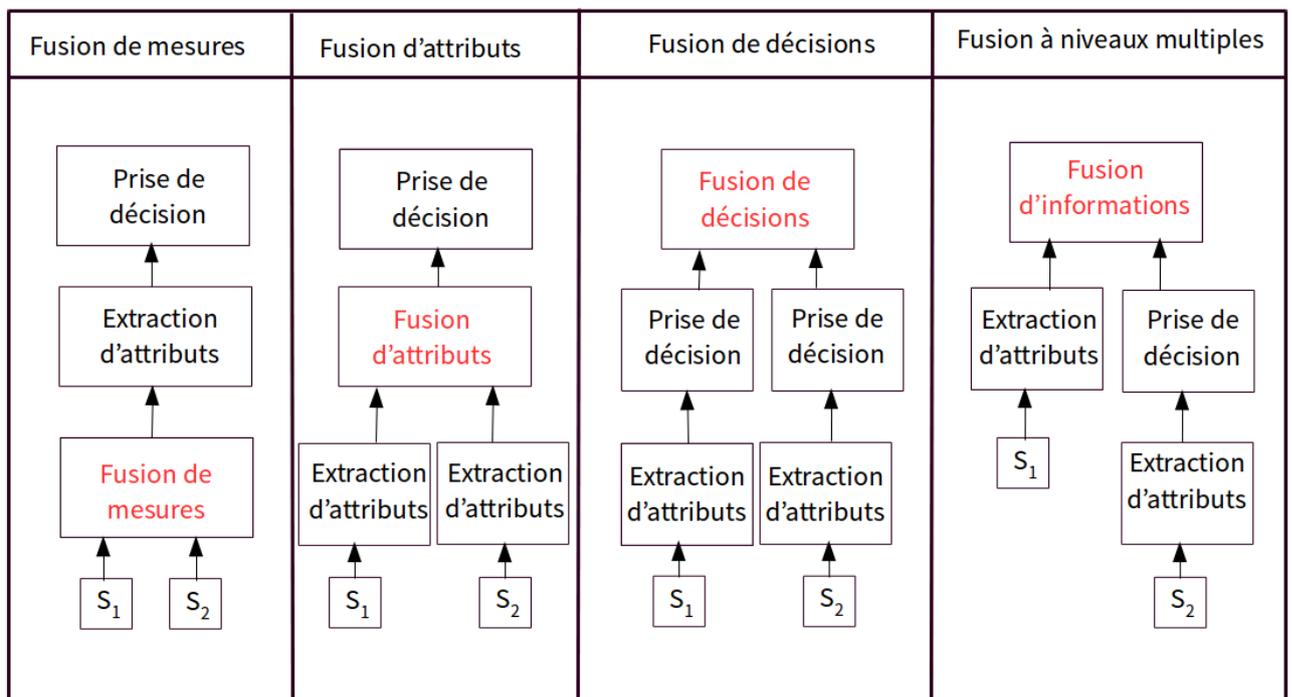


FIGURE II.8 – Niveaux de fusion selon les niveaux d'abstraction

quent, chaque nœud fusionne ses informations locales. Le principal inconvénient de cette architecture est le coût de communication, qui est de l'ordre de  $O(n^2)$  ( $n$  est le nombre de nœuds) lorsque le nombre de nœuds est très élevé.

(c) architecture distribuée : les mesures de chaque nœud source sont traitées indépendamment avant que les informations ne soient envoyées au nœud de fusion. Par ailleurs, l'association de données et l'estimation de l'état sont effectuées dans le nœud source avant que l'information ne soit communiquée au nœud de fusion. Ce type d'architecture offre différentes options et variations qui vont d'un seul nœud de fusion à plusieurs nœuds de fusion intermédiaires.

(d) architecture hybride : une architecture conçue à partir des architectures décentralisée et distribuée.

- sur la base du modèle JDL proposé par le Joint Directors of Laboratories du ministère de la défense aux États-Unis d'Amérique. C'est un modèle extrêmement populaire dans le domaine militaire qui a fait l'objet de nombreuses études (Foo and Ng, 2013) (Swart et al., 2019). Ce modèle de fusion est composé de cinq niveaux de traitement, une base de données associée et un bus d'informations qui relie les cinq composants. Les cinq niveaux peuvent être regroupés en deux groupes, la fusion de bas niveau et la fusion de haut niveau, qui comprennent les composants suivants :

(a) les sources : peuvent être de différents types (ou de différents niveaux d'abstraction) allant des données capteurs à des informations a priori (des références ou données géographiques) et des bases de données à la connaissance humaine.

(b) système de gestion de bases de données : le système de gestion de base de données stocke les informations fournies et les résultats fusionnés. Ce système est un élément essentiel en raison de la grande quantité d'informations très diverses qui y sont stockées.

(c) Interface Homme-Machine (IHM) : est une interface qui permet à l'utilisateur d'intervenir. Elle permet aussi de faciliter la communication des résultats de fusion aux utilisateurs.

Ce modèle est présenté plus en détails dans (Reynaud and Bouaziz, 2005), (Lamalle, 2012) et (Castanedo, 2013).

Le modèle JDL diffère du modèle Dasarathy en ce qui concerne la terminologie adoptée et l'approche utilisée. Il ne donne aucun détail sur la façon dont une sortie globale d'un système de fusion particulier peut-être générée et contrôlée. Il s'agit d'un schéma d'organisation fonctionnelle dans lequel les flux d'informations entre niveaux ne sont pas décrits.

- selon la stratégie d'exploitation des sources d'informations. En effet, la façon d'exploiter les informations peut être classée en deux parties : les approches ascendantes Bottom-Up et les approches descendantes Top-Down. Nous détaillerons dans la partie §II.6.3 ces deux grands principes car notre approche se base sur ce type de classification.

Pour conclure, un système de fusion d'informations comme pour tout processus, doit être organisé et construit de manière à répondre à l'objectif de l'application. En conséquence, il n'existe pas un modèle générique pour la classification d'un système de fusion d'informations. Cette dernière dépend fortement de l'application.

### II.6.3 Stratégies de fusion ascendante et descendante

Deux stratégies de traitement de l'information sont largement utilisées dans le domaine de la fusion que sont : l'approche ascendante (*Bottom-Up*) et l'approche descendante (*Top-Down*) qui seront présentées dans les deux parties qui suivent.

#### II.6.3.1 Les systèmes Bottom-Up

La stratégie Bottom-Up (ascendante) consiste à combiner plusieurs données pour obtenir des données plus complexes. En effet, c'est une approche dirigée par les données (appelée en anglais *data-driven*). La plupart des applications existantes utilisent ce principe de fusion de données ascendante. L'idée est de faire remonter toutes les informations disponibles et de les fusionner pour en déduire de l'information qui permettra ou non d'atteindre les objectifs fixés. Ce principe est illustré dans la figure II.8 où toutes les données issues des différents niveaux d'abstraction sont remontées et transmises au nœud de fusion. Nous citons à titre d'exemples les applications d'évitement d'obstacles (Jha et al., 2019) (Zong et al., 2020) et de détection de piétons (Cheng and Zhang, 2019) (Zhang et al., 2019a).

Un système de fusion de type Bottom-Up se caractérise par la gestion d'un grand nombre de données. Les auteurs dans (Atanassov et al., 2018) mettent en avant le problème lié à la complexité de la classification des mesures de télédétection et du grand volume de données à traiter dans un contexte de l'observation de la Terre EO (pour Earth Observation). Ils présentent, dans un contexte d'analyse et d'interprétation des données avec une grande précision, un système multi-capteurs combinant les images et les informations spectrales. La technique montrée comprend la validation des systèmes multi capteurs, le traitement de données volumineuses et l'identification et la classification des objets pour améliorer la qualité de l'information par la fusion des données.

Gérer un grand volume de données implique le traitement de données redondantes. Les auteurs dans (Bhinge et al., 2017), introduisent une solution pour la détection des objets abandonnés dans des zones très fréquentées avec des problèmes d'occultation. La solution s'appuie sur le modèle (tIVA) (pour transposed Independent Vector Analysis). L'objectif de cette technique est de tirer parti de la similarité entre les différentes caméras ainsi que de la complémentarité des informations et d'éviter par conséquent les étalonnages répétitifs

des caméras.

Cependant, ce type d'approche ne prend pas en compte le contexte actuel et les données a priori pour sélectionner les données entrantes en fonction de leur poids. En effet, cette approche commence par collecter toutes les données capteurs disponibles et ensuite les fusionne pour en déduire des informations qui permettront ou non d'atteindre les objectifs fixés. Des travaux dans le domaine de la robotique qui utilisent la fusion de toutes les données collectées peuvent être consultés dans (Bian and Li, 2017) (Keatmanee et al., 2018) (Cabi et al., 2019) (Niewola and Podszędkowski, 2020) .

Dans le domaine de la robotique mobile, l'utilisation de cartes et leur construction sont largement utilisées comme nous l'avons décrit dans la partie II.3.3 pour l'application de SLAM. Les techniques de construction de cartes sont nécessairement des approches ascendantes (Ahmed et al., 2015) (Gao et al., 2018) (Alsahfi et al., 2019). En effet, pour construire des cartes, il est nécessaire de détecter les éléments de l'environnement même si nous ne sommes pas sûrs de l'existence de ces éléments. Pour cela, aucun a priori n'est considéré. De nombreuses applications s'appuyant sur différentes bases mathématiques pour la localisation de robots en milieu urbain ont été réalisées dans lesquelles le système traite cette grande quantité d'informations stockées dans la carte pour se localiser (Yokoya et al., 2012) (Osman et al., 2019) (Jin et al., 2019). D'autres applications sont possibles en utilisant l'approche ascendante liée à l'utilisation d'une carte comme par exemple l'identification des objets ou des caractéristiques les plus saillants d'une scène donnée (Zhang and Wang, 2020). Aussi, il est possible d'améliorer et enrichir les données de la carte comme dans (Delobel et al., 2017) (Bastani et al., 2019). D'autre part, cela peut conduire à un temps de calcul important, en particulier avec les systèmes multi-capteurs, car toutes les sorties des capteurs seront traitées et analysées.

### II.6.3.2 Les systèmes Top-Down

Contrairement à l'approche ascendante, l'approche Top-Down (descendante) est dirigée par un objectif à atteindre. En effet, l'approche *Top-Down* part des informations connues ou des objectifs souhaités et sélectionne les données utiles en fonction des objectifs fixés pour prendre une décision. Par exemple, pour la détection d'objets, la connaissance de la distance entre le détecteur et l'objet à détecter ainsi que la connaissance des caractéristiques intrinsèques du capteur peuvent considérablement améliorer la qualité de la détection.

La principale technique utilisée dans un système descendant est *la focalisation*. Au lieu de considérer l'ensemble de la scène, le système sélectionne les données les plus informatives et ignore celles qui ne sont pas pertinentes. Une focalisation dans les ensembles capteurs et détecteurs pour sélectionner le meilleur capteur et/ou détecteur à utiliser pour obtenir les données prévues est aussi effectuée. En fait, le fonctionnement d'un système Top-Down est très similaire au fonctionnement du cerveau humain (Karabegović and Doleček, 2017). Prenons l'exemple d'une scène de conversation dans un environnement bruyant où plusieurs personnes échangent en même temps. Le cerveau humain est capable dans de telles circonstances de focaliser toute son attention par l'intermédiaire de ses sens (capteurs/détecteurs) de l'ouïe et de la vue sur ce qui l'intéresse et de ne pas se laisser perturber par ce qu'il ne juge pas d'intérêt.

Dans (Aynaud et al., 2017), les auteurs ont introduit un système perceptif de prise de décision dans un contexte de localisation d'un robot mobile utilisant un processus descendant permettant de sélectionner : l'amer de l'environnement à observer, le capteur qui doit être utilisé, la région d'intérêt ROI (pour Region Of Interest) dans laquelle l'amer doit être observé et le détecteur correspondant à utiliser pour la détection dans la ROI choisie. Une recherche similaire est donnée par (Schult et al., 2015). Qualifié de système actif, un système Top-Down est un système qui anticipe le futur en utilisant les informations précédentes et des données a priori. D'autre part, les applications citées plus haut dans §II.6.3.1 pour l'approche ascendante (ou Bottom-Up), pourraient être aussi bien conduites d'une manière descendante. Par exemple, nous trouvons les applications de navigation (Nguyen et al., 2015), (Hou et al., 2020), (Wang, 2020), de planification de trajectoire (Schwartz et al., 2018), de suivi (Ruan et al., 2018), de détection et de classification d'objets (Roheda et al., 2019), de détection des points saillants (Murabito et al., 2018) et de SLAM (Grant et al., 2019).

### II.6.3.3 Synthèse des approches Bottom-Up et Top-Down

Nous avons présentés deux grands types d'approches multi sensorielles : les méthodes ascendantes dites Bottom-Up et les méthodes descendantes dites Top-Down. Le domaine qui nous intéresse est la fusion d'informations. Ces méthodes ont été utilisées dans le but de traiter un flux d'informations pour répondre à l'objectif initial de l'application. Le tableau II.6 donne un comparatif entre ces deux approches.

Dans la littérature, des travaux ont été réalisés pour intégrer les deux approches et tirer profit des avantages de chacune d'entre elles. C'est le cas par exemple pour l'enrichissement de cartes pour des fins de localisation (Delobel et al., 2018) ou bien pour les cartes de saillance (Shariatmadar and Faez, 2019).

Approche Bottom-Up	Approche Top-Down
(+) facile à mettre en œuvre	(-) plus complexe à mettre en œuvre
(+) nécessite peu d'intelligence de l'algorithme	(-) nécessite une intelligence de l'algorithme
(-) aucun a priori n'est utilisé	(+) utilisation des a priori et des données contextuelles de la scène
(-) sensible aux bruits de la scène et pas très performante face aux problèmes d'occultation	(+) la modélisation préalable et la focalisation spatiale permettent de gérer des situations de non détection et d'occultation
(+) gestion facile des imprévus	(-) incapable de gérer les événements imprévus
(-) le grand volume de données à traiter augmente la complexité du problème d'association des données	(+) une meilleure réduction des possibilités de mauvaises associations
(+) très performantes en terme de précision dans la mesure où la fusion d'un grand volume de données permet d'améliorer considérablement la précision	(+) une focalisation dans l'espace permettant une meilleure détection de l'objet
	(+) une focalisation au niveau de l'espace, des caractéristiques et des ressources
	(+) une économie des ressources

TABLE II.6 – Table comparative entre les deux approches Bottom-Up et Top-Down

## II.7 Conclusion

La fusion de données multisensorielle est très répandue en robotique mobile. En effet, la combinaison de toutes les données proprioceptives et extéroceptives permet d'améliorer la localisation des robots mobiles, qui était souvent obtenue soit par localisation à l'estime, soit par des méthodes géométriques exploitant les mesures d'environnement. Toutefois, le robot peut disposer d'autres sources d'informations a priori qui pourront améliorer la qualité de son estimation de pose.

En général, l'information fournie a priori décrit l'environnement de navigation du robot. Les éléments statiques peuvent être spécifiés et exploités par le robot comme nous l'avons montré dans la partie de localisation absolue basée amers. Cette information a priori peut être collectée par le robot mobile lui-même ou fournie dans une phase d'initialisation par une source extérieure. Une source classique d'information a priori en robotique mobile est la mise à disposition d'une carte de l'environnement. Au travers du chapitre suivant, nous illustrerons différentes méthodes de localisation sur une carte existante de l'environnement permettant d'exploiter au mieux les informations collectées tout en prenant en compte deux critères principaux que sont la précision et la confiance des données.

# Chapitre III

## Localisation sur une carte existante

### Sommaire

---

III.1	Introduction	48
III.2	Map-Matching et fusion multi-sensorielle	48
III.2.1	Approches géométriques	49
III.2.2	Approches topologiques	50
III.2.3	Approches avancées	51
III.3	Récapitulatif	56
III.4	Choix et conséquence	56
III.4.1	Choix de la carte	57
III.4.2	Choix de l'approche	58
III.5	Bilan	59
III.6	Conclusion	59

---

## III.1 Introduction

Le choix de la méthode de localisation est primordial. C'est d'ailleurs en fonction des capteurs à disposition que ce choix doit être pris. En effet, cette méthode doit être en adéquation avec les capteurs choisis ainsi que les informations a priori dont dispose le robot. Au travers du chapitre précédent, nous avons montré les méthodes de localisation exploitant ces capteurs. Ce chapitre traitera les méthodes de localisation exploitant des cartes existantes de l'environnement qui peuvent être considérées comme des sources initiales d'informations.

Une carte peut être construite et utilisée en temps réel et c'est le cas des applications de SLAM (cf [sous-section II.3.3](#)) ou être apprise au préalable dans une phase d'initialisation et exploitée par un algorithme de localisation. Dans ce chapitre, nous nous intéressons à cette seconde catégorie de localisation sur une carte existante. Dans la première partie, une étude détaillée sur les techniques de localisation sur une carte, basées sur différents outils théoriques, sera exposée. Ensuite, un bilan récapitulatif sur les principales thématiques que nous avons évoquées est donné. Finalement, une partie justifiant nos choix de carte et d'approche sera présentée à la fin de ce chapitre.

## III.2 Map-Matching et fusion multi-sensorielle

Une grande partie de cette thèse traite le problème de la localisation d'un robot mobile sur une carte. Le terme anglo-saxon utilisé pour désigner la localisation sur une carte (souvent routière) est 'map-matching' (ou bien 'road-matching' pour un problème de sélection de la voie de circulation). Il s'agit de mettre en correspondance les données perçues avec une représentation cartographique donnée dans une carte. Afin d'améliorer les applications automobiles, les cartes sont de plus en plus enrichies et contiennent des informations plus variées. Par exemple, une carte routière peut contenir, en plus des informations géométriques et topologiques les informations suivantes : nom des routes, types et classes des routes, ponts et tunnels et des points d'intérêt (i.e. restaurants, école, hôpital, etc.).

Toutefois, l'enrichissement des cartes et l'ajout des détails conduisent à des situations ambiguës. Il est en effet beaucoup plus difficile d'atteindre l'objectif fixé si la carte contient beaucoup d'éléments semblables. La [figure III.1](#) donne un exemple de situation ambiguë pour un problème de sélection de route de circulation d'un véhicule. La position estimée étant en plus incertaine, la sélection de la voie de circulation basée sur un simple critère de calcul de distance sera par conséquent erronée. En effet, de nombreuses ambiguïtés peuvent être présentes dans ce genre de situation.

Une des problématiques de ces travaux de thèse est de gérer efficacement ces situations ambiguës pour localiser un robot dans une carte. À cet égard et en fonction des informations et des propriétés capteurs dont dispose le robot, une étude de la quantification de la confiance associée à l'estimation de la pose du robot sera présentée.

Différentes approches et méthodes de localisation sur une carte existent dans la littérature. Le lecteur intéressé pourra consulter la thèse de ([El Badaoui El Najjar, 2003](#)) ou l'article de ([Quddus et al., 2007](#)) pour un état de l'art sur les méthodes en ligne et hors ligne de localisation sur une carte en décrivent les avantages et les inconvénients de chaque méthode. Nous nous sommes inspirés de ces travaux pour classer les différentes méthodes de localisation sur une carte. Ces travaux restent pertinents malgré leur ancienneté. Dans ceux-ci, nous pouvons distinguer trois classes d'approches :

- les approches géométriques
- les approches topologiques
- les approches avancées

Les paragraphes suivants détailleront le principe de chacune de ces approches.

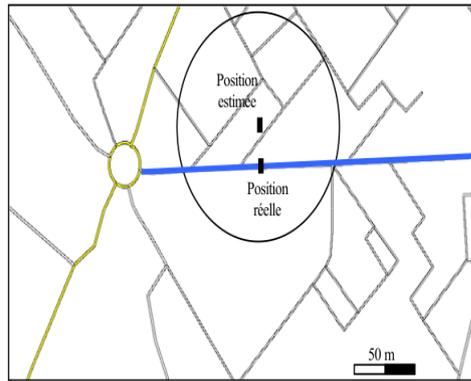


FIGURE III.1 – Exemple d’un problème de sélection de route où la position estimée est très proche d’une route qui est différente de la route de circulation du véhicule. (source : (El Badaoui El Najjar, 2003))

### III.2.1 Approches géométriques

C’est une approche métrique qui s’intéresse aux propriétés géométriques de l’environnement sans se soucier de comment les segments de la carte sont reliés les uns par rapport aux autres. Cette approche est souvent utilisée dans un contexte de sélection de la voie de circulation d’un véhicule (un problème de *road-matching*).

Dans ce type d’approche, nous distinguons trois méthodes :

- une méthode de type point à point : consiste à faire correspondre le point estimé au point le plus proche de la carte en utilisant un critère basé sur un calcul de distance. C’est le principe utilisé par des systèmes automobiles exploitant des données GPS. La performance de cette méthode est modeste (bien qu’elle soit simple à utiliser) dans la mesure où le point le plus proche n’appartient pas nécessairement à la voie réelle de circulation du robot. La figure III.2, illustre les limites de cette méthode. Le principe de cette figure consiste à faire correspondre des données GPS au nœud le plus proche appartenant à un segment modélisant une voie de circulation. La vraie trajectoire du véhicule correspond aux segments GE et EF. Les données GPS associées aux nœuds B et C sont par conséquent des fausses associations. Une recherche basée sur cette méthode peut être consultée sur (Quddus et al., 2003).
- une méthode de type point segment : l’idée consiste à déterminer le segment le plus proche à une estimée de type point. Bien que la performance de cette méthode est meilleure par rapport à la méthode précédente, elle présente toutefois des limites face aux similitudes et ambiguïtés comme montré par la figure III.1.
- une méthode de type arc à arc : c’est une méthode qui exploite les résultats des méthodes précédentes. La trajectoire du robot étant modélisée par un ensemble d’estimations successives, l’idée consiste à mettre en correspondance cette trajectoire avec un arc de la route. Dans (White et al., 2000), l’arc de la route à tester est segmenté de la même longueur que la trajectoire réelle du robot. Le chemin de circulation est donné par l’arc le plus proche de l’arc formé par la trajectoire du robot. Bien que cette approche ait été améliorée, elle est toutefois sensible à l’initialisation (Quddus, 2006).

Les approches géométriques font partie des premiers algorithmes de map-matching. Bien qu’elles semblent faciles à mettre en œuvre, elles manquent toutefois de précision et de confiance (représentant le degré de certitude dans l’estimation effectuée). D’autres méthodes plus fines sont données par les approches topologiques.

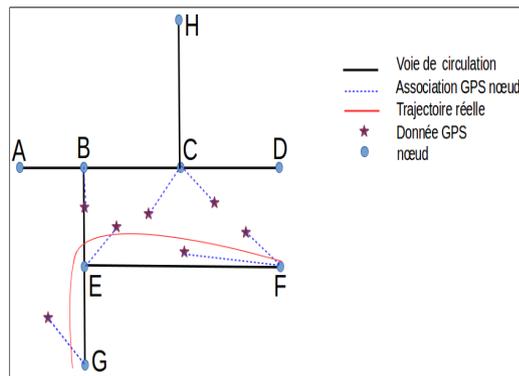


FIGURE III.2 – La localisation de type point à point est basée sur un critère de calcul de la distance entre le point estimé et le nœud le plus proche.

### III.2.2 Approches topologiques

La carte utilisée est dans ce cas, une représentation symbolique, non métrique, de l’environnement qui sera segmenté en des endroits distinctifs. Le graphe obtenu est constitué d’arêtes représentant les connexions entre les lieux. La figure III.3 donne un exemple de carte topologique. Les nœuds définissent des poses possibles du robot déduites des informations perceptuelles provenant du capteur et les arêtes sont les transitions d’une pose à une autre.

L’objectif de cette approche est de réduire le nombre de segments à traiter pour déterminer le segment de la carte où se trouve le robot. En fixant préalablement un seuil basé sur un calcul de distance, les segments ne les satisfaisant pas seront éliminés. En outre, en utilisant des caractéristiques liées à la route et au robot (position du robot, son orientation, l’orientation de la route, distance entre la pose estimée du robot et un segment de la route, les virages, etc.), cette méthode permet de réduire les ambiguïtés. Un exemple d’utilisation de cette méthode est donné dans (Greenfeld, 2002) où l’auteur utilise d’une part l’orientation du robot et celle de la route pour en déduire le degré de similitude sur l’orientation et d’autre part la distance de proximité entre les deux via la méthode point segment. Ensuite, un poids plus important sera attribué à la similitude d’angle qu’à la proximité pour sélectionner le segment le moins ambigu. Bien que cette méthode a été améliorée par (Quddus, 2006) en ajoutant une nouvelle caractéristique qui est la vitesse du robot, elle reste néanmoins insuffisante dans certaines situations comme par exemple à la sortie d’un carrefour (Greenfeld, 2002).

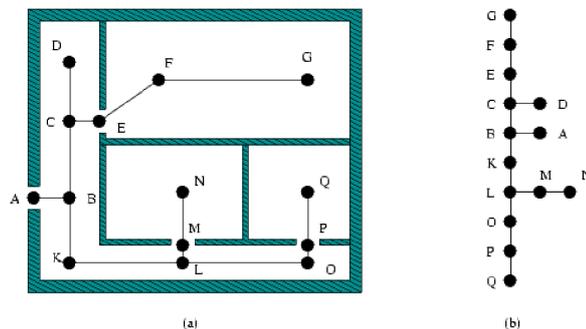


FIGURE III.3 – Un exemple de carte topologique 2D tiré de (Dufourd, 2005). a) Représentation métrique de l’environnement associée à des lieux topologiques reliés entre eux par des relations d’adjacence. b) Représentation purement topologique correspondante.

Les méthodes montrées jusqu’à présent illustrent bien la complexité liée à la tâche de localisation sur une carte. Bien que les approches topologiques offrent quelques améliorations par rapport aux approches géomé-

triques, elles sont toutefois mises en difficulté dans certaines situations. Cela nous amène à considérer d'autres méthodes plus robustes pour répondre à notre problématique.

### III.2.3 Approches avancées

Avec des cartes numériques plus détaillées, le robot est confronté à plusieurs éléments qui rendent la tâche de localisation sur une carte plus complexe. Parmi ces éléments nous trouvons :

- la carte peut présenter elle aussi des imprécisions qui sont générées lors des étapes de cartographie et de numérisation.
- avec des cartes de plus en plus détaillées où les liens sont condensés et très rapprochés, des situations ambiguës seront plus fréquentes comme par exemple la présence de routes parallèles très proches. Suite à ces configurations ambiguës que présente le milieu de navigation du robot, ce dernier risque d'effectuer des mauvaises associations (entre les données capteurs et les données issues de la carte) qui vont par la suite dégrader la performance du système de localisation. Le robot pense être à un endroit très similaire à celui de sa vraie position.
- suite aux changements permanents du réseau routier (ajout d'un rond point par exemple), les cartes déjà construites peuvent manquer de conformité avec la réalité.
- les imprécisions des données capteurs utilisées pour mettre à jour l'état du véhicule ajouteront de l'imprécision à l'estimée retournée.

Pour ces raisons, des approches avancées qui tiennent compte de l'ensemble de ces éléments sont nécessaires afin d'assurer une localisation précise et robuste face aux ambiguïtés. Nous désignons par *approches avancées* des algorithmes basés sur différents outils théoriques. Les paragraphes suivant donneront un aperçu de différentes méthodes de localisation sur carte qui existent dans la littérature.

#### III.2.3.1 Localisation sur carte en utilisant l'approche de Dempster-Shafer

Créée par Dempster en 1967 (Dempster, 2008) puis reprise par Shafer (Shafer, 1976), cette approche porte aussi le nom de la théorie des croyances ou la théorie de l'évidence. Elle est connue par son efficacité à modéliser explicitement, en plus de l'imprécision, l'incertitude (aussi connue sous le terme de *confiance*) à partir d'outils mathématiques (des fonctions de masse). En utilisant la notion de *croyance* et en calculant la probabilité d'un événement, le but de cette méthode est de combiner des connaissances pour la prise de décision. En effet, cette méthode présente deux niveaux : un niveau crédal pour la modélisation et la combinaison des données et un niveau pignistique pour la prise de décision (Chahine, 2016). Enfin, du fait de sa gestion de l'ambiguïté et des conflits entre les sources, elle est souvent utilisée pour des applications de classification et d'estimation.

Dans (Frittella et al., 2020), une présentation détaillée du formalisme mathématique de cette théorie est donnée. Aussi, dans (Mehranian et al., 2018) un exemple applicatif dans un cadre de la fusion d'informations pour la détection et localisation des incidents de circulation est présenté. La partie suivante de ce paragraphe exposera quelques recherches de map-matching dans le cadre de la théorie des croyances.

Dans (Gruyer et al., 2014), les auteurs ont développé un algorithme de map-matching basé sur la théorie des croyances prenant en compte les imperfections des données disponibles afin d'assurer le positionnement et le suivi d'un véhicule sur une carte routière. À l'aide d'une décision multi-hypothèses, cette méthode permet de gérer les conflits et les ambiguïtés des données. L'idée consiste dans un premier temps à convertir les données numériques en un référentiel symbolique. Ensuite, une combinaison de toutes les données décrivant l'état du véhicule et un segment donné est effectuée suivant des règles de combinaison multicritères de la théorie de

l'évidence. À partir de cette étape, une décision sur le segment de circulation du véhicule peut être prise. Toutefois, dans certaines conditions particulières, une solution conflictuelle ou bien ambiguë peut être retournée comme montré par la figure III.4. Dans ce cas, une méthode basée sur des tests multi-hypothèses est utilisée afin de résoudre ces deux problèmes et réduire l'influence de l'association ambiguë.

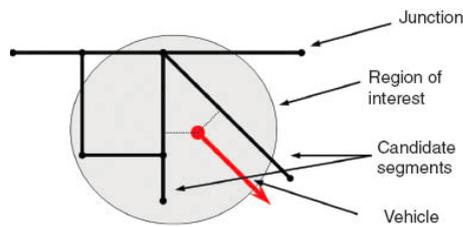


FIGURE III.4 – Map-matching dans un croisement (source : (Gruyer et al., 2014))

Une autre recherche présentée dans (Nassreddine et al., 2008), a développé un algorithme de map-matching qui fournit une estimation précise de la position du véhicule par rapport à une carte routière numérique en utilisant la théorie des croyances et l'analyse par intervalles. L'idée principale de l'algorithme proposé est de ne traiter que les connaissances d'intervalles acquises par les capteurs et d'utiliser la technique des hypothèses multiples pour sélectionner un ensemble de routes candidates dans une région rectangulaire. Cette méthode utilise simultanément, outre les méthodes d'estimation des erreurs bornées, des cadres géométriques et topologiques afin d'attribuer une assignation de croyance à un ensemble de routes candidates extraites d'une carte locale existante. La meilleure route candidate est finalement choisie en utilisant une règle de décision de la théorie des croyances. Cette technique s'avère pertinente pour traiter des situations de jonction de routes ou de routes parallèles.

Un exemple de SLAM présenté dans un formalise de la théorie des croyances peut être consulté dans (Jo et al., 2018). La méthode (SLAMCU) (pour Simultaneous Localization And Map Change Update) présentée utilise des cartes à haute définition (HD). Le but de cette approche est de détecter et mettre à jour les changements de la carte. La théorie de Dempster-Shafer est appliquée pour déduire les changements de la carte en se basant sur l'évaluation de l'existence des caractéristiques de la carte HD. Ensuite, une approche de filtre à particules Rao-Blackwellized (RBPF) est utilisée pour estimer simultanément la position du véhicule et mettre à jour le nouvel état de la carte. Les changements de carte détectés et mis à jour par la SLAMCU sont signalés à la base de données de la carte HD afin de refléter les changements de la carte et de partager les informations changeantes avec les autres voitures autonomes.

Si cette théorie semble très séduisante, elle présente néanmoins certaines limites (Martin, 2005) :

- une complexité croissante avec la taille du cadre de discernement (l'ensemble de toutes les décisions possibles)
- souvent appliquée avec un cadre de discernement discret, cette approche est généralement utilisée pour des applications de classification, de reconnaissance de formes et de détection.
- la modélisation des fonctions de masse doit être réalisée à partir d'une bonne connaissance des données ce qui peut être contrariant par rapport à l'aspect générique de l'application.

### III.2.3.2 Localisation sur carte en utilisant l'approche de la logique floue

Le principe de cette approche est de considérer des variables non booléennes (une valeur booléenne ne peut prendre que deux valeurs : 0 ou 1) pour présenter des connaissances données par des termes linguistiques vagues. Autrement dit, un système flou est un sous-ensemble où la fonction d'appartenance peut prendre des valeurs dans tout l'intervalle  $[0,1]$ . Pour bien illustrer son principe, nous pouvons considérer la définition d'un système flou donnée par (Martin, 2005) et l'exemple qui suit :

**Définition :** Un sous-ensemble flou  $F$  de  $S$  est défini par une fonction d'appartenance  $\mu_F$  qui à tout  $x$  de  $S$  associe une valeur réelle  $\mu_F(x)$  dans  $[0,1]$  représentant le degré d'appartenance de  $x$  au sous-ensemble flou  $F$ .

**Exemple :** considérons la règle de connaissance floue suivante : Si la distance entre la position estimée du robot et l'amer à détecter est grande, alors la possibilité de détecter l'amer est faible. Pour cette règle nous avons :

- variable d'entrée : la distance entre la position estimée du robot et l'amer à détecter
- sous-ensemble flou d'entrée : grande
- variable de sortie : la possibilité de détecter l'amer
- sous-ensemble flou de sortie : faible

Comme il s'agit de concepts vagues donnés par des sous ensembles flous, la vérité de sortie devient par conséquent une question de degré d'appartenance (El Mokhtari, 2015). Une étude plus exhaustive sur la théorie de la logique floue pourra être consultée dans (Marasini et al., 2016) (Mohan, 2010).

Les algorithmes de map-matching basés sur la théorie de la logique floue ont fait l'objet de plusieurs recherches. Nous citons quelques travaux à titre d'exemple.

Une recherche de map-matching basée sur cette théorie est présentée dans (Ma, 2015). L'objectif de ces travaux est de trouver le segment de la voie de circulation d'un véhicule. Les auteurs utilisent trois types d'informations : la position du véhicule donnée par un GPS, la vitesse angulaire du véhicule, et la structure topologique de la carte numérique. L'algorithme développé comprend trois étapes. Une première étape consiste à trouver les segments candidats compris dans une zone centrée autour de la donnée GPS. Ensuite, un calcul de l'orientation du véhicule ( $A_i$ ) et de la distance ( $D_i$ ) entre la donnée GPS et le segment de la route est effectué pour chaque segment candidat ( $R_i$ ). Une étape de fuzzification est menée avec les deux règles de connaissance floue suivantes : 1)  $A_i$  est faible et 2)  $D_i$  est faible. La fonction d'appartenance est ainsi une combinaison en un seul ensemble flou en utilisant les deux méthodes d'agrégation Max-Min. Finalement, la position du robot est donnée par une projection sur le segment retourné par la deuxième étape. Une recherche similaire est fournie dans (Gupta and Shanker, 2020)

Un autre exemple est présenté dans les travaux de thèse de (Yashar, 2016). La recherche proposée utilise une carte numérique, une caméra stéréo, un LiDAR et une centrale inertielle. L'algorithme commence par réaliser une odométrie visuelle stéréo pour estimer le mouvement de l'ego à six degrés de liberté (DoF), y compris les paramètres de rotation et de translation pour enregistrer les nuages de points. Ensuite, un algorithme GICP (Generalized Iterative Closest Point) est utilisé pour affiner l'estimation du mouvement par l'odométrie visuelle stéréo. La localisation du robot est ensuite donnée par un EKF en utilisant la vitesse avant, l'azimut donné par le LiDAR et les sorties de l'IMU. Cette solution est l'entrée d'un algorithme de map-matching (MM) basé sur la logique floue, qui prend les entrées imprécises et bruyantes et retourne des sorties nettes. L'objectif de la logique floue est d'identifier la bonne voie routière et de déterminer l'emplacement du véhicule sur la voie routière sélectionnée en améliorant la précision de la solution retournée par le filtre EKF. Une recherche similaire est donnée par (Chen and Chen, 2012).

D'autres travaux de map-matching basés sur la logique floue peuvent être consultés dans (Herrero-Pérez et al., 2010) (Tsukamoto and Uehara, 2012). Cependant, étant donnée la diversité des choix des fonctions d'appartenance dans la littérature, la plus grande difficulté liée à cette approche est le choix de cette fonction qui se fait souvent de manière empirique.

### III.2.3.3 Localisation sur carte en utilisant un filtrage particulaire

La localisation Monte Carlo MCL, également connue sous le nom de localisation par filtre à particules, est un algorithme permettant aux robots de se localiser à l'aide d'un filtre à particules (Thrun et al., 2001) (Thrun,

2002a) (Rekleitis, 2004). En exploitant une carte de l'environnement, l'algorithme estime la position et l'orientation d'un robot mobile. Le filtre à particules est utilisé pour représenter la distribution des états probables. Dans la technique classique, chaque particule représente un état possible du robot. L'algorithme commence généralement par une distribution uniforme et aléatoire des particules dans l'espace de configuration, ce qui signifie que le robot n'a aucune information sur l'endroit où il se trouve et suppose qu'il est également probable qu'il se trouve en tout point de la carte. Chaque fois que le robot se déplace, il déplace les particules pour prédire son nouvel état après le mouvement. Si le robot réussit une détection, les particules sont échantillonnées en utilisant une estimation bayésienne récursive. Les particules ayant un poids trop faible seront éliminées. Après mise à jour, les particules finiront par converger vers l'état optimal estimé.

Plusieurs travaux utilisant la localisation sur une carte à base de la méthode Monte Carlo ont été présentés dans la littérature. Dans (Mandel and Birbach, 2013), le système de localisation utilise un GPS, les informations cartographiques issues de carte et l'odométrie. La contribution principale de ce travail est donnée par l'extension des techniques de map-matching communément utilisées, dans la mesure où non seulement le réseau routier qui est exploité, mais aussi différents types d'entités cartographiées représentant des obstacles pour le robot.

La recherche donnée par (El Mokhtari et al., 2016) montre bien le principe de fonctionnement d'une méthode de map-matching basée sur un filtre à particules. Les données utilisées dans cette recherche sont les mesures de vitesse angulaire et de direction du robot. Le filtre à particules estime la position du véhicule sur une carte numérique en faisant correspondre la direction du véhicule avec la direction des routes sur la carte. Les particules représentent la distribution après mise à jour du vecteur d'état. Chaque particule a un poids proportionnel à sa probabilité. La position initiale du véhicule étant inconnue, l'ensemble initial de particules est donc généré le long de tous les segments. Dans l'étape de prédiction, chaque particule est propagée sur le segment de route à une distance  $d = v(t)\Delta(t)$  de sa dernière position.  $v(t)$  est la vitesse et  $\Delta(t)$  est le temps entre deux mesures. Lorsqu'une particule atteint la fin d'un segment, un choix au hasard est effectué parmi les segments reliés à celui en cours pour placer la particule. Dans l'étape de mise à jour, chaque particule est affectée d'un poids qui est la probabilité de la direction observée par rapport à la direction prédite. Par conséquent, une particule qui est placée sur un segment dont la direction est similaire à celle du véhicule reçoit un poids plus important et a une plus grande probabilité de rester le long des prochaines estimations.

Une autre méthode est donnée dans (Seow et al., 2017) en utilisant deux types de capteurs : un télémètre laser (si utilisé tout seul, cette technique est non performante lorsque des situations ambiguës se présentent (Milstein et al., 2002)) et un capteur Wifi (n'est pas très précis mais robuste). Cette méthode exploite la bonne précision du télémètre laser et la robustesse du capteur Wifi pour localiser un robot. L'avantage de cette méthode est d'utiliser une distribution à partir des données Wifi à l'initialisation au lieu d'une distribution uniforme. Cela permet une convergence plus rapide des particules. Une autre recherche très similaire est donnée par (Luo and Hsiao, 2018). Le même principe est aussi utilisé dans (Zhou et al., 2018) avec une double estimation de la pose à l'état initial en utilisant une image caméra et le scan d'un lidar. De même, dans (Ito et al., 2014), Ito et al. ont introduit l'approche W-RGB-D qui utilise les images des caméras RGB-D et les signaux Wifi pour la localisation de robot mobile via MCL. La technique de localisation par un réseau sans-fil a aussi été utilisée dans (Miyagusuku et al., 2016) où les auteurs ont montré que cette technique permet une localisation permanente du robot sous réserve que le robot reste toujours dans la zone de couverture du signal Wifi. Cette technique reste néanmoins une technique de localisation indoor qui est limitée à la précision de cette technique sans-fil sachant que l'erreur moyenne de cette localisation peut aller jusqu'à quelques mètres (Biswas and Veloso, 2010).

Un problème souvent étudié en rapport avec la localisation d'un robot mobile est le problème du robot kidnappé. Dans ce cas, le robot pense être à une position différente de sa vraie pose. Cette situation est critique pour un algorithme utilisant la technique classique MCL où le système échoue à localiser le robot. En effet, les particules sont d'abord réparties uniformément sur la carte, puis convergent vers la zone où le robot est probablement situé. Si le robot mobile est ensuite kidnappé dans une position arbitraire et que cette position n'est couverte par aucune particule, cette technique ne sera pas en mesure de localiser le robot (Bukhori et al.,

2015). Une solution à cette problématique est présentée dans un contexte de localisation coopérative où les auteurs dans (Khorshidi et al., 2016) ont proposé une solution en cas de perte de communication entre un robot maître et un autre suiveur. La solution proposée est une nouvelle variante du filtre à particules PF qu'ils ont nommée EAPF (pour Evolutionary Assisted Particle Filter). En calculant les erreurs d'estimations par rapport à la vérité terrain, les erreurs divergent en utilisant un filtre PF ou un EKF. Néanmoins, EAPF permet d'assurer des estimations précises de la pose du robot.

#### III.2.3.4 Localisation sur carte en utilisant des multi-hypothèses

Les méthodes de localisation sur une carte utilisant le filtre de Kalman constituent une classe d'approches largement utilisées (Kassas et al., 2020) (Asgar et al., 2019). En exploitant une série d'hypothèses restrictives, telles qu'un bruit gaussien et une incertitude initiale gaussienne, elles représentent les états postérieurs par des gaussiennes et exploitent un algorithme très efficace pour ajouter de nouvelles données de capteurs. Comme un seul filtre de Kalman ne peut estimer qu'une seule pose à la fois, cette approche standard est ensuite étendue pour résoudre le problème de la localisation globale en formant un filtre de Kalman multi-hypothèses (Jensfelt and Kristensen, 2001) (Kristensen and Jensfelt, 2003) (Hemanth et al., 2008). La méthode de localisation correspondante est généralement appelée localisation multi-hypothèses MHL (pour Multi Hypothesis Localization). L'idée de base est d'utiliser une distribution uni modale multiple pour se rapprocher de la distribution de probabilité multimodale globale (Arras et al., 2001).

Dans (Yun and Miura, 2007), une méthode de localisation de robot mobile sur une carte approximative (rough map) utilisant la vision stéréo a été présentée. Cette méthode utilise de multiples caractéristiques visuelles pour détecter et segmenter les bâtiments dans le champ de vision du robot. La carte brute est une carte imprécise avec de grandes incertitudes sur les formes, les dimensions et l'emplacement des objets afin qu'elle puisse être construite facilement. L'estimation de la pose du robot est assurée via un algorithme de MHL. Même en utilisant une condition de regroupement et de fusion des hypothèses plausibles, l'inconvénient majeur de cette méthode reste le nombre élevé d'hypothèses qui peut être généré.

Une solution palliative est présentée dans (Zhang et al., 2017). Les auteurs proposent une solution récursive de map-matching. Dans ce système, les hypothèses de l'algorithme sont organisées sous forme d'arbre. Chaque nœud d'hypothèse contient l'état du véhicule et une route possible et génère quelques nouvelles hypothèses en fonction de l'état actuel du véhicule (position et vitesse angulaire) et de la relation du véhicule avec la route (si le véhicule est proche d'une intersection, etc.). Ensuite, le poids pour chaque nouvelle hypothèse est calculé. Le poids se compose de deux parties : l'une est héritée de l'hypothèse mère et l'autre est calculée en tenant compte de la similitude entre l'état actuel du véhicule et la route correspondante. Lorsque chaque hypothèse principale a fini de générer une nouvelle hypothèse, la route de la nouvelle hypothèse ayant le poids le plus élevé représentera la route correspondante. De plus, pour éviter que le nombre de nœuds n'augmente de manière exponentielle, l'arbre doit être réduit, en éliminant le nœud ayant le plus petit poids. Avec ce type de structure, même si une mauvaise association s'est produite, la route correcte peut toujours être retrouvée en remontant dans l'arbre jusqu'à l'hypothèse correcte.

Cependant, cette technique souffre de quelques limites. En effet, elle est sensible aux modèles d'incertitudes et aux erreurs de linéarisation. La stabilité d'un EKF est difficile à garantir et dépend souvent de sa bonne initialisation et de la bonne modélisation du système et du bruit. Cela peut conduire à une estimation non fiable de la pose ce qui est problématique en utilisant un MHL dans des configurations présentant des ambiguïtés. En effet, afin de trouver la pose du robot, l'algorithme MHL va générer un ensemble d'hypothèses de poses possibles du robot selon les observations actuelles. Pour chacune de ces hypothèses, le MHL affectera un nouvel EKF pour l'estimation. Si nous considérons le cas avec une seule hypothèse qui se rapproche parfaitement de la pose réelle du robot et que cette hypothèse retourne une estimation peu fiable à cause des erreurs de modélisation (mauvaise modélisation du système et/ou des bruits), cela va altérer les autres hypothèses résultantes et entraîner un temps de calcul inutile. De plus, l'algorithme MHL est considéré comme compliqué à mettre en place surtout avec des systèmes qui sont fortement non linéaires (Liu et al., 2010).

### III.3 Récapitulatif

Nous avons présenté jusqu'à présent une étude de différentes approches de localisation sur une carte. Selon l'objectif de l'application, ces approches diffèrent les unes des autres selon le contexte théorique utilisé. Une brève description de chaque théorie et des exemples de recherches a été donnée. Toutefois, nous avons identifié quelques limites liées à ces approches. Parmi lesquelles nous citons :

- *imperfections liées à la carte* : les étapes de numérisation et de cartographie peuvent ajouter des imprécisions à la carte. Toutefois, une reconstruction non cohérente pourra dégrader la qualité de l'estimation des paramètres d'état du robot et engendrer une dérive par rapport à la vraie pose.
- *des cartes denses avec beaucoup de détails* : avec de telles cartes, des situations d'une grande ressemblance sont plus fréquentes et présentent par conséquent des ambiguïtés. Il devient difficile de faire correspondre correctement des éléments ressemblants (par exemple des murs ou des routes proches et parallèles) ce qui peut conduire à des mauvaises associations et par conséquent à des mauvaises estimations. En outre, les imprécisions des capteurs peuvent conduire aussi à des mauvaises associations.
- *une méthode tributaire des capteurs utilisés* : souvent la méthode appliquée est liée aux propriétés des données utilisées. Par exemple, un système basé sur un capteur GPS ne peut pas assurer une localisation continue suite aux phénomènes de masquage des satellites. Une solution alternative basée sur la fusion de capteurs doit être mise en place. D'autre part, la précision de l'estimation est très liée à la précision du capteur qui est souvent liée à son prix.
- *estimation de la pose initiale* : une problématique liée au problème de map-matching est la première association. Une fausse association à l'étape initiale peut donner lieu à des incohérences lors des étapes postérieures. Des solutions proposent un traitement particulier pour le lien initial et un autre pour les estimations suivantes.
- *correction des mauvaises associations* : lorsque des mauvaises associations ont eu lieu, le système de localisation doit être en mesure de revenir dans l'historique jusqu'à la première mauvaise association pour la corriger. Nous parlons dans ce cas d'un processus récursif.
- *non estimation de la confiance* : peu de travaux fournissent une métrique caractérisant la confiance que l'on a dans l'estimation retournée de la pose.

Compte tenu de ces limitations, il nous faut maintenant définir notre approche de localisation sur carte. Dans la partie suivante, une justification de nos choix sera présentée.

### III.4 Choix et conséquence

Au terme de cette analyse, nous rappelons les objectifs de nos travaux et les informations à notre disposition. Un choix sur la meilleure exploitation des informations et sur notre méthodologie sera exposé.

Les travaux menés dans cette thèse s'inscrivent dans le contexte de l'interprétation de scène et l'exploitation des données pour la localisation de robots mobiles dans un milieu urbain. L'objectif est de concevoir une architecture générique, modulaire et multi-sensorielle qui exploite au mieux les données contextuelles de l'environnement ainsi que les données fournies par les capteurs embarqués pour assurer une estimation optimale de la position du véhicule. La localisation que nous cherchons à assurer doit gérer simultanément les aspects imprécision et ambiguïté qui peuvent dégrader le bon fonctionnement du système de localisation. L'originalité de ces travaux tient à ce que l'on cherche à :

- exploiter une carte déjà existante composée de types d'amers de diverse nature.
- développer une approche de fusion, modulaire et multi-sensorielle et dans un contexte où on dispose de plusieurs capteurs potentiellement différents, permettant selon un objectif donné, liant précision et confiance, de sélectionner les informations à fusionner. Autrement dit, nous ne cherchons pas à analyser la totalité des données.
- développer une méthode de localisation sur une carte qui est générique et non tributaire de capteurs très spécifiques.
- proposer une méthode qui gère les ambiguïtés et qui est robuste à l'étape d'initialisation.
- prendre en compte explicitement les possibilités de mauvaise détection ou d'ambiguïté des capteurs et être en mesure de les corriger.
- fournir non seulement une estimation de la pose du robot mais aussi de la confiance dans cette pose.

### III.4.1 Choix de la carte

Différents types de cartes existent et sont utilisés pour assurer la localisation d'un robot mobile. Nous citons ici quelques exemples de cartes qui sont largement utilisées :

- les cartes routières : sont généralement des cartes topologiques et pouvant être métriques. Elles sont souvent utilisées pour assurer de la navigation grand public ou bien de la localisation de véhicule à base de capteur GPS.
- les grilles d'occupation : ce sont des cartes métriques représentant une discrétisation de l'environnement en un ensemble de cases. Chaque case est caractérisée par son état (vide ou occupée), sa position et la probabilité de son état. Cette dernière est mise à jour à chaque fois qu'une nouvelle mise à jour est réalisée. Cela signifie que la carte évolue au cours du temps. Ce type de carte est souvent utilisé pour indiquer au robot quelles zones sont navigables.
- les cartes absolues : différentes cartes grand public sont aujourd'hui disponibles, telles que Google Earth, Virtual Earth 3D ou OpenStreetMap. Ce sont des cartes absolues géoréférencées. L'inconvénient majeur de ce type de carte est la précision et la consistance (fiabilité) des données qui y sont stockées. Citons l'exemple du projet Map Quest qui offre la possibilité aux bénévoles d'enrichir ou mettre à jour des cartes OpenStreetMap. La précision et la fiabilité des données sont donc hétérogènes et fortement liées aux types de capteurs utilisés.
- les cartes sémantiques : une carte sémantique est une représentation de haut niveau de l'environnement. Contrairement aux types de cartes géométriques et les grilles d'occupation qui ne se soucient pas de la description des objets de la carte, une carte sémantique fournit en plus des caractéristiques spatiales, des données supplémentaires sur la nature des objets et leurs utilités. Ces données seront exploitées par n'importe quel type de capteur. Par exemple, un poteau ne sera pas identifié comme un cylindre vertical avec une certaine hauteur et un certain diamètre mais plutôt comme un poteau d'éclairage ou bien un feu de signalisation et cela soit en utilisant une caméra ou bien un lidar. Notre approche se base sur ce type de carte.

En fait, le choix de la carte que nous utilisons dans nos travaux est directement issu des travaux antérieurs de l'équipe ((Aynaud, 2015) et (Delobel, 2018)). Dans Aynaud, une méthode de localisation de robot mobile

sur une carte supposée exacte et parfaitement connue est proposée. Cette carte est utilisée comme donnée a priori. Dans Delobel, l'approche proposée exploite une carte incomplète et imprécise afin de se confronter à des situations plus proche de la réalité. La carte qu'ils ont utilisée est une carte sémantique contenant des murs.

Par conséquent, nous proposons de reprendre cette méthode et d'y apporter nos contributions. Le type de carte qui sera donc utilisé dans nos travaux est une carte sémantique avec différents types d'amers dans un système de coordonnées terrestres absolues. Cela répond à notre objectif de méthode *multi sensorielle* et *générique*. Dans l'étude de la littérature scientifique que nous avons présentée, nous remarquons que les méthodes proposées sont liées aux types de capteurs utilisés et aux formalismes décrivant le monde. La méthode que nous proposons autorise l'utilisation de différents types d'amers de la carte et par la suite l'exploitation de différents types de capteurs extéroceptifs. La solution étant générique et modulaire, l'ajout ou la suppression d'une modalité capteur n'affectera pas le fonctionnement général de notre algorithme mais aura des conséquences sur la qualité des résultats obtenus.

Les travaux antérieurs utilisaient néanmoins un seul type de capteur (un télémètre laser) pour détecter des murs de la carte. Cependant, l'avantage qu'offre ces recherches basées sur l'utilisation d'une carte sémantique est la possibilité de détecter plusieurs amers différents de manière à pouvoir fusionner des mesures provenant de diverses modalités capteur, ce qui correspond en fait à notre deuxième contribution.

Toutefois, la localisation sur une carte nécessite une prise en compte des possibilités de mauvaises détections suite à des phénomènes d'ambiguïtés. Ainsi, nous proposons une méthode permettant de gérer les ambiguïtés. Pour cela, nous utilisons d'une part un critère de sélection basée sur l'apport informationnel de la donnée à exploiter et d'autre part en mettant en oeuvre un processus de retour en arrière pour corriger les éventuelles mauvaises détections.

### **III.4.2 Choix de l'approche**

L'application proposée doit s'adapter aux conditions environnementales et exploiter d'une manière optimale les ressources à disposition. Par exemple, lorsque l'utilisation d'un GPS s'avère impossible dans un environnement fermé, notre algorithme devra trouver une autre modalité capteur à utiliser (un lidar par exemple ou une caméra) pour assurer la continuité de la tâche de localisation. C'est une approche multisensorielle et générique que nous voulons mettre en place.

À la différence d'un grand nombre de travaux de localisation qui existent, nous ne nous contentons pas d'une estimation précise de la position du robot. Comme nous allons le voir dans le chapitre III, cette estimation doit être justifiée. Une métrique décrivant la confiance que l'on a sur cette estimation doit être donnée. Cette fonctionnelle est fortement altérée par les ambiguïtés et le problème d'associations de données qui en découle. Pour cela, notre application doit être performante face aux ambiguïtés ainsi qu'aux problèmes d'association. L'approche Top-Down semble être la plus appropriée à ce genre de problème. En effet, pour ce type d'approche, nous ne cherchons pas à analyser la totalité des données, seule la donnée la plus pertinente sera exploitée. Cela nous évitera un nombre important d'associations de données et nous permettra de mieux gérer les ambiguïtés de l'environnement.

Comme nous partons d'une carte existante dont la position des amers à détecter est supposée connue, le cadre d'une approche Bottom-Up ne se présente pas. Cette dernière serait utile si nous ne connaissons pas l'existence ou la position des objets à détecter ou si nous cherchons à construire ou bien à enrichir la carte, ce qui n'est pas notre cas ici.

Pour ces raisons, il semble judicieux d'opter pour une approche Top-Down permettant selon le contexte de sélectionner la meilleure action à effectuer (comme détaillé dans le premier chapitre) grâce à son atout de focalisation sur l'amer recherché lors de la phase de détection.

### III.5 Bilan

L'intérêt de cette dernière partie est de positionner nos travaux par rapport aux recherches de la littérature que nous avons exposées. Plusieurs méthodes montrées ont traité le problème de localisation d'un robot sur une carte. Ces méthodes diffèrent les unes des autres et répondent à des objectifs fixés par les auteurs et ne satisfont pas nécessairement les nôtres.

Une des problématiques de ces travaux de thèse est de gérer efficacement les situations ambiguës pour localiser un robot dans une carte. Outre la précision de l'estimation de la position retournée, cette estimation doit être en plus fiable. Pour cela, une première contribution de nos travaux est l'utilisation de deux critères pour caractériser *conjointement* la qualité de l'estimation de la pose du robot qui sont la précision (définie par une zone d'incertitude autour de la pose estimée) et la confiance (la garantie qu'a notre estimation sur la présence du robot dans la zone d'incertitude).

Une deuxième problématique est liée à l'estimation de la position initiale du robot. La solution que nous proposons doit assurer l'estimation de toutes les positions du robot à tout instant de la même façon.

En outre, afin d'améliorer la robustesse de notre application, une prise en compte des ambiguïtés et des mauvaises détections est réalisée à travers un premier processus de focalisation et un deuxième processus de retour en arrière pour corriger les mauvaises détections qui sont toujours possibles.

Finalement, l'approche que nous proposons est multisensorielle (plusieurs modalités capteurs peuvent être exploitées) et générique (insensible au type de capteur utilisé).

La table III.1 résume les points que nous venons de citer en donnant un comparatif entre les travaux de la littérature et notre approche.

Travaux dans la littérature	Notre approche
- peu de recherches accordent une métrique pour quantifier la possibilité d'avoir détecté le mauvais amer et par conséquent avoir associé le mauvais amer de la carte à l'amer détecté	- notre objectif est double : un premier objectif de précision et un deuxième de confiance
- sans utilisation d'a priori, l'estimation de la position initiale d'une part et des positions postérieures d'autres part sont réalisées par deux techniques différentes	- aucun traitement spécifique pour déterminer la première pose
- le problème d'association de données se poserait de manière plus importante pour les approches ascendantes	- une approche descendante, permet de sélectionner selon le contexte, l'information la plus pertinente et la moins ambiguë
- les méthodes de localisation sont liées aux capteurs utilisés et sont contraintes par la disponibilité de ceux ci	- notre approche est une approche générique et multi-sensorielle qui se déroule de la même manière pour toute modalité capteur

TABLE III.1 – Table comparative entre les travaux dans la littérature et l'approche que nous proposons

### III.6 Conclusion

Pour résumer, l'objectif principal des travaux cités dans la littérature est d'assurer une localisation continue et précise de la pose du véhicule en dépit des ambiguïtés. Néanmoins, deux aspects très importants ne sont pas vraiment abordés à savoir : l'exploitation de la richesse des données de la carte de l'environnement et la fiabilité

que l'on a sur l'estimation de la pose du robot retournée.

La diversité des informations stockées dans la carte, nous incite à utiliser différentes modalités de capteurs pour profiter au mieux de la richesse de ces informations. C'est dans un cadre multi sensoriel, que nos travaux se positionnent et c'est en cherchant la meilleure méthodologie qui s'adapte à cette idée que nous nous sommes confrontés au problème des ambiguïtés et la similitude des données dans la carte. Dans le chapitre suivant, nous introduisons l'approche que nous avons développée en déclinant nos principales contributions.

# Chapitre IV

## Description de l'architecture de fusion proposée

### Sommaire

---

IV.1	Introduction	62
IV.2	Algorithme	62
IV.2.1	Principes généraux	62
IV.2.2	Notion de triplet perceptif	63
IV.2.3	Objectifs	64
IV.2.4	Objectif de précision	64
IV.2.5	Objectif de confiance	66
IV.2.6	Critère unifiant de précision et confiance	68
IV.3	Critère de sélection d'un triplet	68
IV.3.1	Disponibilité du triplet	68
IV.3.2	Gain en précision escompté	69
IV.3.3	Gain en confiance	70
IV.3.4	Pertinence globale	71
IV.3.5	Bilan	73
IV.4	Principe de focalisation et détection	73
IV.5	Mise à jour	74
IV.6	Processus de retour en arrière	75
IV.6.1	Principe	75
IV.6.2	Algorithme récursif développé	75
IV.6.3	Exemple	77
IV.6.4	Bilan	78
IV.7	Conclusion	79

---

## IV.1 Introduction

Nous allons maintenant présenter, dans ce chapitre, en se référant à la bibliographie précédente, notre approche descendante proposée qui a été appliquée à la localisation d'un robot mobile à partir de mesures issues de différentes modalités capteurs et d'une carte sémantique de l'environnement. Cette carte contient l'emplacement précis des amers dans le repère du monde. Quatre critères principaux doivent être assurés continuellement par notre algorithme :

1. une bonne précision de l'estimation de la pose du robot
2. une estimation intègre (la garantie qu'a notre estimation sur la présence du robot dans l'endroit où on le pense être) et, a minima, la confiance en l'estimation proposée
3. une fusion d'informations économe qui, au lieu de fusionner un grand nombre d'informations, préfère sélectionner en amont les plus prometteuses au vu d'un objectif donné
4. être en mesure de corriger une mauvaise estimation de la pose du robot par une phase de retour arrière sur la base d'un critère de confiance mis à jour de manière continue

## IV.2 Algorithme

### IV.2.1 Principes généraux

Afin de répondre à la problématique posée, nous proposons une méthode de type Top-Down, qui en fonction d'un objectif fixé et de l'état courant du robot, sélectionne la source ayant le plus grand apport informationnel. L'objectif final est fixé préalablement. Il peut être un objectif de précision ou de confiance comme il peut être une combinaison des deux. Il est en fait fixé en fonction de l'application finale envisagée. Comme il s'agit dans notre cas d'une application de localisation d'un robot mobile en environnement urbain, une précision centimétrique est ainsi nécessaire (par exemple inférieure à 10 cm) avec une confiance de l'estimation supérieure à 85%.

Le principe de l'approche descendante consiste à exploiter au mieux les informations dont dispose le robot. Le robot est équipé de capteurs extéroceptifs utilisés pour retourner des mesures par la voie de détection des éléments de la carte géo-référencée. Ces éléments ayant des positions connues permettront de retrouver la pose du robot. Pour cela, nous avons choisi d'utiliser un formalisme bayésien que ce soit pour le calcul de la pose ou de la confiance en celle-ci. Nous expliquerons dans la suite de ce chapitre l'utilité de ces deux outils mais l'objectif sera de gérer en parallèle la précision (donnée par un filtre bayésien, de type Kalman ou autre) et la confiance dans l'estimation de la pose (donnée par le réseau bayésien). La problématique est alors la suivante : dans une configuration donnée, quel amer le véhicule doit-il essayer de détecter afin de répondre au mieux aux objectifs de précision et confiance ? Avec quel capteur ? Avec quel détecteur ? Une fois la détection est effectuée, comment interpréter le résultat retourné par le détecteur ?

Dans le cas de cette approche, l'idée est à la fois de fournir une estimation  $\hat{X}_{rw}$  de la pose réelle du robot  $X_{rw} = (x_{rw}, y_{rw}, \theta_{rw})^T$  et de la covariance associée  $\mathbf{C}_{X_{rw}}$  (qui représente l'incertitude de l'estimation) mais aussi de fournir une idée de la confiance  $\mathbf{P}$  en cette estimation. Ainsi, initialement, la position réelle du robot est supposée être une variable aléatoire inconnue donnée par :

$$X_{rw} \sim \mathcal{N}(\hat{X}_{rw}, \mathbf{C}_{X_{rw}})$$

Ainsi, plutôt que modéliser la fonction de vraisemblance de la localisation par exemple par un ensemble de particules, nous ferons l'hypothèse d'une estimation de pose Gaussienne mais avec une confiance dans celle-ci. L'idée sera ensuite éventuellement de remettre en cause celle-ci selon sa vraisemblance  $\mathbf{P}$  actualisée au fur et à mesure de détections de plus en plus focalisées. En effet dans le cas de la robotique mobile, il est utile d'avoir une représentation de l'estimation de pose qui soit monomodale : un système de commande doit, quoiqu'il en soit, n'avoir qu'une seule estimation définitive pour choisir une action à mener. Dans ces travaux nous considérons donc une estimation de pose et nous y associons la notion de confiance (paramètre  $\mathbf{P}$ ). Dans une

hypothèse multimodale (par exemple issue d’ambiguïtés de détection), nous traitons l’hypothèse la plus crédible, puis nous remettons en cause celle-ci plus tard. Par exemple, nous nous engageons dans une rue avec un doute en tête sur le fait que cette rue soit la bonne jusqu’à ce qu’une détection nous confirme (la bonne rue) ou nous infirme l’hypothèse (nous n’avons pas pu identifier la poste dans cette rue).

Pour résumer, nous désignons par  $E_{rw} = \{\hat{\underline{X}}_{rw}, \mathbf{C}_{X_{rw}}, \mathbf{P}\}$  l’état global du robot. L’algorithme que nous proposons est présenté par la figure IV.1. Il comprend quatre étapes :

- une évaluation de l’objectif final (précision et confiance données)
- une sélection de la meilleure action de perception à assurer pour répondre à cet objectif
- une perception de l’environnement
- une mise à jour de l’état global du robot : la pose estimée, la covariance associée ainsi que la confiance

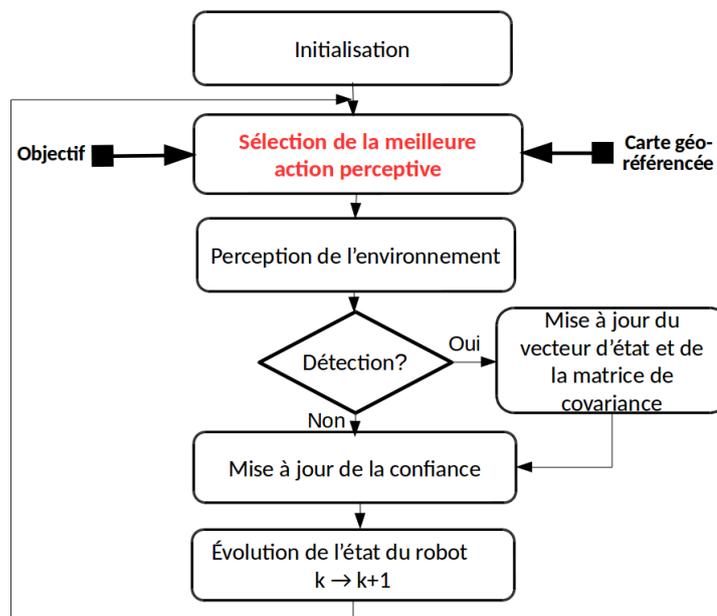


FIGURE IV.1 – Algorithme de localisation

## IV.2.2 Notion de triplet perceptif

Pour exploiter au mieux la variété des informations dont dispose le robot, à savoir les différentes modalités capteurs extéroceptifs (caméra, lidar, GPS, etc.) et les données de la carte, nous avons utilisé dans nos travaux la notion de triplet perceptif qui a été déjà introduite dans des travaux précédents comme (Aynaud et al., 2014a).

Un triplet perceptif  $T = \{A, D, C\}$  est l’ensemble formé par :

- un amer ’A’ : de position connue dans la carte, utilisée ici comme une donnée a priori
- un détecteur ’D’ : représente la partie software du triplet
- un capteur ’C’ : représente la partie hardware du triplet et dépend du robot (ouverture, résolution, portée minimale et maximale, etc.)

Par exemple, nous définissons le triplet *Lidar – Poteau* pour détecter des poteaux de la carte. Le triplet perceptif est ainsi donné par l’ensemble composé d’un :

- capteur : un télémètre laser (LiDAR)
- détecteur : un détecteur de demi cercle

— amer : un poteau

Supposons qu’il existe dix poteaux autour du robot qu’il pourra détecter. De ce fait, chaque mesure retournée par le capteur (dans ce cas une mesure de distance et d’angle d’orientation du poteau par rapport au capteur) sera différente l’une de l’autre. Pour cela, si nous considérons qu’un seul type de capteur et de détecteur sont utilisés pour détecter un poteau, nous pouvons par conséquent définir dix triplets perceptifs différents donnés par  $T_i = \{A_i, D, C\}$  avec  $1 \leq i \leq 10$ . Ensuite, comme nous le montrerons dans la suite de ce chapitre, c’est grâce au critère de sélection que nous allons définir dans la partie IV.3, que le système de localisation choisira le triplet qui ramènera le plus d’information conformément à notre choix d’approche descendante. Différents triplets ont été ajoutés dans nos travaux. Une description de ces triplets peut être trouvée dans (Ladhari et al., 2019).

### IV.2.3 Objectifs

Comme cela a été évoqué dans le chapitre précédent, la tâche principale de notre algorithme est la localisation d’un robot mobile. À la différence de plusieurs recherches, nous ne nous contentons pas d’une localisation précise. Cette estimation doit être en plus *justifiée*. C’est le degré de confiance que l’on a sur cette estimation qui doit être en plus calculé. Pour cela, notre objectif sera double : un objectif de précision d’une part dans la mesure où nous cherchons à assurer rapidement une estimation précise de la pose du robot, et d’autre part un objectif de confiance (ou de fiabilité) justifiant cette estimation que l’on a. Dans les paragraphes qui suivent, nous précisons plus en détails ce que l’on entend par objectifs de précision et de confiance.

### IV.2.4 Objectif de précision

Dans ce paragraphe, nous cherchons à illustrer l’objectif de précision que nous voulons assurer. En effet, deux scénarios peuvent se présenter : un premier scénario en utilisant des triplets de la même nature (par exemple deux triplets pour détecter un amer de type mur) et un deuxième pour des triplets de natures différentes (par exemple un triplet pour détecter un poteau et un autre utilisant une mesure GPS). Nous montrerons pour chacun de ces scénarios, l’apport de précision que nous pouvons avoir.

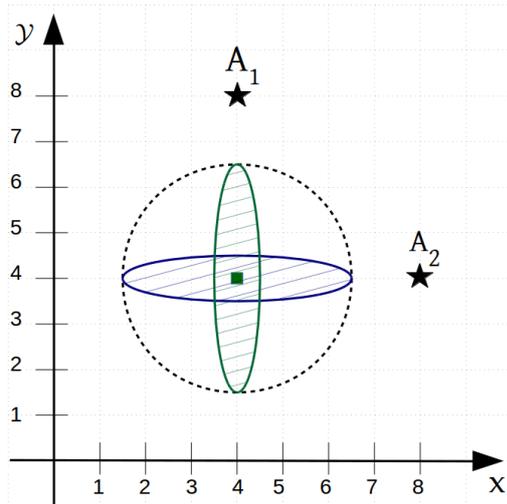
Dans l’ensemble de nos travaux, nous avons utilisé des estimateurs bayésiens en exploitant des lois normales. Le vecteur d’état de la position du robot que l’on essaye d’estimer est toujours accompagné d’une matrice de covariance. Cette matrice peut être représentée graphiquement par une hyper-ellipsoïde de même dimension que le vecteur d’état correspondant à un seuil précis de la densité de probabilité.

La position estimée du robot est définie par le vecteur d’état  $\hat{X}_{rw} = (\hat{x}, \hat{y}, \hat{\theta})^T$ . La position réelle est par contre inconnue, les coordonnées  $x$ ,  $y$  et  $\theta$  seront par conséquent considérées comme des variables aléatoires gaussiennes. La zone d’incertitude est donnée par une matrice de covariance  $\mathbf{C}_{X_{rw}}$  modélisant l’incertitude sur la position estimée du vecteur d’état. Les composantes de cette matrice représentent les écarts-types sur les variables du vecteur d’état comme indiqué par l’équation IV.1. Afin d’améliorer la précision de l’estimation, l’objectif sera donc de minimiser la zone d’incertitude. Plus cette zone est réduite, plus l’estimation est précise. On notera que le déterminant de cette matrice portera l’information d’imprécision. Il correspond en effet au volume de l’ellipsoïde de probabilité.

$$\mathbf{C}_{X_{rw}} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_\theta^2 \end{pmatrix} \quad (\text{IV.1})$$

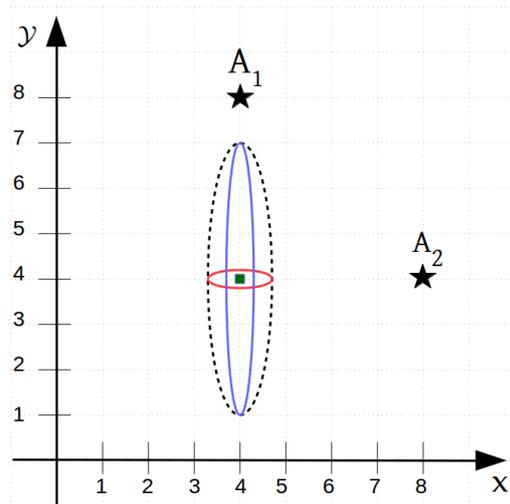
#### IV.2.4.1 Scénario 1 : apport de précision suivant une direction donnée

Pour ce scénario, considérons les deux représentations 2D de la figure IV.2. La position initiale du robot est donnée par  $\hat{X}_0 = (4, 4, 0)^T$  avec deux matrices de covariance différentes :  $\mathbf{C}_{X_a}$  pour la figure de gauche et  $\mathbf{C}_{X_b}$  pour celle de droite. Deux triplets  $T_1$  et  $T_2$  sont disponibles pour détecter deux amers  $A_1$  et  $A_2$  en utilisant



(a) Une amélioration de précision potentielle après mise à jour identique dans les deux directions  $x$  et  $y$  en partant d'une incertitude initiale donnée par la matrice  $C_{X_a}$

$$C_{X_a} = \begin{pmatrix} 2.5^2 & 0 & 0 \\ 0 & 2.5^2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



(b) Une meilleure amélioration de la précision suivant la direction  $y$  par le premier triplet après mise à jour et en partant d'une incertitude initiale donnée par la matrice  $C_{X_b}$

$$C_{X_b} = \begin{pmatrix} 0.7^2 & 0 & 0 \\ 0 & 3^2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

FIGURE IV.2 – Apport de précision suivant une direction donnée pour deux matrices de covariance initiales différentes. L'ellipse en pointillé noire représente la covariance initiale a priori avant mise à jour.  $A_1$  et  $A_2$  sont deux amers de la carte de positions connues et les ellipses en traits pleins sont les covariances estimées après détection.

la même configuration de capteur et de détecteur, cela veut dire que les deux triplets se distinguent aux niveaux des amers. Chacun de ces deux triplets effectue une mesure différente qu'il utilisera pour estimer la position du robot. Nous montrons par cette figure, les attentes de précision que nous pouvons avoir après une mise à jour par la détection d'un amer.

Pour la figure IV.2a, la matrice de covariance initiale  $C_{X_a}$  est diagonale et d'écart-types identiques sur  $x$  et  $y$  (l'erreur est donc isotrope dans toutes les directions). La zone d'incertitude initiale sera donc représentée par un cercle (le cercle noir pointillé). Après mise à jour, nous constatons que chaque triplet rapporte de l'information suivant une direction donnée (le premier triplet  $T_1$  en détectant l'amer  $A_1$  rapporte une information suivant la direction  $y$  et le deuxième suivant la direction  $x$  en détectant  $A_2$ ). Le déterminant de la matrice de covariance  $C_{X_a}$  vaut 39.062. Après mise à jour, chaque zone d'incertitude est caractérisée par une matrice de covariance ayant un déterminant égal à 1.5625. Cette égalité se traduit par un apport de précision identique pour les deux triplets mais suivant deux directions différentes. La sélection du meilleur triplet dépendra donc de l'objectif fixé (quelle direction nous voulons privilégier).

Par ailleurs, en partant d'une zone d'incertitude plus précise suivant la direction  $x$ , figure IV.2b (déterminant égal à 4.41), et en effectuant les mêmes détections que précédemment, le déterminant de matrice de covariance après mise à jour par le triplet  $T_1$  est égal à 0.019 et celui donné par le triplet  $T_2$  est de 0.81. C'est donc le triplet  $T_1$  qui permet d'améliorer considérablement la précision suivant la direction  $y$  (ellipse rouge). Ainsi, il sera judicieux de choisir ce triplet comme triplet qui ramène le plus d'information de précision (soit qui diminue le plus le déterminant de la matrice de covariance).

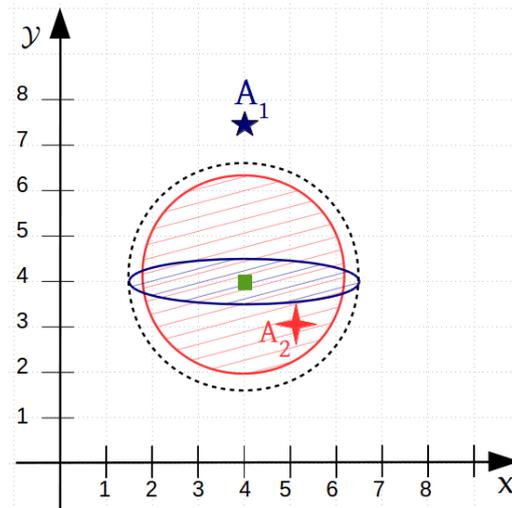


FIGURE IV.3 – Illustration de la différence d’apport de la précision entre deux triplets. En trait pointillé noir : la covariance initiale. En traits pleins : les covariances estimées après détection ; en bleu en utilisant le triplet  $T_1$  et en rouge en utilisant le triplet  $T_2$ .

#### IV.2.4.2 Scénario 2 : apport de précision dépendant de la précision du capteur

Dans l’exemple de la figure IV.3, nous montrons les attentes de précision pour deux triplets de nature différente : un triplet  $T_1 = \{A_1, S_1, D_1\}$  basé sur une mesure de distance et d’angle (par exemple un triplet pour détecter des poteaux) et un deuxième triplet  $T_2 = \{A_2, S_2, D_2\}$  basé sur une mesure de position (par exemple une mesure retournée par un GPS bas coût). Le déterminant initial est de 39.062. Après mise à jour, il passe à 1.5625 en utilisant le triplet  $T_1$  et à 23.426 en utilisant  $T_2$ . Cela peut être expliqué par le fait qu’un Lidar présente localement une meilleure précision qu’un système GPS classique. Le résultat de mise à jour apporté par le premier triplet est par conséquent meilleur (ellipse bleue pour le premier triplet et rouge pour le deuxième). Cela montre l’influence de la mesure effectuée et par conséquent du type de capteur utilisé. Dans cette situation, il est judicieux de choisir le premier triplet qui permet d’assurer une meilleure précision.

#### IV.2.5 Objectif de confiance

Nous venons de montrer les différentes attentes de précision que nous pouvons avoir après avoir effectué une détection et mis à jour la position du robot. Toutefois, un tel objectif de précision reste insuffisant selon nous. Nous avons souligné dans le chapitre précédent, la nécessité d’assurer une estimation précise et *fiable* de la pose du robot. Nous avons énuméré différentes situations qui peuvent dégrader la performance du système de localisation à savoir les occultations des amers, des situations ambiguës et les imprécisions liées aux capteurs et à la carte. Pour cela, nous avons choisi d’ajouter une métrique que nous appelons *confiance* permettant de privilégier les tâches les plus certaines et qui présentent le moins d’ambiguïté. En outre, elle permettra de faire appel à un processus correctif des erreurs (mauvaises associations de données, capteurs défectueux, détection d’outlier, etc.) comme nous le verrons dans la partie IV.6.

Afin de sélectionner le triplet qui conduira à une localisation la plus fiable, le calcul de la confiance est assuré par un réseau bayésien, que nous décrivons dans le chapitre qui suit pour divers types de triplets. Ce réseau bayésien permettra de :

- modéliser les événements qui peuvent avoir lieu, comme la possibilité d’une mauvaise association ou la présence d’un phénomène d’occultation
- utiliser des a priori
- prendre en compte la possibilité de détecter un outlier

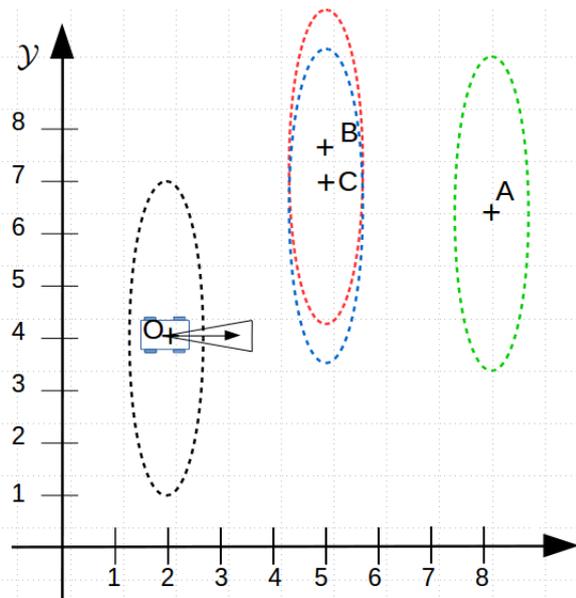


FIGURE IV.4 – Illustration du principe de sélection selon l’apport de confiance. L’ellipse en pointillé noire représente l’incertitude autour de la pose estimée du robot. Les points A, B et C représentent des amers. Les autres ellipses en couleur présentent la zone de recherche pour chaque amer.

— considérer les caractéristiques des capteurs, détecteurs et amers

En effet, ici, nous cherchons à donner une probabilité à l’estimation de pose or cette probabilité peut être déduite de beaucoup d’évènements (observabilité d’un amer, mauvaises performances des détecteurs, etc). Pour bien illustrer l’objectif de confiance que nous souhaitons avoir, considérons la figure IV.4. La pose estimée du robot est donnée par le point O avec une certaine incertitude représentée par l’ellipse en pointillé noire. Cela veut dire que le robot peut se trouver n’importe où dans cette zone. L’orientation du robot, donnée par la flèche noire, présente aussi une incertitude de  $10^\circ$  représentée par le cône en noir. Trois amers, de positions connues, sont disponibles dans la carte géo-référencée. Nous aurons par conséquent trois triplets différents. Toutefois, pour des raisons de simplification, nous considérons que le robot est équipé d’un seul capteur et détecteur. Les triplets se distinguent donc entre eux uniquement par leur amer.

Le principe consiste à chercher à détecter les amers de la carte. Connaissant les amers de la carte, nous pouvons déduire pour chacun une région de recherche (ROI : Region Of Interest). Nous représentons cette zone de recherche par une ellipse centrée autour de l’amer en question. Cette zone tient compte des incertitudes liées à la position estimée du robot comme nous le montrerons dans la partie IV.4. Ensuite, l’idée consiste à associer l’amer détecté avec la carte et finalement mettre à jour la position du robot ainsi que la covariance associée afin d’améliorer la précision de la localisation.

Dans les zones de recherche des deux amers B et C (ellipses rouge et bleue respectivement), ces derniers sont voisins. Pour une telle configuration, il y a un grand risque de les confondre. En plus, si la mesure utilisée est une mesure d’angle et étant donnée l’incertitude sur l’angle d’orientation du robot, cela augmentera le risque d’une mauvaise association. Nous aurons en effet une chance sur deux d’avoir associé correctement l’amer détecté à celui de la carte. Toutefois, l’amer A ne présente pas d’ambiguïté. C’est celui qui permettra plus naturellement d’améliorer la confiance de l’estimation. Par conséquent, c’est le triplet composé de l’amer A qu’il faut choisir pour éviter les ambiguïtés.

## IV.2.6 Critère unifiant de précision et confiance

Comme nous l'avons déjà évoqué, l'approche que nous proposons dans nos travaux est une approche descendante. Cela veut dire que les données utilisées seront sélectionnées selon un objectif fixé. C'est un objectif double que nous allons utiliser : un objectif de précision et un autre de confiance. Il est important de satisfaire ces deux objectifs ensemble dans la mesure où avoir une estimation très précise mais non fiable sera inutile et vice versa.

Dans le cadre d'un objectif multiple, la définition d'un critère regroupant tous les objectifs n'est pas aisée. Dans (Aynaud et al., 2014a) et pour un même contexte de localisation d'un robot mobile, le critère utilisé était de la forme  $\lambda P_i + \psi P_p$  où  $P_p$  et  $P_i$  sont les deux gains de précision et de confiance et  $\lambda$  et  $\psi$  sont des poids utilisés afin d'assigner une importance plus ou moins grande pour chacun des deux objectifs. Dans nos travaux, nous allons introduire la notion de gain entropique basé sur la théorie de l'information (voir annexe VIII.1) que nous détaillerons dans la partie suivante.

## IV.3 Critère de sélection d'un triplet

### IV.3.1 Disponibilité du triplet

Après une étape d'initialisation pendant laquelle l'état du véhicule est fixée et les ressources dont dispose le robot sont évaluées, une mise en œuvre des différents triplets est réalisée. L'étape suivante de sélection du meilleur triplet consiste à choisir le triplet perceptif le plus pertinent tant en précision qu'en confiance. Le terme de *Pertinence* sera ainsi utilisé par la suite.

Ce paragraphe s'intéresse à une tâche qui s'effectue juste en amont de l'étape de sélection du meilleur triplet. Cette tâche retourne le nombre de triplets qui va être considéré à cette étape afin d'en tirer le meilleur triplet. En effet, un critère basé sur la disponibilité du triplet doit être utilisé afin de réduire la complexité de l'algorithme et le temps de calcul. Toutefois, il est judicieux d'utiliser des triplets qui ont une meilleure probabilité de réussir une détection. Pour cela nous considérons deux phénomènes :

- la disponibilité du triplet dépend de ses caractéristiques intrinsèques : la position de l'amer étant connue via la carte et sachant l'estimation de la pose du robot, serait-il possible de détecter l'amer ? Étant donnée la distance de perception du capteur et son angle d'orientation, l'amer est-il visible ?
- la prise en compte de la répartition des amers dans la carte est importante dans la mesure où des éventuelles occultations peuvent avoir lieu : bien que l'amer recherché soit dans la zone d'observabilité du triplet, il peut être cependant occulté par d'autres amers. C'est ce que nous allons appeler l'observabilité extrinsèque du triplet basée sur l'exploitation des données a priori (puisque la disponibilité du triplet dépend du milieu extérieur).

Pour résumer, pour calculer la pertinence de chaque triplet, il est primordial de prendre en compte la disponibilité (l'observabilité) du triplet. Pour cela, considérons la figure IV.5. Soient  $N_a$ ,  $N_c$ ,  $N_d$  le nombre d'amers, de capteurs et de détecteurs, ainsi le nombre maximal de triplets dans le pire des cas est de  $N_a \times N_c \times N_d$ . Dans cet exemple de la figure IV.5, nous supposons que pour un seul type d'amer, un seul type de détecteur et de capteur lui est associé ce qui fait que d'une part, le nombre maximal de triplets que nous pouvons avoir est donné par le nombre des amers de la carte et d'autre part, les triplets se distinguent uniquement entre eux au niveau des amers. Dans cette illustration les amers à détecter sont des murs. Le robot est orienté vers la droite avec une incertitude sur l'angle donnée par le cône autour de la flèche représentant son orientation et une incertitude sur les deux directions  $x$  et  $y$  donnée par l'ellipse en pointillés. Le robot est équipé d'un capteur centré sur le robot et ayant un angle de vue allant de  $75^\circ$  à  $-75^\circ$  et une distance de perception maximale de  $5\text{ m}$ . La zone de perception du capteur lorsque le robot est au centre de l'ellipse est donc donnée par la zone verte. Un amer est considéré comme observable s'il est possible de le détecter en entier ou bien partiellement.

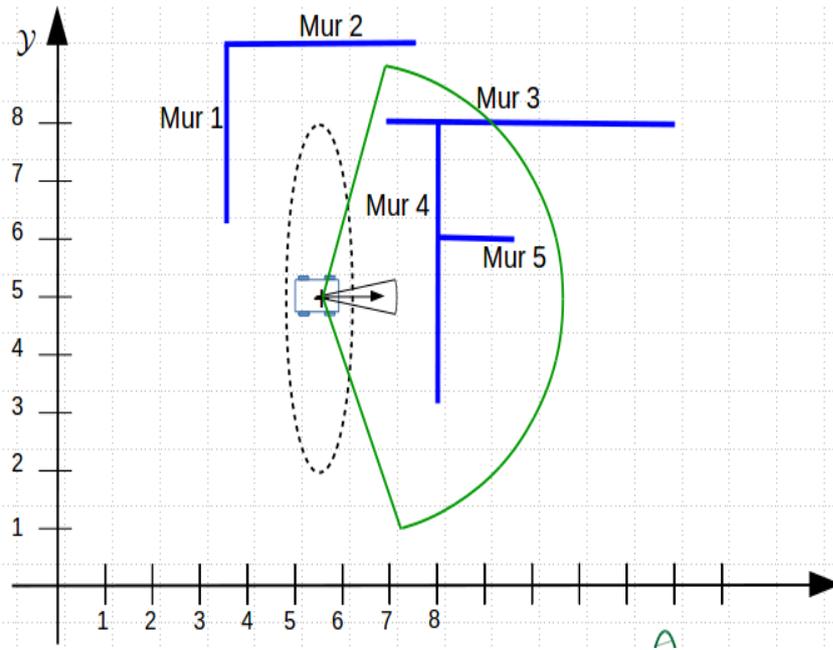


FIGURE IV.5 – Illustration du principe d'observabilité des triplets.

Dans cette illustration, chaque amer se distingue par les caractéristiques suivantes :

- mur 1 : ne sera jamais observable puisqu'il se trouve toujours derrière le robot. Il n'est donc pas utile de chercher à le détecter. Sa pertinence sera donc nulle
- mur 2 : n'est observable que si le robot se trouve dans la partie supérieure de la zone d'incertitude.
- mur 3 : n'est observable que si le robot se trouve entre la partie supérieure de la zone d'incertitude et la moitié supérieure de la partie inférieure de la zone d'incertitude.
- mur 4 : est toujours observable. Sa probabilité d'observation est par conséquent supérieure à celle du mur 2 et du mur 3.
- mur 5 : ne sera jamais observable car il est occulté par le mur 4. Il est donc inutile de chercher à le détecter.

Par conséquent, en se référant aux propriétés intrinsèques (types du capteur et de l'amer, état du véhicule, caractéristiques du capteur et du détecteur, position de l'amer, etc.) et extrinsèques (distribution des amers) du triplet, le nombre de triplets qu'il faut considérer dans cette situation est trois au lieu de cinq. Nous montrerons dans le chapitre suivant, comment la probabilité d'observation est calculée pour chaque type de triplet.

### IV.3.2 Gain en précision escompté

L'objectif de précision que nous voulons atteindre sera donné par une zone d'incertitude autour de la position du robot, par exemple : 5 cm sur  $x$ , 2 cm sur  $y$  et  $1^\circ$  sur  $\theta$ . Cette zone sera donnée par la matrice de covariance de l'estimation  $\mathbf{C}_{X_k}^-$  (la matrice de covariance a priori avant la mise à jour) ou  $\mathbf{C}_{X_k}^+$  (la matrice de covariance après une simulation de mise à jour). Dès lors, afin d'assurer la sélection du meilleur triplet, le choix effectué doit considérer l'apport en précision de chaque triplet amené par la détection de l'amer considéré. Le calcul du gain de précision consiste donc à effectuer une simulation de mise à jour de la pose du robot en utilisant le modèle d'observation et de mise à jour propre au triplet.

Comme montré dans l'annexe VIII.1.3, l'entropie  $\mathbf{H}_p$  d'une variable aléatoire normale multivariée  $x$  de dimension  $N$  et de matrice de covariance  $\mathbf{C}_{X_k}$  est donnée par :

$$\mathbf{H}_p = 1.4189.N + \frac{\ln(|\mathbf{C}_{X_k}|)}{2}$$

Comme nous chercherons à caractériser le *gain en précision* amené par la détection de l'amer considéré, nous calculerons ce gain en précision par la différence entropique entre l'entropie escomptée après mise à jour  $\mathbf{H}_p^+$  et l'entropie a priori  $\mathbf{H}_p^-$ .

$$\mathbf{G}_p = \mathbf{H}_p^+ - \mathbf{H}_p^- \quad (\text{IV.2})$$

Comme la dimension  $N$  est identique pour  $\mathbf{C}_{X_k}^-$  et  $\mathbf{C}_{X_k}^+$  :

$$\mathbf{G}_p = \frac{\ln(|\mathbf{C}_{X_k}^+|) - \ln(|\mathbf{C}_{X_k}^-|)}{2}$$

Soit :

$$\boxed{\mathbf{G}_p = \frac{1}{2} \ln \left( \frac{|\mathbf{C}_{X_k}^+|}{|\mathbf{C}_{X_k}^-|} \right)} \quad (\text{IV.3})$$

Après une détection, la précision sera améliorée (incertitude diminuée) ce qui se traduit par :  $|\mathbf{C}_{X_k}^+| < |\mathbf{C}_{X_k}^-|$ . Le triplet ayant la meilleure précision doit par conséquent présenter le gain  $\mathbf{G}_p$  le plus négatif possible.

### IV.3.3 Gain en confiance

Comme nous l'avons déjà évoquée, une métrique retournant la confiance que l'on a sur l'estimation est calculée. Cette fonctionnelle est caractérisée par le terme d'intégrité. Une estimation est dite intègre si la position réelle du robot appartient à la zone d'incertitude estimée (donnée pour un seuil fixé d'écart-type, par exemple pour  $1\sigma$ ) comme montré par la figure IV.6. Une mauvaise détection peut ne pas être l'unique raison d'un mauvais positionnement du robot : elle peut être due à la qualité du détecteur, l'observabilité de l'amer selon le point de vue considéré, le fait que le robot ne soit en effet pas à l'endroit attendu, etc. Le calcul de cette confiance est réalisé à l'aide d'un réseau Bayésien. Ce dernier est souvent utilisé pour formaliser les connaissances dans un environnement incertain et il est donc tout à fait approprié dans notre cas. Pour chaque type de triplet, un réseau bayésien approprié lui est associé. Nous nous contentons ici de le prendre pour acquis et nous le décrirons dans le chapitre V.

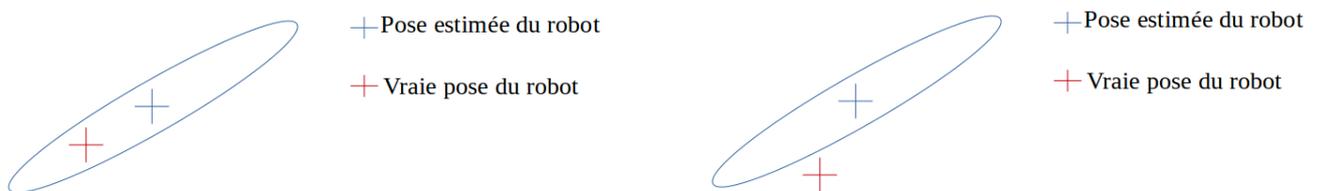


FIGURE IV.6 – Illustration de l'intégrité d'une estimation. La zone d'incertitude, représentée par une ellipse, est centrée autour de la position estimée. Dans la figure de gauche, la vraie pose du robot appartient à la zone d'incertitude estimée. L'estimation est donc intègre. Cependant celle de droite est non intègre puisque la zone d'incertitude n'englobe pas la pose réelle. On notera que cette notion implique un grand nombre d'estimation : une seule hors de la zone étant finalement possible bien que moins probable.

Un réseau bayésien contient un ensemble de nœuds représentant chacun un événement. Par exemple, l'événement *détection réalisée* (noté  $d_b$ ) sera modélisé par un nœud binaire (détection réussie ou non). Le calcul de la probabilité de ce nœud ( $P(d_b)$ ) peut être utilisée par inférence pour calculer des probabilités a priori. Il est en de même pour l'événement présentant l'intégrité de l'estimation. En effet, les deux événements binaires  $O_k^-$  (la localisation a priori à l'instant  $k$  est intègre) et  $O_k^+$  (la localisation a posteriori c'est-à-dire après mise à jour de l'état à partir de la détection réalisée à l'instant  $k$  est intègre) permettront de calculer la gain en confiance souhaité. Le gain en information peut être calculé en connaissant les probabilités  $P(O_k^-)$  et  $P(O_k^+)$ . Ces dernières seront extraites d'un réseau bayésien.

Sachant que les événements  $O_k^-$  et  $O_k^+$  sont binaires, l'entropie  $\mathbf{H}_c$  relative à un événement binaire  $O_k$  sera donnée par (voir annexe VIII.1.2) :

$$\mathbf{H}_c = - \sum_{O_k} P(O_k) \log_2(P(O_k)) \quad (\text{IV.4})$$

$$= -P(O_k = 0) \log_2(P(O_k = 0)) - P(O_k = 1) \log_2(P(O_k = 1)) \quad (\text{IV.5})$$

$$= -(1 - P(O_k = 1)) \log_2(1 - P(O_k = 1)) - P(O_k = 1) \log_2(P(O_k = 1)) \quad (\text{IV.6})$$

Ainsi si on note simplement  $P(O_k^- = 1) = P(O_k^-)$  et  $P(O_k^+ = 1) = P(O_k^+)$ , l'expression du gain en confiance  $\mathbf{G}_c$  sera donc :

$$\mathbf{G}_c = \mathbf{H}_c^+ - \mathbf{H}_c^- \quad \text{avec} \quad \begin{cases} \mathbf{H}_c^- &= -(1 - P(O_k^-)) \log_2(1 - P(O_k^-)) - P(O_k^-) \log_2(P(O_k^-)) \\ \mathbf{H}_c^+ &= -(1 - P(O_k^+)) \log_2(1 - P(O_k^+)) - P(O_k^+) \log_2(P(O_k^+)) \end{cases} \quad (\text{IV.7})$$

Pareillement à l'aspect précision, le triplet ayant la meilleure confiance doit par conséquent présenter le gain  $\mathbf{G}_c$  le plus négatif possible. Ainsi, pour un triplet donné,  $\mathbf{G}_c$  sera donc le gain espéré en confiance amené par la détection de son amer. On se rappelle ici, qu'il s'agit de sélectionner les amers pour choisir le plus prometteur.

### IV.3.4 Pertinence globale

La *pertinence globale* sera donc la somme des gains d'information. Cependant, dans le cas où la détection n'a pas été réalisée, aucune mise à jour de l'état u robot ne sera faite et le gain en précision est naturellement nul. Néanmoins, la confiance sera mise à jour. En effet, un succès de détection la fait augmenter (le robot est vraiment où il pense être) et le cas contraire (échec de détection) la fait baisser. Ainsi, la *pertinence globale G* (qui est en fait le *gain informationnel* amené par le choix de ce triplet) sera donnée par :

$$\mathbf{G} = [1 - P(d_b = 1)] \mathbf{G}_c + P(d_b = 1)[\mathbf{G}_p + \mathbf{G}_c] \quad (\text{IV.8})$$

Dans la relation de l'équation IV.8, nous constatons qu'il faudra donc que l'on prenne en compte la probabilité de la bonne détection  $P(d_b)$  de l'amer considéré. Pour cela, nous nous référerons au réseau bayésien de la figure IV.7. Le fonctionnement du réseau bayésien sera expliqué plus en détails dans le chapitre V dans la partie V.3.2.3. Nous nous contentons dans cette partie des événements qui sont devenus maintenant familiers modélisant chacun une hypothèse donnée, en particulier :

- $O_k^-$  : la localisation a priori à l'instant  $k$  est intègre
- $d_b$  : la détection est celle de l'amer recherché

Comme nous l'avons indiqué dans la partie IV.3.1, l'amer recherché doit être observable. Pour cela, l'événement *obs* (observabilité de l'amer) est ajouté au réseau bayésien et sa probabilité sera utilisée dans le critère de sélection. Cet événement est directement lié à l'événement  $O_k^-$ , par conséquent, la table des probabilités conditionnelles  $P(obs|O_k^-)$  sera donnée par la table IV.1. Il en est de même pour l'événement  $d_b$  dépendant de l'événement *obs*. La table des probabilités conditionnelles correspondante est donnée par la table IV.2. Les probabilités indiquées dans les deux tables seront expliquées dans le chapitre suivant car spécifiques à chacun des triplets considérés. Nous nous contentons dans cette partie des liens qui existent entre les différents événements du réseau bayésien.

$O_k^-$	$P(obs = 1 O_k^-)$
0	0
1	$P(obs O_k^- = 1)$

TABLE IV.1 – Table des probabilités conditionnelles  $P(obs|O_k^-)$

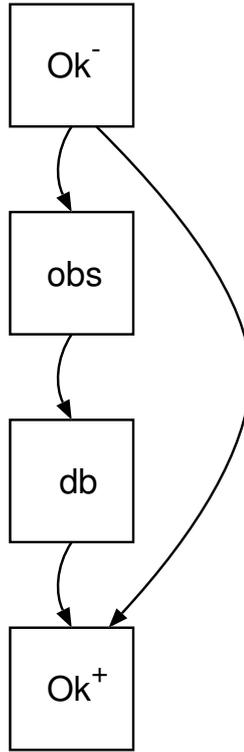


FIGURE IV.7 – Réseau bayésien utilisé dans le triplet lidar-mur prenant en compte le nœud  $d_b$  qui sera utilisé pour le calcul de la probabilité  $P(d_b)$

$obs$	$P(d_b = 1 obs)$
0	$1 - \alpha^{N_c+N_o}$
1	$1 - \alpha^{1+N_c+N_o}$

TABLE IV.2 – Table des probabilités conditionnelles  $P(d_b|obs)$

Avec :

- $\alpha$  : la probabilité de ne pas détecter l’amer alors qu’il existe dans la zone de focalisation (ROI)
- $N_c$  : nombre d’amers de la carte compatibles avec l’amer recherché
- $N_o$  : le nombre d’amers potentiellement détectables mais ne faisant pas partie de la carte (des outliers)

Ainsi, selon les deux tables, la probabilité de détection d’un amer sera donnée par :

$$\begin{aligned}
 P(d_b) &= P(d_b|obs = 0).P(obs = 0) + P(d_b|obs = 1).P(obs = 1) \\
 &= P(d_b|obs = 0). \left[ \overbrace{P(obs = 0|O_k^- = 0)}^1 P(O_k^- = 0) + \overbrace{P(obs = 0|O_k^- = 1)}^{1-P_{obs}} P(O_k^- = 1) \right] + \\
 &\quad P(d_b|obs = 1). \left[ \overbrace{P(obs = 1|O_k^- = 0)}^0 P(O_k^- = 0) + \overbrace{P(obs = 1|O_k^- = 1)}^{P_{obs}} P(O_k^- = 1) \right]
 \end{aligned}$$

Avec  $P_{obs} = P(Obs = 1|O_k^- = 1)$

En notant  $P(0_k^-) = P(O_k^- = 1)$  et en développant nous aurons :

$$P(d_b) = (1 - \alpha^{N_c+N_o}) [1 - P(O_k^-) + (1 - P_{obs})P(O_k^-)] + (1 - \alpha^{1+N_c+N_o}) [P_{obs}.P(O_k^-)]$$

Soit :

$$P(d_b) = (1 - \alpha^{N_c+N_o}) [1 - P_{obs} \cdot P(O_k^-)] + (1 - \alpha^{1+N_c+N_o}) [P_{obs} \cdot P(O_k^-)] \quad (IV.9)$$

### IV.3.5 Bilan

Jusqu'à présent, nous avons présenté le critère de sélection du meilleur triplet que nous avons utilisé dans nos travaux. Ce critère est basé sur la théorie de l'information pour calculer les *gains informationnels en précision et en confiance* pour chaque triplet. Ces deux gains seront ensuite fusionnés en un seul critère appelé *la pertinence globale du triplet* (ou bien le gain informationnel). La simulation de l'apport informationnel de chaque triplet permet de sélectionner le triplet le plus pertinent ayant le meilleur gain informationnel. C'est ce triplet qui permettra d'assurer une estimation optimale de la pose du robot : d'une part pour atteindre rapidement l'objectif de précision (et par conséquent une zone d'incertitude réduite associée à la pose estimée) et d'autre part pour assurer une estimation fiable et robuste face aux ambiguïtés.

La sélection du meilleur triplet s'inscrit dans le cadre de notre choix d'approche descendante. C'est seulement le meilleur triplet qui sera utilisé parmi tous les autres pour estimer et mettre à jour l'état du robot (la position avec la zone d'incertitude associée et la confiance). En outre et comme nous l'avons signalé dans le [chapitre II](#), une approche Top-Down permet de limiter le problème d'association de données qui se poserait d'une manière moins importante qu'avec une approche Bottom-Up et cela à travers le procédé de la détection dite active ou *focalisation* qui sera abordée dans la partie qui suit.

## IV.4 Principe de focalisation et détection

À cette étape, le critère informationnel décrit dans la partie précédente a été utilisé pour sélectionner le triplet le plus pertinent  $T_m = \{A_m, D_m, C_m\}$ . La prochaine étape consiste donc à détecter effectivement l'amer  $A_m$  en utilisant le capteur  $C_m$  et le détecteur  $D_m$ . Cette étape est en effet facilitée par l'utilisation des informations a priori et des ressources dont dispose le robot via la technique de focalisation.

La technique de focalisation est basée sur l'exploitation des connaissances comme la distance estimée entre le détecteur et l'amer sélectionné ainsi que la connaissance des caractéristiques intrinsèques du capteur. En effet, à partir des données du capteur et connaissant la pose géoréférencée de l'amer, le détecteur sait approximativement la distance entre le robot et l'amer à détecter. Nous définissons cette zone de recherche par la zone de focalisation ou ROI (pour Region Of Interest). Dans cet ordre d'idée, cette focalisation considère un certain nombre de paramètres à savoir :

- la position de l'amer recherché : cette position est supposée connue avec précision puisque nous ne remettons pas en cause la précision de la carte dans ce travail.
- la position estimée du robot : il est important de prendre en compte l'incertitude autour de la pose estimée, dans la mesure où le robot peut se trouver n'importe où dans cette zone ce qui peut ramener à des mauvaises associations de données et par conséquent à des estimations non fiables de la pose du robot. Ainsi plus l'incertitude sur la position estimée est faible, plus la recherche de l'amer à détecter est précise et plus nous serons sûrs d'avoir détecté le bon amer (le rapport signal/bruit s'améliore donc au fur et à mesure que la précision grandit).
- la position du capteur sur le robot : afin d'améliorer la précision de l'estimation il est important de prendre en compte la position du capteur, c'est à dire la configuration du bras de levier (le bras de levier caractérise le fait que le capteur n'est pas parfaitement centré sur le robot).

Le principe de la focalisation spatiale consiste à réduire la taille de la zone de recherche. Au lieu de considérer l'ensemble de la scène, le système utilise les informations dont il dispose et cherche à détecter l'amer dans la zone de focalisation calculée. Pour illustrer ce principe considérons la figure IV.8. La position réelle du robot est  $X_k$  et celle estimée est  $\hat{X}_k$ . Cette dernière est intègre puisque la zone d'incertitude (en bleu) englobe la position réelle. Le système sélectionne le poteau à détecter. L'ellipse d'incertitude est par conséquent projetée dans l'espace de recherche (ellipse en noir) où on pense trouver l'amer recherché. Dans cet ordre d'idée, la focalisation permet d'améliorer le rapport Signal sur Bruit et de diminuer le temps de calcul puisque la réduction de la taille de la zone de recherche permet de réduire la quantité de données à traiter.

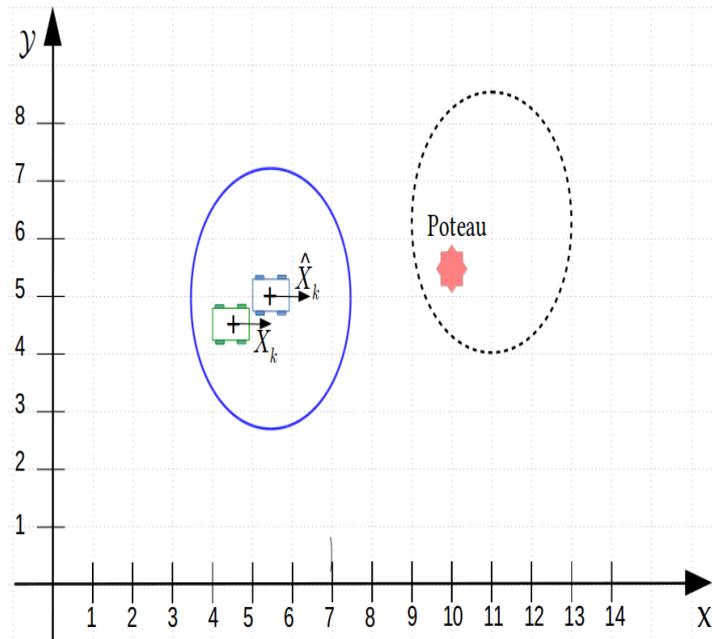


FIGURE IV.8 – Illustration du principe de focalisation spatiale. La position réelle du robot est  $X_k$  et celle estimée est  $\hat{X}_k$  avec une zone d'incertitude associée donnée par l'ellipse bleue. La zone de recherche ROI est donnée par l'ellipse en noir. Étant donnée l'incertitude de la pose estimée, cette zone ne sera pas centrée sur l'amer.

Une deuxième focalisation est aussi réalisée dans l'espace des paramètres. Elle s'intéresse particulièrement aux caractéristiques de l'amer recherché et au problème d'association des données. Supposons par exemple que le robot est équipé d'un télémètre et que nous cherchons à détecter un poteau. Le détecteur en question va chercher donc à trouver un demi cercle. Cependant le détecteur détecte trois demi cercles. Le problème consiste donc à associer les données détectées avec les amers de la carte. Dans cette situation, l'algorithme doit considérer le poteau qui se situe à la distance  $d_A$  supposée connue entre la position du robot et celle de l'amer.

## IV.5 Mise à jour

Une fois le triplet le plus pertinent sélectionné, la phase de détection de celui-ci commence. Si le détecteur détecte quelque chose dans la ROI (l'amer recherché se trouve dans la zone de focalisation ou bien le détecteur détecte un outlier (donnée aberrante)), la mesure retournée par le capteur est donc utilisée pour mettre à jour le vecteur d'état du robot avec la matrice de covariance associée. Pour cela, nous avons utilisé des filtres bayésiens différents selon les triplets (voir chapitre V). Sinon (aucun amer détecté), l'ancien vecteur d'état et sa matrice seront conservés.

Quant à la confiance d'estimation, elle est tenue à jour à chaque essai de détection. Cette mise à jour est assurée aussi par un réseau bayésien. En effet, à cette étape, la valeur de l'évènement  $d_b$  (retour du détecteur)

est retournée et sera ensuite intégrée dans le réseau comme une évidence (ce qui signifie mettre  $P(d_b)$  à 1 si quelque chose a été détectée, 0 sinon).

Après avoir détaillé les trois premières étapes de notre algorithme, nous passons à la dernière étape : le retour en arrière. Dans cette partie, nous allons nous intéresser au phénomène d'ambiguïté qui peut avoir lieu en cas d'amers voisins. Bien que le réseau bayésien utilisé pour le calcul de la confiance permet de gérer en partie le problème d'ambiguïté en baissant le gain de confiance des triplets ambigus mais aussi en définissant une région d'intérêt optimale de recherche des amers (ce qui améliore grandement le rapport signal / bruit de la détection), ce phénomène reste cependant présent dans le cas où tous les triplets sont ambigus surtout en situation d'estimation de pose initiale car l'incertitude est très grande et plusieurs amers peuvent être compatibles avec l'amer recherché.

## IV.6 Processus de retour en arrière

### IV.6.1 Principe

La similitude des amers dans la carte de l'environnement ainsi que l'imprécision de la pose estimée sont deux facteurs qui mènent à une mauvaise association des données détectées avec les amers de la carte. Typiquement l'association se fait suivant un critère de sélection qui permet de choisir la meilleure association possible. La méthode la plus simple et la plus intuitive d'association de données est l'association avec le plus proche voisin. Cette méthode consiste à calculer la distance de tous les amers obtenus avec l'amer recherché et de sélectionner parmi ceux restants celui dont la distance est la plus courte. La distance la plus souvent utilisée et que nous avons employée dans nos travaux est la distance de Mahalanobis. Cette méthode est cependant loin d'exclure les mauvaises associations d'où l'intérêt du processus de remise en cause.

L'objectif du processus récursif que nous avons mis en place est :

- d'améliorer la qualité de l'estimation dans le cas où cette dernière ne répond pas aux objectifs de précision et/ou de confiance fixés préalablement. Sachant qu'une détection réussie permet d'améliorer la précision et la confiance de l'estimation, à chaque fois que la dernière estimation n'est pas assez satisfaisante, le robot va chercher à détecter d'autres amers afin d'améliorer davantage la qualité de l'estimation.
- de remettre en cause l'échec de détection et par conséquent la baisse de la confiance. En effet, l'échec de détection peut être dû à une mauvaise association de données à l'étape précédente. Un retour en arrière est ainsi nécessaire pour la corriger.
- d'assurer une estimation robuste de la position du robot à l'état statique ou bien à l'étape d'initialisation où la pose du robot doit être fiable et précise.

### IV.6.2 Algorithme récursif développé

Le processus que nous proposons est ainsi donné par la fonction localisation de l'algorithme 1. Ce dernier montre la fonction qui gère le retour en arrière. Elle est assimilée à un arbre de possibilité. Nous descendons dans l'arbre si la détection est réussie tant que l'objectif à atteindre (de précision et/ou de confiance) est toujours en dessous du seuil désiré. Si la détection échoue, nous mettons à jour la confiance de l'état (la confiance diminue) et nous cherchons à détecter les autres feuilles du même niveau. Si la détection échoue toujours, nous remontons dans l'arbre afin de remettre en cause la dernière détection réussie jusqu'à parcourir toutes les branches de l'arbre. Dans cette situation, l'algorithme reprend la dernière pose estimée (le vecteur d'état avec la covariance associée) avant cette dernière détection.

---

**Algorithme 1** : localisation(Etat, listeTriplets)

---

**Require:** Trier listeTriplets par ordre de pertinence décroissant

```
for triplet dans listeTripletsOrdonnee do
  detectionOk , EtatNew = triplet.DetectionMAJ (Etat,listeTriplets)
  Etat.confiance = EtatNew.confiance
  if detectionOk == True then
    if EtatNew.SeuilHautAtteint() == True then
      return detectionOk,EtatNew
    end if
  else
    if longueur(listeTripletsOrdonnee) == 1 then
      return detectionOk,EtatNew
    else
      nouvelleListeTriplets=listeTriplets.remove(triplet)
      for triplet dans nouvelleListeTriplets do
        calculer la pertinence
        OkObservable = triplet.IsObservable(EtatNew)
        if OkObservable == True then
          ajouter triplet à listeObservables
        end if
      end for
      detectionOk , EtatNew = Reconnaissance(Etat, listeObservables)
      if detectionOk == True then
        return detectionOk,EtatNew
      end if
    end if
  end if
end for
return detectionOk,Etat
```

---

### IV.6.3 Exemple

Considérons l'exemple montré par la figure IV.9. Nous supposons que nous disposons de trois triplets observables et de pertinences décroissantes  $T_1$ ,  $T_2$  et  $T_3$ . Nous commençons par le niveau  $L_0$  avec l'état  $E_{rw0} = \{\hat{X}_{rw0}, C_{rw0}, P_0\}$ . Le triplet  $T_1$  ayant le gain informationnel probable (c'est à dire la pertinence) le plus élevé, le système va chercher à détecter l'amer correspondant. Deux possibilités se présentent :

- le détecteur détecte l'amer recherché et l'état sera mis à jour. On obtient l'état  $E_{rw1}$  (dans lequel la pose et la confiance ont été actualisées). Ensuite on vérifie si l'objectif préalablement fixé est atteint.
  - si le critère est atteint : l'algorithme retourne l'état  $E_{rw1}$  mis à jour par la mesure retournée par le capteur  $T_1$ .
- sinon (objectif non atteint), on se déplace verticalement et on passe au niveau  $L_1$ . Si  $T_2$  est toujours le plus pertinent après un recalcul du critère (si la détection échoue le critère est calculé à nouveau et l'ordre de pertinence des triplets peut être changé par conséquent), on cherche à détecter l'amer du triplet  $T_2$ .
  - si la détection échoue, la confiance est mise à jour et une détection du triplet  $T_3$  est lancée.
    - si la détection échoue : on remonte au niveau  $L_0$ . À partir de  $\hat{X}_{rw0}$  et  $C_{rw0}$  de l'état  $E_{rw0}$ , on cherchera à détecter l'amer du triplet  $T_2$
    - si la détection est réussie : on obtient le nouvel état  $E_{rw2}$  de mise à jour. Une vérification du seuil désiré sera effectuée afin de décider si on quitte le processus de retour ou bien si on continue vers le niveau  $L_2$  pour détecter le triplet  $T_2$ .
  - si  $T_2$  du niveau  $L_1$  est détecté, on obtient le nouvel état  $E_{rw2}$  et on vérifie à nouveau le seuil à atteindre.
    - le seuil est vérifié : on quitte le processus de retour avec l'état  $E_{rw2}$  en sortie
    - le seuil n'est pas vérifié : on descend au niveau  $L_2$  pour détecter le triplet  $T_3$ . Si  $T_3$  est détecté, une mise à jour aura lieu. On obtient alors l'état  $E_{rw3}$ . Sinon, la confiance sera mise à jour (elle diminue) et on remonte au niveau  $L_1$ . À partir de  $\hat{X}_{rw1}$  et  $C_{rw1}$  de l'état  $E_{rw1}$  on cherchera à détecter l'amer du triplet  $T_3$  de la branche formée par  $(T_1, T_3, T_2)$
- le détecteur n'a rien détecté. La confiance ayant baissé, on garde le même vecteur d'état  $\hat{X}_{rw0}$  et la même covariance  $C_{rw0}$  et on se déplace horizontalement dans l'arbre en cherchant à détecter le triplet  $T_2$ .

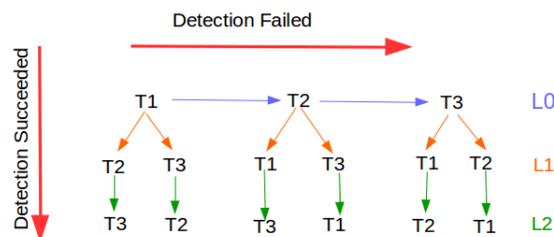


FIGURE IV.9 – Illustration du principe du retour en arrière

Pour résumer, il faut retenir que :

- lorsqu'une détection est réalisée avec succès, on met à jour, on focalise et on descend dans l'arbre jusqu'à la satisfaction de l'objectif
- sinon, le critère de pertinence est à nouveau calculé et le triplet le plus pertinent de ce niveau est considéré pour effectuer le même processus

— si la détection de tous les triplets d'un même niveau a échoué, on remonte d'un niveau

Nous précisons que lorsqu'on remonte d'un niveau la précision revient là où elle en était restée mais la confiance (altérée par les non détections observées) est elle, affectée dans tous les cas..

#### IV.6.4 Bilan

Dans cette partie, nous nous sommes intéressés au problème d'ambiguïté qui peut avoir lieu en cas d'amers voisins et faire baisser la performance de notre estimation. Bien que le réseau bayésien utilisé pour le calcul de la confiance permet de gérer en partie le problème d'ambiguïté en baissant le gain de confiance des triplets ambigus, ce phénomène reste cependant présent dans le cas où *tous* les triplets sont ambigus. Le problème lié à ce phénomène est l'association des données qui se manifeste à cause d'une détection non efficace. Ce problème consiste à associer les données détectées avec les amers de la carte. Typiquement l'association se fait suivant un critère de sélection qui permet de choisir la meilleure association possible et en utilisant une technique de focalisation afin de réduire au mieux les possibilités de mauvaises associations. Néanmoins dans certains cas cela s'avère insuffisant, par exemple lorsque la précision de notre estimation est faible (la zone d'incertitude est grande) et par conséquent la zone de focalisation l'est aussi. Dans ce cas, le problème d'une mauvaise association entre l'amer recherché et l'amer détecté persiste toujours.

Ainsi, un système de remise en cause a été mis en place. Notre objectif sera donc d'être capable, lorsque cela s'avère nécessaire de revenir sur une telle association et de modifier le choix effectué jusqu'à la bonne association. Par conséquent, notre algorithme prenant en compte ce processus récursif est donné par la figure IV.10.

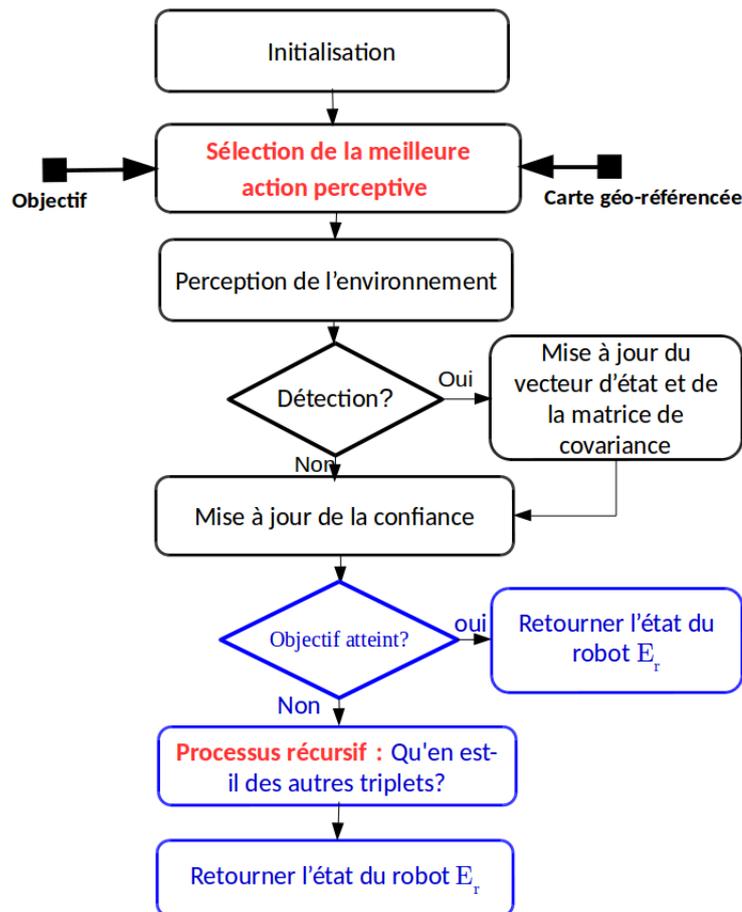


FIGURE IV.10 – Algorithme de localisation avec un processus récursif

## IV.7 Conclusion

À l'issue de ce chapitre, nous avons donné une description de notre approche avec les principales étapes de notre algorithme. La méthode que nous avons proposée est conçue pour une application de localisation de véhicules dans un milieu présentant des ambiguïtés avec prise en compte de l'environnement, des capteurs et des détecteurs. Elle est basée sur l'utilisation d'un critère informationnel pour la sélection du triplet perceptif le plus pertinent. Elle présente en prime l'avantage d'assurer simultanément une estimation précise et fiable de la pose du robot. Le processus de focalisation utilisé permet en partie de réduire le problème des mauvaises détections. Le problème lié aux fausses détections ou mauvaises associations est corrigé par le processus de retour qui réagit de manière efficace en fonction des objectifs de précision et de confiance fixés préalablement.

Cette méthode que nous proposons est une méthode générique et multisensorielle dans la mesure où différentes modalités capteurs peuvent être exploitées. Nous présentons dans le chapitre qui suit les différents types de triplets que nous avons mis en place dans nos travaux avec une description détaillée pour chacun des calculs de pertinence et des fonctions de mise à jour.

# Chapitre V

## Description des triplets perceptifs

### Sommaire

---

V.1	Introduction	81
V.2	Architecture multi-sensorielle générique	81
V.3	Triplet Lidar-Poteau	82
V.3.1	Notations	82
V.3.2	Analyse de pertinence d'un triplet lidar-poteau	83
V.3.3	Mise à jour d'un triplet lidar-poteau	89
V.4	Triplet Lidar-Mur	90
V.4.1	Notations	90
V.4.2	Analyse de pertinence d'un triplet lidar-mur	91
V.4.3	Mise à jour d'un triplet lidar-mur	99
V.5	Triplet UWB	100
V.5.1	Notations	100
V.5.2	Analyse de pertinence d'un triplet UWB	100
V.5.3	Mise à jour d'un triplet UWB	102
V.6	Triplet GPS	104
V.6.1	Notations	104
V.6.2	Analyse de pertinence d'un triplet GPS	105
V.6.3	Mise à jour d'un triplet GPS	109
V.7	Conclusion	112

---

## V.1 Introduction

L'idée générale de notre algorithme de fusion est de concevoir une architecture modulaire et générique permettant d'exploiter au mieux les données environnantes et d'utiliser d'une manière optimale l'ensemble des capteurs dont dispose le robot. Le système principal est une architecture modulaire haut-niveau disposant d'un ensemble de briques logicielles intégrables qui seront les *triplets perceptifs*. Dans nos travaux, nous introduisons différents types de triplets. Pour des raisons de simplification, le triplet sera désigné par le nom du capteur qui lui correspond ou bien par le nom du capteur et de l'amer en cas de triplets utilisant le même type de capteur. Dans ce chapitre, nous décrivons l'implémentation de ces triplets perceptifs qui sont :

1. un triplet de type *Lidar – Poteau* : nous utilisons un lidar pour détecter un poteau de la carte géo-référencée
2. un triplet de type *Lidar – Mur* : nous utilisons un lidar pour détecter un mur de la carte géo-référencée
3. un triplet de type *UWB* : une balise UWB placée sur le véhicule communiquant avec une autre placée dans l'environnement et retournant la distance entre les deux balises (technique range only)
4. un triplet de type *GPS* : ce triplet n'exploite pas vraiment une carte mais nous considérons qu'il exploite des amers actifs de positions connues ce qui peut être considéré comme une carte

## V.2 Architecture multi-sensorielle générique

Le principe de notre approche descendante proposée est d'exploiter les données de la scène de navigation et les informations stockées dans la carte de l'environnement pour localiser un robot en temps réel. Le robot est équipé de capteurs extéroceptifs tels qu'un récepteur GPS, un télémètre laser et un capteur UWB. La carte géo-référencée utilisée contient des amers de positions connues (des poteaux, des balises et des murs). Ceux-ci considérés comme précis dans la carte, sont néanmoins ambigus (par exemple, rechercher un poteau dans une zone où se trouvent 4 poteaux est ambigu).

Dans le chapitre précédent, nous avons défini dans [IV.2.2](#) la notion d'un *triplet perceptif* comme étant l'ensemble  $\{Amer, Capteur, Détecteur\}$ . Pour chaque type d'amer, un capteur spécifique et un détecteur sont utilisés pour le détecter. Par exemple, pour détecter un mur, un télémètre laser peut être utilisé avec un détecteur de segment.

Nous rappelons les principales étapes de notre algorithme :

- une étape de simulation selon un critère informationnel est lancée afin de sélectionner, parmi tout les triplets observables, le triplet ayant le meilleur gain informationnel (c'est-à-dire le plus prometteur en termes d'apport en information)
- une fois que le meilleur triplet est sélectionné, une tentative de détection du triplet sera exécutée <sup>1</sup>
- selon le résultat de la détection :
  - si la détection est réussie : l'état du robot  $E_{rw} = \{\hat{X}_{rw}, \mathbf{C}_{rw}, \mathbf{P}\}$  est mis à jour

---

1. Dans la suite nous parlerons de "détection de triplet" pour caractériser le processus de détection de l'amer du triplet avec le capteur et l'algorithme de ce triplet

- si la détection a échoué : nous conservons le vecteur d'état  $\hat{\underline{X}}_{rw}$  et la covariance  $\mathbf{C}_{rw}$  précédents, nous mettons à jour la confiance de l'estimation  $\mathbf{P}$
- si l'état après mis à jour ne satisfait pas les objectifs fixés (objectifs de précision et/ou de confiance), le processus de retour en arrière sera lancé afin d'améliorer l'estimation retournée

L'intérêt de l'approche descendante est donc de choisir à chaque instant le triplet susceptible d'apporter la meilleure information. Contrairement à une approche ascendante, l'objectif ne sera pas d'utiliser toutes les données qui sont disponibles mais plutôt de sélectionner la donnée la plus pertinente. Pour cette étape, un critère basé sur l'apport informationnel, défini dans IV.3.4, sera utilisé pour calculer la pertinence de chaque triplet de point de vue précision et intégrité. Dans nos travaux, nous utilisons souvent le terme intégrité ou confiance pour désigner la garantie qu'a notre estimation sur la présence du robot dans l'endroit où on le pense être. Nous considérons que notre estimation est intègre si la vraie position du robot se trouve dans la zone d'incertitude de l'estimation.

Une fois, le système a évalué un triplet comme le plus pertinent, une étape de focalisation aura lieu. Le système utilisera le détecteur de ce triplet pour détecter l'amer en question. Si la détection est réussie, la mesure retournée par le capteur sera ainsi utilisée pour mettre à jour l'état  $E_{rw}$  du robot. Finalement, l'état estimé du robot sera évalué afin de vérifier si les objectifs fixés préalablement sont satisfaits.

Afin de tenir en compte les ambiguïtés de l'environnement et par conséquent la possibilité d'une mauvaise association de données, nous avons ainsi mis en place un processus de retour en arrière (présenté dans la partie IV.6) qui sera utilisé en cas d'échec de détection ou bien si les objectifs fixés ne sont pas atteints.

Dans ce chapitre, nous cherchons à montrer l'aspect générique et multisensoriel de notre approche. Pour cela, au vu de l'algorithme général présenté en IV.6.4 à la page 78, nous allons nous intéresser *systématiquement* pour chaque triplet à :

1. l'analyse de sa pertinence pour permettre son choix ou non parmi les autres triplets,
2. la détection d'un tel triplet et la mise à jour de l'état du robot.

D'une manière générale, chaque triplet sera vu comme un objet qui doit être en mesure de fournir :

- une pertinence en fonction de l'estimation de l'état courant
- une procédure de détection qui renvoie une mesure dans la zone de recherche
- une procédure de mise à jour qui, en fonction de la détection réalisée (ou non), mettra à jour l'état global

## V.3 Triplet Lidar-Poteau

### V.3.1 Notations

Nous considérons ici le cas d'un triplet formé d'un lidar et d'un amer ponctuel (par exemple un poteau) donné par la figure V.1. Nous utilisons les *notations spécifiques* suivantes :

- $\underline{X}_{lw} = (x_{lw}, y_{lw})^T$  : position de l'amer (landmark) dans le repère monde (supposé connu)
- $\underline{X}_{lr} = (x_{lr}, y_{lr})^T$  : position de l'amer dans le repère du robot
- $\underline{X}_{ls} = (x_{ls}, y_{ls})^T$  : position de l'amer dans le repère capteur (lidar ici)
- $\underline{X}_{rw} = (x_{rw}, y_{rw}, \theta_{rw})^T$  : état du robot dans le repère monde,
- $\underline{X}_{sr} = (x_{sr}, y_{sr}, \theta_{sr})^T$  : position du capteur dans le repère robot (connu)
- $\underline{X}_{sw} = (x_{sw}, y_{sw}, \theta_{sw})^T$  : position du capteur dans le repère monde
- $\alpha_{ls}$  : angle que fait l'amer dans le repère lidar

- $\phi$  : demi angle d'ouverture du lidar,
- $R_{min}, R_{max}$  : distances min et max de perception du Lidar.

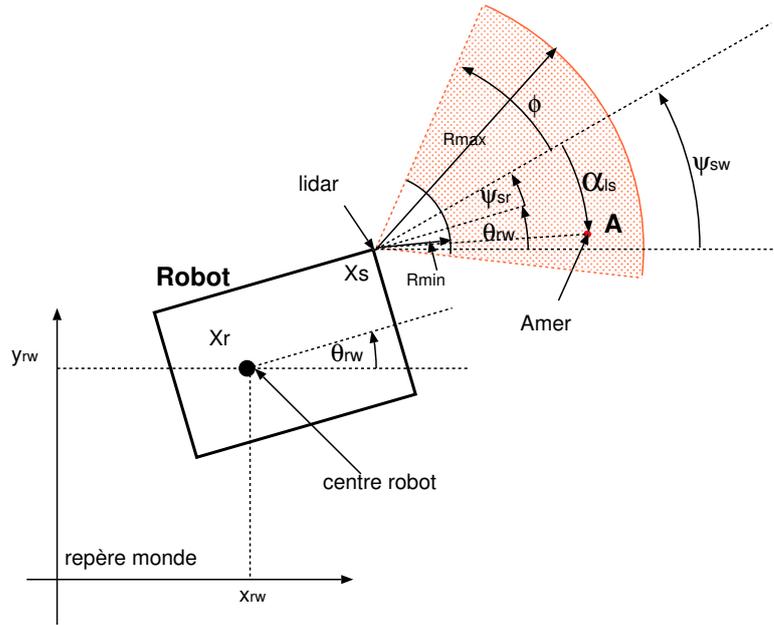


FIGURE V.1 – Représentation du robot, du lidar et de la zone de recherche lidar

### V.3.2 Analyse de pertinence d'un triplet lidar-poteau

L'analyse de pertinence est basée sur les étapes suivantes :

1. détermination de l'observabilité  $P(obs)$  de l'amer compte tenu de l'état a priori du robot mais aussi des caractéristiques du lidar (angle de vue, portée)
2. Calcul de confiance donnée comme suit :
  - (a) Déterminer la ROI (Région d'intérêt) : région dans laquelle on attend cet amer (nous rappelons que nous supposons connaître sa position dans le monde)
  - (b) Déterminer le nombre d'amers compatibles dans cette ROI pour en déduire l'ambiguïté possible
  - (c) Calcul de la confiance  $P(O_k^+)$  : confiance dans le nouvel état une fois que cet amer aura été détecté
3. Calcul de la précision apportée par la détection du poteau (issue d'un filtrage de Kalman)
4. Calcul de la pertinence globale de cet amer

#### V.3.2.1 Observabilité de l'amer de type point

Nous recherchons ici quelle est la probabilité que l'amer  $A$  se trouve dans la zone de scrutation du lidar connaissant :

- la position moyenne du robot dans le monde  $\hat{X}_{rw}$  et sa covariance  $C_{X_{rw}}$ . Ainsi, la position du robot est une variable aléatoire donnée par :

$$\underline{X}_{rw} \sim \mathcal{N}(\hat{X}_{rw}, C_{X_{rw}})$$

- la position du lidar dans le repère robot :  $\underline{X}_{sr}$
- la portée du lidar  $R$  et son angle d'ouverture  $2\phi$

Pour cela, dans un premier temps, il est nécessaire de savoir si l'amer est dans la zone d'observation du capteur. Pour cela il faudra connaître la pose  $\underline{X}_{ls}$  de l'amer dans le repère capteur (nous considérons le bras de levier) puis tester d'une part la distance  $D_{ls}$  de l'amer au centre du lidar, et d'autre part l'angle qu'il fait avec l'axe du lidar  $\alpha_{ls}$ .

### **Test de distance**

La position du capteur dans le monde sera donnée par :

$$\underline{X}_{sw} = \underline{X}_{rw} + \mathbf{R}_{rw}\underline{X}_{sr} = \begin{pmatrix} x_{rw} \\ y_{rw} \end{pmatrix} + \begin{pmatrix} \cos(\theta_{rw}) & -\sin(\theta_{rw}) \\ \sin(\theta_{rw}) & \cos(\theta_{rw}) \end{pmatrix} \begin{pmatrix} x_{sr} \\ y_{sr} \end{pmatrix}$$

Soit :

$$\underline{X}_{sw} = \begin{pmatrix} x_{rw} + x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ y_{rw} + x_{sr} \sin(\theta_{rw}) + y_{sr} \cos(\theta_{rw}) \end{pmatrix} \quad (\text{V.1})$$

Nous cherchons tout d'abord la distance  $D_{ls}$  de l'amer au centre du lidar. Celle-ci sera donnée par :

$$D_{ls}^2 = \|\underline{X}_{lw} - \underline{X}_{sw}\|^2$$

L'amer doit rester dans la zone comprise entre  $R_{min}$  et  $R_{max}$ , il faudra donc avoir simultanément :

$$D_{ls}^2 > R_{min}^2 \quad \text{et} \quad D_{ls}^2 < R_{max}^2$$

### **Test angulaire**

On définit de plus l'angle  $\alpha_{ls}$  que fait la droite reliant l'amer et le centre lidar avec l'axe du lidar (voir figure V.1).  $\alpha_{ls}$  sera donné par :

$$\alpha_{ls} = \arctan\left(\frac{y_{lw} - y_{sw}}{x_{lw} - x_{sw}}\right) - \theta_{rw} - \psi_{sr}$$

avec  $\psi_{sr}$  est l'angle que fait le lidar avec le repère robot (connu par calibration préalable).

Il faudra veiller à ce que l'amer reste dans l'angle d'ouverture  $2\phi$  du lidar.

Une fois que les deux tests de distance et d'angle sont validés, un deuxième test consiste à prendre en compte la zone d'incertitude de la pose a priori du robot. Pour cela, un ensemble de réalisations de  $\underline{X}_{rw}$  suivant la loi  $\sim \mathcal{N}(\hat{\underline{X}}_{rw}, \mathbf{C}_{X_{rw}})$  est généré. Ensuite, par comptage, la probabilité qu'a l'amer  $A$  d'appartenir à la zone d'observation du lidar est estimée. C'est à dire, pour chaque réalisation, une nouvelle pose du robot selon la loi  $\sim \mathcal{N}(\hat{\underline{X}}_{rw}, \mathbf{C}_{X_{rw}})$  est générée et utilisée pour évaluer à nouveau les tests de distance et d'angle. Un comptage des réalisations ayant validé les deux tests est ensuite effectué. Finalement, *la probabilité d'observabilité*  $P(obs|O_k^- = 1)$  sera déduite par une analyse moyenne de l'observabilité de l'amer.

Nous supposons par la suite que  $P(obs|O_k^- = 0) = 0$  (c'est à dire, si la position a priori n'est pas intègre ( $P(O_k^-) = 0$ ), l'amer recherché ne sera pas observable).

### **V.3.2.2 Détermination de la confiance dans la détection du bon amer : $P(db|obs = 1)$**

L'objectif est de connaître la confiance dans la détection d'un triplet de type lidar-poteau (cette confiance caractérise l'ambiguïté de détection). Pour cela nous devons :

- a) Déterminer la ROI (Région d'intérêt)
- b) Déterminer le nombre d'amers compatibles dans cette ROI pour évaluer l'ambiguïté
- c) Prendre en compte les éventuels inliers et outliers de détection
- d) Calculer la confiance en fonction de tous les éléments calculés

### a. Calcul de la ROI

La ROI est la région centrée sur l'amer à détecter dans l'espace capteur (espace lidar). Comme montré par l'équation V.2, cette zone va dépendre de :

- la position du robot  $\underline{X}_{rw}$
- la zone d'incertitude  $\mathbf{C}_{Xrw}$  du robot :  $\underline{X}_{rw}$  étant imprécise et étant donné que la détection de l'amer va dépendre nécessairement de la position du capteur et par conséquent celle du robot, la zone dans laquelle le capteur va chercher à détecter l'amer va elle aussi dépendre de  $\mathbf{C}_{Xrw}$
- la position de l'amer dans le repère lidar  $\underline{X}_{ls}$

$$ROI = f(\underline{X}_{rw}, \underline{X}_{ls}, \mathbf{C}_{Xrw}) \quad (\text{V.2})$$

Nous considérons ainsi que la zone ROI est l'aire formée par l'ensemble des points appartenant à l'ellipse donnée par  $\mathbf{C}_{Xrw}$  projetée dans l'espace de recherche et ayant une distance de Mahalanobis inférieure à un seuil (inférieur à 1 par exemple pour une zone d'incertitude à 1 écart-type).

### b. Évaluation du nombre d'amers dans la ROI

La méthode utilisée pour calculer la probabilité d'observabilité  $P(obs|O_k^- = 1)$  de l'amer est aussi utilisée pour déduire le nombre d'amers présents dans la zone ROI. Cela veut dire qu'un ensemble de réalisations de chaque amer  $A$  observable de position  $\underline{X}_{ls}$  est généré suivant la loi  $\sim \mathcal{N}(\underline{X}_{ls}, \mathbf{C}_{Xrw})$  ( $\underline{X}_{ls}$  est la position de l'amer recherché dans le repère capteur qui est aussi le centre de la ROI) puis par comptage estimer la probabilité qu'a l'amer  $A$  d'appartenir à la zone ROI. Si cette probabilité est positive alors l'amer appartient à la ROI. Ce test est appliqué sur l'ensemble des amers observables et retournera le nombre total d'amers compatibles  $N_c$ .

### c. Prise en compte des inliers et outliers de détection dans la ROI

Selon les performances du détecteur il est nécessaire de prendre en compte :

- la probabilité de détecter un *outlier* dans la ROI (en plus des autres poteaux de la carte) : on notera  $d_{out}$  la densité d'outlier potentielle par unité de surface de la ROI. On notera par  $A_{ROI}$  l'aire observable de la ROI
- la probabilité de ne rien détecter dans la ROI (alors qu'il est censé y avoir au moins le poteau concerné) : on notera cette probabilité  $\alpha$

### d. Calcul de la confiance de détection du bon amer : $P(db|obs = 1)$

Ainsi, si nous considérons un triplet observable, alors, la probabilité  $P(db|obs = 1)$  d'une bonne détection sera donnée par :

$$P(db|obs = 1) = \frac{1 - \alpha}{1 + N_c + N_o} \quad \text{et} \quad P(db|obs = 0) = 0 \quad (\text{V.3})$$

Avec :

- $\alpha$  : la probabilité de ne pas détecter l'amer alors qu'il existe dans la zone. Cette probabilité caractérise les problèmes de détection qui peuvent être dus à cause de différentes raisons comme les conditions météorologiques et la qualité du détecteur
- $N_c$  : nombre d'amers de la carte se retrouvant dans la ROI
- $N_o$  : le nombre d'amers potentiellement détectables mais ne faisant pas partie de la carte. Ces amers peuvent être vus aussi comme des fausses détections (*outliers*) du détecteur.  $N_o$  sera fonction de la densité  $d_{out}$  d'amers par unité de surface dans l'espace capteur

Ainsi  $N_c + N_o$  représente le nombre des autres amers potentiellement détectables à la place du mur considéré.

### V.3.2.3 Calcul de la confiance espérée $P(O_k^+)$

Nous cherchons ici à connaître la confiance dans la prochaine estimation dès lors que nous aurons lancé une détection du triplet. Cette confiance peut être extraite d'un réseau bayésien donné dans la figure V.2.

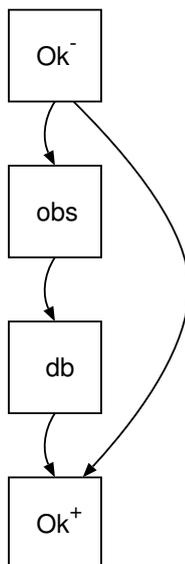


FIGURE V.2 – Réseau bayésien utilisé dans ce triplet lidar-poteau

Les probabilités conditionnelles sont :

$O_k^-$	$P(obs = 1 O_k^-)$	obs	$P(db = 1 obs)$	$O_k^-$	db	$P(O_k^+ = 1 O_k^-, db)$
0	0	0	0	0	0	0
0	$P(obs O_k^- = 1)$	0	0	0	1	0
1	$P(obs O_k^- = 1)$	1	$P(db obs = 1)$	1	0	0
				1	1	1

Avec :

- $P(obs|O_k^- = 1)$  : la probabilité d'observabilité de l'amer montrée dans la partie §V.3.2.1
- $P(db|obs = 1)$  : la probabilité d'une bonne détection donnée par l'équation (V.3).

Du fait des valeurs essentiellement nulles des probabilités conditionnelles, le calcul de  $P(O_k^+)$  donnera simplement :

$$P(O_k^+ = 1) = P(O_k^- = 1) \cdot P(obs|O_k^- = 1) \cdot P(db|obs = 1)$$

### V.3.2.4 Calcul de précision apportée par la détection de l'amer

#### a. Objectifs

Lorsque nous avons sélectionné un amer donné, en plus de la confiance dans sa détection il faut calculer la précision qu'il va amener. Cela va se faire grâce à un filtre de Kalman avec les particularités suivantes :

- nous utiliserons uniquement la partie mise-à-jour de l'état (pas d'évolution ici)
- nous ne calculerons que la covariance de l'état mis à jour (pas l'espérance) car c'est elle qui caractérisera l'apport en précision probable de cet amer.

#### b. Équation de mesure

Nous cherchons ici à modéliser la relation  $\mathbf{h}_{lw}$  décrivant le lien entre la mesure  $\underline{z}_k = (\rho, \phi)^\top$  donnée par le capteur  $s$  lorsqu'il observe l'amer  $l$  de position connue  $\underline{X}_{lw}$  dans le monde et la position du robot dans le monde  $\underline{X}_{rw}$ . Nous avons ainsi :

$$\underline{z}_k = \begin{pmatrix} \rho \\ \phi \end{pmatrix}_k = \mathbf{h}_{lw}(\underline{X}_{lw}, \underline{X}_{rw}) = \begin{pmatrix} h_{lw\rho}(\underline{X}_{lw}, \underline{X}_{rw}) \\ h_{lw\phi}(\underline{X}_{lw}, \underline{X}_{rw}) \end{pmatrix}_k$$

Le bras de levier caractérise le fait que le capteur n'est pas parfaitement centré sur le robot (voir figure V.3).

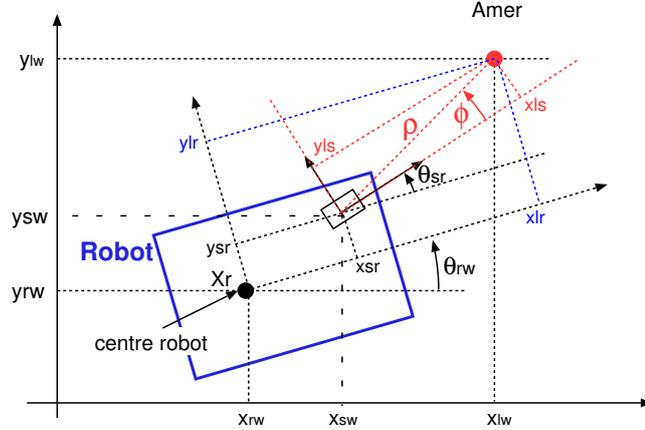


FIGURE V.3 – Notation utilisées lorsque le capteur n'est pas centré sur le robot (phénomène de "bras de levier")

Dans la pratique, le capteur donne sa mesure dans son repère, ainsi on dispose plutôt de la relation  $\mathbf{h}_{ls}$  donnée par :

$$\underline{z}_k = \begin{pmatrix} \rho \\ \phi \end{pmatrix} = \mathbf{h}_{ls}(\underline{X}_{lw}, \underline{X}_{sw}) = \begin{pmatrix} h_{ls\rho}(\underline{X}_{lw}, \underline{X}_{sw}) \\ h_{ls\phi}(\underline{X}_{lw}, \underline{X}_{sw}) \end{pmatrix}$$

Ici, le lidar, observant un amer ponctuel  $\underline{X}_{lw} = (x_{lw}, y_{lw})^\top$  fournira une mesure  $\underline{z}_k = (\rho, \phi)^\top$  donnée par :

$$\underline{z}_k = \begin{pmatrix} \rho \\ \phi \end{pmatrix} = \mathbf{h}_{ls}(\underline{X}_{lw}, \underline{X}_{sw}) = \begin{pmatrix} \sqrt{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2} \\ \arctan\left(\frac{y_{lw} - y_{sw}}{x_{lw} - x_{sw}}\right) - \theta_{sr} - \theta_{rw} \end{pmatrix} \quad (\text{V.4})$$

La position  $\underline{X}_{sw}$  du capteur dans le monde est dépendante à la fois de la position du capteur dans le repère robot et de celle du robot dans le monde selon la relation :

$$\underline{X}_{sw} = \begin{pmatrix} x_{sw} \\ y_{sw} \\ \theta_{sw} \end{pmatrix} = \underline{X}_{rw} + \mathbf{R}_{rw} \underline{X}_{sr} = \begin{pmatrix} x_{rw} \\ y_{rw} \\ \theta_{rw} \end{pmatrix} + \begin{pmatrix} \cos(\theta_{rw}) & -\sin(\theta_{rw}) & 0 \\ \sin(\theta_{rw}) & \cos(\theta_{rw}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{sr} \\ y_{sr} \\ \theta_{sr} \end{pmatrix}$$

Soit :

$$\underline{X}_{sw} = \begin{pmatrix} x_{sw} \\ y_{sw} \\ \theta_{sw} \end{pmatrix} = \mathbf{h}_{sw}(\underline{X}_{sr}, \underline{X}_{rw}) = \begin{pmatrix} x_{rw} + x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ y_{rw} + x_{sr} \sin(\theta_{rw}) + y_{sr} \cos(\theta_{rw}) \\ \theta_{sr} + \theta_{rw} \end{pmatrix} \quad (\text{V.5})$$

Ce qui nous permettra d'obtenir :

$$\underline{z}_k = \begin{pmatrix} \rho \\ \phi \end{pmatrix} = \mathbf{h}_{ls}(\underline{X}_{lw}, \underline{X}_{sw}) = \begin{pmatrix} \sqrt{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2} \\ \arctan\left(\frac{y_{lw} - y_{sw}}{x_{lw} - x_{sw}}\right) - \theta_{sr} - \theta_{rw} \end{pmatrix} \quad (\text{V.6})$$

Avec :

$$\underline{X}_{sw} = \begin{pmatrix} x_{sw} \\ y_{sw} \\ \theta_{sw} \end{pmatrix} = \mathbf{h}_{sw}(\underline{X}_{sr}, \underline{X}_{rw}) = \begin{pmatrix} x_{rw} + x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ y_{rw} + x_{sr} \sin(\theta_{rw}) + y_{sr} \cos(\theta_{rw}) \\ \theta_{sr} + \theta_{rw} \end{pmatrix} \quad (\text{V.7})$$

Les valeurs des mesures théoriques  $\underline{z}_k = (\rho, \phi)^\top$  du poteau dans le repère du capteur ( $s$ ) étant calculées, nous supposons que cette mesure est entachée de bruits blancs gaussiens indépendants  $w_\rho$  et  $w_\phi$  sur  $\rho$  et  $\phi$  d'écart-types  $\sigma_\rho$  et  $\sigma_\phi$ , soit :

$$\underline{w}_{\rho\phi} = (w_\rho, w_\phi)^\top \quad \text{avec pour covariance :} \quad \mathbf{R}_{\rho\phi} = \begin{pmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix}$$

L'équation de mesure sera donc :

$$\underline{z}_k = \begin{pmatrix} \rho \\ \phi \end{pmatrix} = \mathbf{h}_{lw}(\underline{X}_{rw}) = \begin{pmatrix} h_{ls\rho} \\ h_{ls\phi} \end{pmatrix} = \begin{pmatrix} \sqrt{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2} \\ \arctan\left(\frac{y_{lw} - y_{sw}}{x_{lw} - x_{sw}}\right) - \theta_{sr} - \theta_{rw} \end{pmatrix} + \begin{pmatrix} w_\rho \\ w_\phi \end{pmatrix} \quad (\text{V.8})$$

### c. Mise à jour de la covariance $\mathbf{C}_k^+$

L'objectif est de déterminer la covariance  $\mathbf{C}_k^+$  de l'état (qui vaut pour l'instant  $\mathbf{C}_k^-$ ) lorsque le poteau aura été détecté. Cette mise à jour sera faite par un filtre de Kalman étendu qui utilisera les matrices jacobiennes de la fonction de mesure  $\mathbf{h}_{lw}$ .

La mise à jour de la covariance s'effectuera par les relations suivantes :

$$\begin{cases} \mathbf{K}_k &= \mathbf{C}_k^- \mathbf{J}_{lwXrw}^T (\mathbf{J}_{lwXrw} \mathbf{C}_k^- \mathbf{J}_{lwXrw}^T + \mathbf{R}_{\rho\phi})^{-1} \\ \mathbf{C}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{J}_{lwXrw}) \mathbf{C}_k^- \end{cases} \quad (\text{V.9})$$

L'expression de la matrice jacobienne  $\mathbf{J}_{lwXrw}$  de la fonction  $\mathbf{h}_{lw}$  par rapport à  $\underline{X}_{rw}$  est donc nécessaire pour les équations du filtre de Kalman.

Ainsi, du fait que  $\underline{z}_k = \mathbf{h}_{lw}(\underline{X}_{lw}, \underline{X}_{rw})$ , on aura :

$$\mathbf{C}_{zk} = \mathbf{J}_{lwXrw}^\top \mathbf{C}_{Xrw} \mathbf{J}_{lwXrw} \quad \text{avec} \quad \mathbf{J}_{lwXrw} = \begin{pmatrix} \frac{\partial h_{lw\rho}}{\partial x_{rw}} & \frac{\partial h_{lw\rho}}{\partial y_{rw}} & \frac{\partial h_{lw\rho}}{\partial \theta_{rw}} \\ \frac{\partial h_{lw\phi}}{\partial x_{rw}} & \frac{\partial h_{lw\phi}}{\partial y_{rw}} & \frac{\partial h_{lw\phi}}{\partial \theta_{rw}} \end{pmatrix}$$

Du fait que la fonction  $\mathbf{h}_{lw}(\underline{X}_{lw}, \underline{X}_{rw})$  utilise la grandeur intermédiaire  $\underline{X}_{sw}$  du capteur dans le monde, il est plus simple de calculer séparément la matrice jacobienne  $\mathbf{J}_{lwXrw}$  comme suit :

$$\mathbf{J}_{lwXrw} = \mathbf{J}_{lsXsw} \times \mathbf{J}_{swXrw}$$

Avec

$$\mathbf{J}_{lsXsw} = \begin{pmatrix} \frac{\partial h_{ls\rho}}{\partial x_{sw}} & \frac{\partial h_{ls\rho}}{\partial y_{sw}} & \frac{\partial h_{ls\rho}}{\partial \theta_{sw}} \\ \frac{\partial h_{ls\phi}}{\partial x_{sw}} & \frac{\partial h_{ls\phi}}{\partial y_{sw}} & \frac{\partial h_{ls\phi}}{\partial \theta_{sw}} \end{pmatrix} \quad \text{et} \quad \mathbf{J}_{swXrw} = \begin{pmatrix} \frac{\partial x_{sw}}{\partial x_{rw}} & \frac{\partial x_{sw}}{\partial y_{rw}} & \frac{\partial x_{sw}}{\partial \theta_{rw}} \\ \frac{\partial y_{sw}}{\partial x_{rw}} & \frac{\partial y_{sw}}{\partial y_{rw}} & \frac{\partial y_{sw}}{\partial \theta_{rw}} \\ \frac{\partial \theta_{sw}}{\partial x_{rw}} & \frac{\partial \theta_{sw}}{\partial y_{rw}} & \frac{\partial \theta_{sw}}{\partial \theta_{rw}} \end{pmatrix}$$

**Calcul de  $\mathbf{J}_{lsXsw}$**  : La matrice jacobienne  $\mathbf{J}_{lsXsw}$  sera donnée par :

$$\mathbf{J}_{lsXsw} = \begin{pmatrix} \frac{\partial \rho}{\partial x_{sw}} & \frac{\partial \rho}{\partial y_{sw}} & \frac{\partial \rho}{\partial \theta_{sw}} \\ \frac{\partial \phi}{\partial x_{sw}} & \frac{\partial \phi}{\partial y_{sw}} & \frac{\partial \phi}{\partial \theta_{sw}} \end{pmatrix}$$

Ainsi, on aura :

$$\mathbf{J}_{lsXsw} = \begin{pmatrix} \frac{x_{sw} - x_{lw}}{\sqrt{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2}} & \frac{y_{sw} - y_{lw}}{\sqrt{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2}} & 0 \\ \frac{y_{lw} - y_{sw}}{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2} & \frac{x_{sw} - x_{lw}}{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2} & -1 \end{pmatrix}$$

**Calcul de  $\mathbf{J}_{swXrw}$**  : Du fait de l'expression (V.5) reprise ici :

$$\underline{X}_{sw} = \mathbf{h}_{sw}(\underline{X}_{sr}, \underline{X}_{rw}) = \begin{pmatrix} x_{sw} \\ y_{sw} \\ \theta_{sw} \end{pmatrix} = \begin{pmatrix} x_{rw} + x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ y_{rw} + x_{sr} \sin(\theta_{rw}) + y_{sr} \cos(\theta_{rw}) \\ \theta_{sr} + \theta_{rw} \end{pmatrix} \quad (\text{V.10})$$

on aura :

$$\mathbf{J}_{swXrw} = \begin{pmatrix} \frac{\partial x_{sw}}{\partial x_{rw}} & \frac{\partial x_{sw}}{\partial y_{rw}} & \frac{\partial x_{sw}}{\partial \theta_{rw}} \\ \frac{\partial y_{sw}}{\partial x_{rw}} & \frac{\partial y_{sw}}{\partial y_{rw}} & \frac{\partial y_{sw}}{\partial \theta_{rw}} \\ \frac{\partial \theta_{sw}}{\partial x_{rw}} & \frac{\partial \theta_{sw}}{\partial y_{rw}} & \frac{\partial \theta_{sw}}{\partial \theta_{rw}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -x_{sr} \sin(\theta_{rw}) - y_{sr} \cos(\theta_{rw}) \\ 0 & 1 & x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{V.11})$$

### V.3.2.5 Calcul de pertinence d'un triplet lidar-poteau : récapitulatif

Le calcul de la pertinence d'un triplet lidar-poteau se fait par l'analyse de son *gain entropique* qui prendra en compte :

- le gain en confiance (évolution entre  $P(O_k^-)$  et  $P(O_k^+)$ ) expliqué dans le chapitre précédent (IV.3.3) et en utilisant la confiance espérée de la partie V.3.2.3
- le gain en précision (évolution entre  $C_k^-$  et  $C_k^+$ ) expliqué dans le chapitre précédent (IV.3.2) et en utilisant la précision espérée de la partie §c. à la page 88

Nous chercherons donc à calculer le gain entropique correspondant à la différence entre l'entropie initiale  $\mathbf{H}^-$  et l'entropie après détection potentielle  $\mathbf{H}^+$  comme vu précédemment (section IV.3.4). Ce gain est donné par l'équation V.12. En effet, si la détection échoue, le processus de mise à jour de l'état n'est pas lancé et la probabilité qu'une détection soit bien réalisée est  $P(d_b)$ .

$$\mathbf{G} = [1 - P(d_b = 1)] \mathbf{G}_c + P(d_b = 1)[\mathbf{G}_p + \mathbf{G}_c] \quad (\text{V.12})$$

La valeur de  $P(d_b)$  sera déduite du réseau de la figure V.2 et sera déterminée par la marginalisation sur  $P(d_b)$  de  $P(O_k^-, obs, d_b) = P(O_k^-) \cdot P(obs|O_k^-) \cdot P(d_b|obs)$ .

### V.3.3 Mise à jour d'un triplet lidar-poteau

Lorsqu'un triplet a été sélectionné, une opération de détection est lancée. Il est nécessaire de prendre en compte :

- le fait que la détection a ou non donné un résultat (impacte donc la confiance dans la pose du robot)
- la mesure qu'a effectivement donné cette détection (précise l'estimation de la pose)

#### V.3.3.1 Mise à jour de la confiance : $P(O_k^+)$

Lorsqu'une détection a été lancée, nous allons avoir ou non un résultat (la détection a donné quelque chose ou non) : il s'agira donc de prendre en compte l'évidence  $d_b$  (une détection a été faite ou non) pour mettre à jour la probabilité  $P(O_k^+|d_b)$ .

Les probabilités conditionnelles sont identiques à celles utilisées dans la sélection de l'amer (§V.3.2.3) :

$O_k^-$	$P(obs = 1 O_k^-)$	$obs$	$P(db = 1 obs)$	$O_k^-$	$db$	$P(O_k^+ = 1 O_k^-, db)$
0	0	0	0	0	0	0
0	0	0	0	0	1	0
1	$P(obs O_k^- = 1)$	1	$P(db obs = 1)$	1	0	0
1	$P(obs O_k^- = 1)$	1	$P(db obs = 1)$	1	1	1

Ainsi, nous pourrions, grâce à l'évidence  $d_b$  (une détection a été faite et a réussi ou non), inférer la probabilité  $P(O_k^+)$  que l'on soit bien à l'endroit escompté. Cette opération se fera grâce à l'utilisation d'une petite librairie logicielle de calcul d'inférence sur un réseau bayésien binaire.

### V.3.3.2 Mise à jour de la pose de l'état

Si la *détection a réussi* (soit  $d_b = 1$ ) alors, l'objectif est de mettre à jour l'état  $\underline{X}_k^-$  et sa covariance  $\mathbf{C}_k^-$  pour donner  $\underline{X}_k^+$  et sa covariance  $\mathbf{C}_k^+$ . On notera que  $\underline{X}_k^-$  et sa covariance  $\mathbf{C}_k^-$  ne seront pas touchés si la détection n'a pas réussi. Dans le cas où la détection n'a pas réussi (le détecteur n'a rien pu détecter), la mise à jour de la pose n'est simplement pas réalisée. Pour la mise à jour, nous utiliserons les relations suivantes :

$$\boxed{\begin{cases} \mathbf{K}_k &= \mathbf{C}_k^- \mathbf{J}_{lw_{X_{rw}}}^T (\mathbf{J}_{lw_{X_{rw}}} \mathbf{C}_k^- \mathbf{J}_{lw_{X_{rw}}}^T + \mathbf{R}_{\rho\phi})^{-1} \\ \underline{X}_k^+ &= \underline{X}_k^- + \mathbf{K}_k (\underline{z}_k - \underline{z}_k^-) \\ \mathbf{C}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{J}_{lw_{X_{rw}}}) \mathbf{C}_k^- \end{cases}} \quad (\text{V.13})$$

La mesure  $\underline{z}_k = (\hat{\rho}_{ls}, \hat{\phi}_{ls})$  sera donnée par le détecteur de segment et la *mesure prédite*  $\underline{z}_k^-$  (voir aussi équation (V.6)) sera donnée par :

$$\underline{z}_k^- = \mathbf{h}_{lw}(\underline{X}_{rw}^-) = \left( \sqrt{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2}, \arctan\left(\frac{y_{lw} - y_{sw}}{x_{lw} - x_{sw}}\right) - \theta_{sr} - \theta_{rw} \right) \quad (\text{V.14})$$

## V.4 Triplet Lidar-Mur

### V.4.1 Notations

Nous nous intéressons ici au cas de triplet avec des amers de type segment (représentant des murs rectilignes) avec un capteur Lidar donné par la figure V.4.

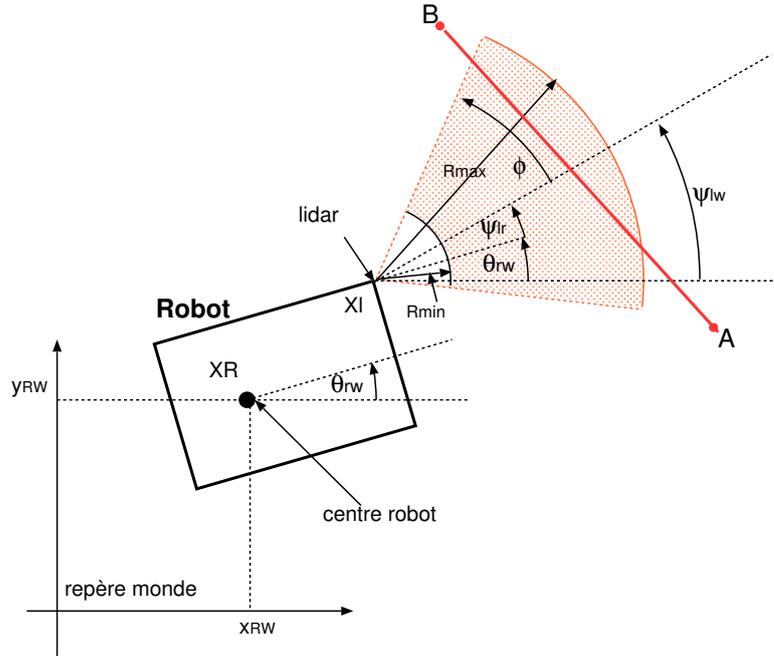


FIGURE V.4 – Représentation du robot, du lidar, de la zone de recherche lidar et du segment AB représentant un mur

En plus des notations précédentes (section V.3.1), nous utiliserons les *notations spécifiques* suivantes :

- $A, B$  : points définissant le segment. Ainsi nous aurons :
  - dans le repère monde :  $(A_{xw}, A_{yw}, B_{xw}, B_{yw})$
  - dans le repère robot :  $(A_{xr}, A_{yr}, B_{xr}, B_{yr})$
  - dans le repère lidar :  $(A_{xs}, A_{ys}, B_{xs}, B_{ys})$

## V.4.2 Analyse de pertinence d'un triplet lidar-mur

De la même manière que le triplet précédent, l'objectif ici est encore de déterminer la *pertinence informationnelle* amenée par le choix d'un triplet lidar-mur. Pareillement, l'*analyse de pertinence* est basée sur le calcul des probabilités des évènements suivants :

1. *obs* : observabilité du mur par le lidar en fonction de la pose probable du robot : nous cherchons ainsi  $P(obs)$
2. *db* : évènement caractérisant le fait qu'on a fait la détection du bon mur parmi éventuellement d'autres murs dans la ROI mais sachant aussi que le détecteur peut détecter des outliers. Nous suivrons les étapes suivantes :
  - (a) Déterminer la ROI
  - (b) Déterminer le nombre d'amers compatibles détectables en lieu et place du mur recherché dans cette ROI
  - (c) Calcul effectif de la probabilité *db*
3. Calcul de la confiance  $P(O_k^+)$
4. Calcul de la précision apportée par la détection de l'amer segment
5. Calcul de la pertinence globale de cet amer

### V.4.2.1 Détermination de la probabilité d'observabilité du segment : $P(obs|O_k^- = 1)$

L'objectif de cette étape est d'estimer la probabilité d'observabilité d'un segment dans la zone lidar (figure V.5) sachant que nous supposons être positionné de manière intègre ( $O_k^- = 1$ ). Nous cherchons l'angle  $\theta$  de visibilité du segment.

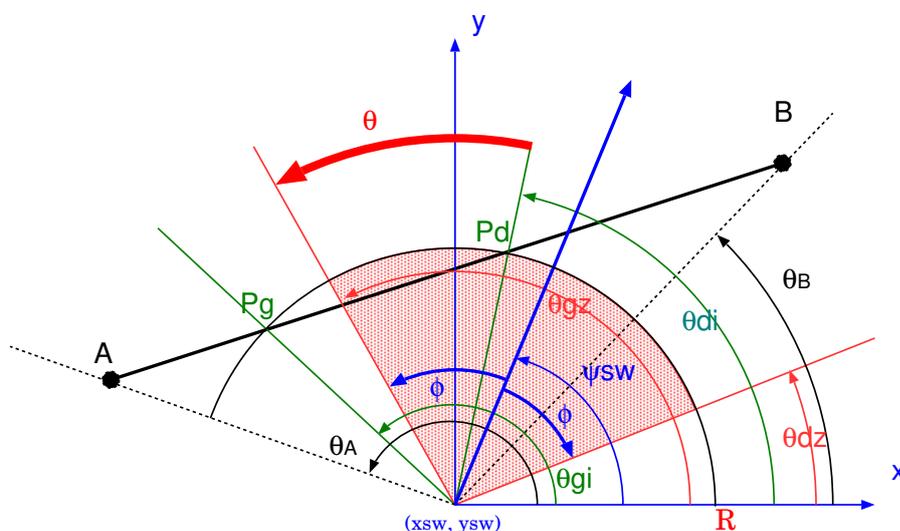


FIGURE V.5 – Différents cas possibles d'observabilité d'un segment. La zone hachurée correspond à l'ouverture lidar.

#### a. Observabilité d'un segment sans occultation et avec positionnement connu du lidar

Dans un premier temps nous cherchons qu'elle est la probabilité d'observabilité d'un segment sans autres segments potentiellement occultants. *Nous supposons connaître précisément ici le positionnement  $\underline{X}_{sw}$  du lidar dans le monde* (nous verrons ensuite le cas plus général).

Nous nous rendons compte que (figure V.5) :

$$\theta = \min(\theta_A, \theta_{gi}, \theta_{gz}) - \max(\theta_B, \theta_{di}, \theta_{dz})$$

— Les angles  $\theta_A$  et  $\theta_B$  sont donnés par :

$$\theta_A = \arctan\left(\frac{A_y - O_y}{A_x - O_x}\right) \quad \text{et} \quad \theta_B = \arctan\left(\frac{B_y - O_y}{B_x - O_x}\right)$$

— les angles  $\theta_{gi}$  et  $\theta_{di}$  sont les angles des points d'intersection du segment (ou plutôt de droite qui le porte) avec le cercle de rayon  $R$ . Regardons la figure V.6.

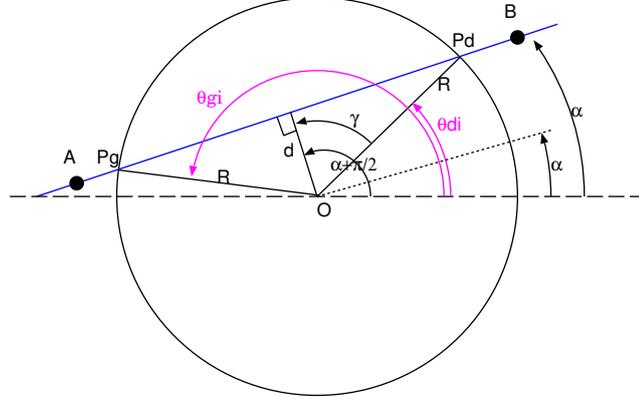


FIGURE V.6 – calcul des angles  $\theta_{di}$  et  $\theta_{gi}$

La distance  $d$  entre  $O$  et la droite  $(AB)$  sera donnée par :  $d = \frac{\|AO \wedge AB\|}{\|AB\|}$ , et l'angle  $\alpha$  vaudra :  $\alpha = \arctan\left(\frac{A_y - B_y}{A_x - B_x}\right)$

De plus l'angle  $\gamma$  est tel que :  $R \cos(\gamma) = d$  soit  $\gamma = \arccos\left(\frac{d}{R}\right)$ .

On aura ainsi :

$$\theta_{di} = \alpha + \frac{\pi}{2} - \gamma \quad \text{et} \quad \theta_{gi} = \alpha + \frac{\pi}{2} + \gamma \quad \text{avec} \quad \begin{cases} d = \frac{\|AO \wedge AB\|}{\|AB\|} \\ \alpha = \arctan\left(\frac{A_y - B_y}{A_x - B_x}\right) \\ \gamma = \arccos\left(\frac{d}{R}\right) \end{cases}$$

— Les angles  $\theta_{dz}$  et  $\theta_{gz}$  seront simplement donnés par :

$$\theta_{dz} = \psi_{sw} + \phi \quad \text{et} \quad \theta_{gz} = \psi_{sw} - \phi$$

Ainsi, pour récapituler, le segment  $AB$  sera considéré visible vu de l'intervalle angulaire  $\theta \in [\theta_g, \theta_d]$  tels que :

$$\boxed{\theta_g = \min(\theta_A, \theta_{gi}, \theta_{gz}) \quad \text{et} \quad \theta_d = \max(\theta_B, \theta_{di}, \theta_{dz})} \quad (\text{V.15})$$

Avec :

$$\begin{cases} \theta_A = \arctan\left(\frac{A_y - O_y}{A_x - O_x}\right) \\ \theta_B = \arctan\left(\frac{B_y - O_y}{B_x - O_x}\right) \\ \theta_{di} = \alpha + \frac{\pi}{2} - \gamma \\ \theta_{gi} = \alpha + \frac{\pi}{2} + \gamma \\ \theta_{dz} = \psi_{sw} + \phi \\ \theta_{gz} = \psi_{sw} - \phi \end{cases} \quad \text{avec} \quad \begin{cases} d = \frac{\|AO \wedge AB\|}{\|AB\|} \\ \alpha = \arctan\left(\frac{A_y - B_y}{A_x - B_x}\right) \\ \gamma = \arccos\left(\frac{d}{R}\right) \end{cases}$$

### b. Observabilité de segments avec occultations et positionnement aléatoire du lidar dans le monde

Dans la pratique, nous ne connaissons pas la pose réelle  $\underline{X}_{sw}$  du lidar dans le monde. Elle dépend de la pose  $\underline{X}_{sr}$  du lidar dans le repère robot et de la pose inconnue  $\underline{X}_{rw}$  du robot dans le monde qui est censée suivre une loi du type :

$$\underline{X}_{rw} \sim \mathcal{N}\left(\hat{X}_{rw}, \mathbf{C}_{rw}\right)$$

Pour ce faire il faut :

- déterminer la pose du lidar dans le monde et sa covariance. On utilisera pour cela les relations (V.10) et (V.11) reprises (V.16) :

$$\underline{X}_{sw} = \begin{pmatrix} x_{rw} + x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ y_{rw} + x_{sr} \sin(\theta_{rw}) + y_{sr} \cos(\theta_{rw}) \\ \theta_{sr} + \theta_{rw} \end{pmatrix} \quad \text{et} \quad \mathbf{J}_{swX_{rw}} = \begin{pmatrix} 1 & 0 & -x_{sr} \sin(\theta_{rw}) - y_{sr} \cos(\theta_{rw}) \\ 0 & 1 & x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{V.16})$$

La position réelle  $\underline{X}_{sw}$  du lidar dans le monde sera donc supposée suivre une loi normale s'exprimant par :

$$\underline{X}_{sw} \sim \mathcal{N}(\hat{X}_{sw}, \mathbf{C}_{sw}) \quad \text{avec} \quad \mathbf{C}_{sw} = \mathbf{J}_{swX_{rw}} \mathbf{C}_{rw} \mathbf{J}_{swX_{rw}}^\top$$

Avec :

$$\hat{X}_{sw} = \begin{pmatrix} x_{rw} + \hat{x}_{sr} \cos(\hat{\theta}_{rw}) - \hat{y}_{sr} \sin(\hat{\theta}_{rw}) \\ y_{rw} + \hat{x}_{sr} \sin(\hat{\theta}_{rw}) + \hat{y}_{sr} \cos(\hat{\theta}_{rw}) \\ \theta_{sr} + \hat{\theta}_{rw} \end{pmatrix} \quad \text{et} \quad \mathbf{J}_{swX_{rw}} = \begin{pmatrix} 1 & 0 & -\hat{x}_{sr} \sin(\hat{\theta}_{rw}) - \hat{y}_{sr} \cos(\hat{\theta}_{rw}) \\ 0 & 1 & \hat{x}_{sr} \cos(\hat{\theta}_{rw}) - \hat{y}_{sr} \sin(\hat{\theta}_{rw}) \\ 0 & 0 & 1 \end{pmatrix}$$

L'approche pour évaluer l'observabilité du segment sera simplement de tirer des réalisations aléatoires de  $\underline{X}_{sw}$  selon la loi  $\underline{X}_{sw} \sim \mathcal{N}(\hat{X}_{sw}, \mathbf{C}_{sw})$  et de faire une analyse moyenne de l'observabilité du segment.

Ainsi nous en déduirons la *probabilité d'observabilité*  $P(obs|O_k^- = 1)$ . Nous supposons par la suite que  $P(obs|O_k^- = 0) = 0$ .

#### V.4.2.2 Détermination de la confiance dans la détection du bon amer : $P(db|Obs = 1)$

L'objectif est de connaître la confiance dans la détection d'un triplet de type lidar-mur. Pour cela nous devons :

- Déterminer la ROI
- Déterminer le nombre d'amers compatibles dans cette ROI pour évaluer l'ambiguïté
- Prendre en compte les éventuels inliers et outliers de détection
- Calculer la confiance en fonction de tous les éléments calculés

##### a. Calcul de la ROI

La ROI est la région centrée sur le segment à détecter dans l'espace capteur (espace lidar). On suppose qu'on connaît les coordonnées des deux points  $A$  et  $B$  du segment à tester dans le monde. Le calcul de cette région se fera de la manière suivante :

- Tirage de  $2N + 1 = 7$  sigma points  $X_w^i$  ( $N = 3$  car on a une dimension 3 du vecteur d'état) représentant les configurations possibles de la pose du robot dans le monde.
- Pour chaque position du robot en  $X_w^i$ , Projeter les deux points  $A_w = (A_x, A_y)$  et  $B_w = (B_x, B_y)$  du segment  $[AB]$  (définis dans le monde) dans l'espace capteur (en considérant que le robot est positionné sur le sigma point  $X_w^i$ ), ce qui donnera les composantes  $z_A^i = (\rho_A^i, \phi_A^i)$  et  $z_B^i = (\rho_B^i, \phi_B^i)$ . Cette projection se fera à partir de la relation (V.6) qui, une fois adaptée donnera (par exemple ici pour le point  $A$ ) la relation (V.17) :

$$z_A^i = \begin{pmatrix} \rho_A^i \\ \phi_A^i \end{pmatrix} = \mathbf{h}_{ls}(A, \underline{X}_{sw}) = \begin{pmatrix} \sqrt{(A_x - x_{sw})^2 + (A_y - y_{sw})^2} \\ \arctan\left(\frac{A_y - y_{sw}}{A_x - x_{sw}}\right) - \theta_{sr} - \theta_{rw} \end{pmatrix} \quad (\text{V.17})$$

Avec :

$$\underline{X}_{sw} = \begin{pmatrix} x_{rw} + x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ y_{rw} + x_{sr} \sin(\theta_{rw}) + y_{sr} \cos(\theta_{rw}) \\ \theta_{sr} + \theta_{rw} \end{pmatrix}$$

- Déterminer les valeurs mini et maxi de  $\phi_i$  et  $\rho_i$  pour fournir les valeurs  $(\phi_{min}, \phi_{max})$  et  $(\rho_{min}, \rho_{max})$  définissant la *ROI* qui sera ainsi une zone définie par :

$$ROI = (\rho_{min}, \rho_{max}, \phi_{min}, \phi_{max})$$

comme montré dans la figure V.7.

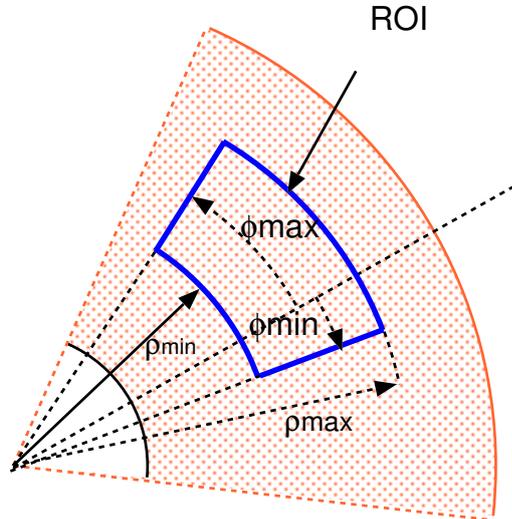


FIGURE V.7 – ROI d'un segment

En fait, il est préférable de décomposer le segment en plusieurs points de manière à être sûr d'englober l'ensemble des points du segment. La figure V.8 donne un exemple de ROI obtenu avec les sigma points lorsqu'une décomposition du segment en 5 points est réalisée.

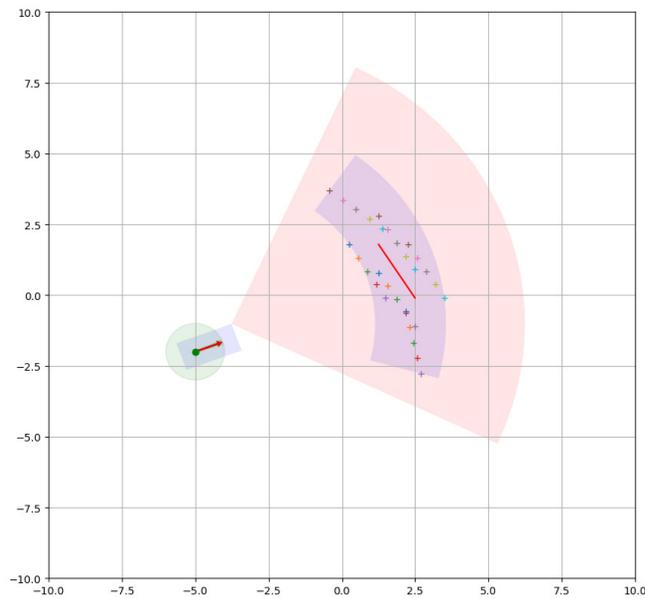


FIGURE V.8 – Exemple de ROI avec les sigma points. La position du robot présente une incertitude  $\sigma_x = \sigma_y = 1m$  et  $\sigma_\theta = 20^\circ$

## b. Évaluation du nombre d'amers dans la ROI

### 1. Vérification de l'appartenance d'un segment dans la ROI

Connaissant le segment que l'on cherche à détecter et ayant calculé sa ROI (voir § a.), l'objectif est ici de connaître le nombre d'amers de la carte susceptibles d'être dans la ROI (et donc d'être détectés) de manière à connaître la probabilité de détection du bon amer à l'intérieur. Pour cela nous allons d'abord chercher à savoir si un segment donné  $CD$  risque d'appartenir à la ROI (voir figure V.9). Ainsi, nous réalisons les opérations suivantes :

- recherche des angles  $\theta_g$  et  $\theta_d$  (voir équation V.15 et figure V.9)

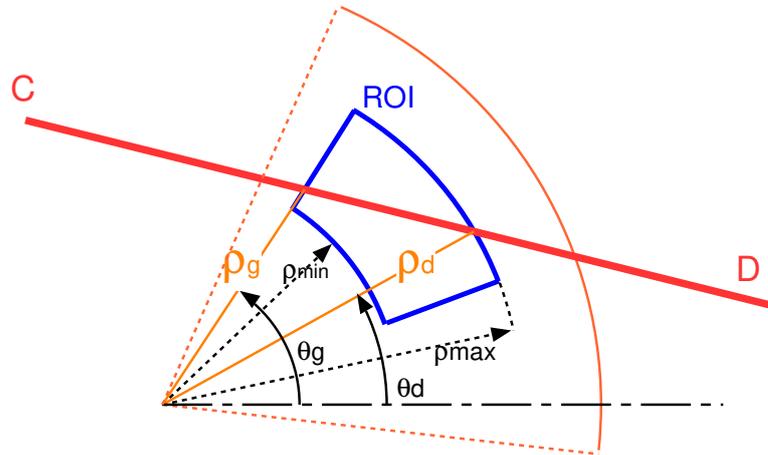


FIGURE V.9 – Critère de détection d'un segment : si  $\rho_g$  et  $\rho_d$  sont tous deux inférieurs à  $\rho_{min}$  ou supérieurs à  $\rho_{max}$  alors le segment ne sera pas détecté.

- vérification des modules  $\rho_g$  et  $\rho_d$  : si  $\rho_g$  et  $\rho_d$  sont tous deux inférieurs à  $\rho_{min}$  ou supérieurs à  $\rho_{max}$  alors le segment ne sera pas détecté.

La figure V.10 montre un exemple de fonctionnement de la procédure.

### 2. Condition de pente

En plus de la recherche des segments dans la zone, nous vérifierons *la condition de pente*. Ainsi, en plus du fait qu'un segment  $CD$  doit être dans la ROI, il faudra que sa pente (son orientation) soit compatible avec celle du segment  $AB$  en cours de détection. Si on note les angles  $\varphi_{AB}$  et  $\varphi_{CD}$  les orientations respectives de  $AB$  et  $CD$  dans le monde, nous accepterons donc un segment  $CD$  tel que :

$$(\varphi_{AB} - \varphi_{CD})^2 < S_\varphi^2 \cdot \sigma_{\theta_{rw}}^2$$

Ici,  $S_\varphi$  est un seuil au sens de la distance de Mahalanobis (par exemple  $S_\varphi = 1$ ) et  $\sigma_{\theta_{rw}}$  est l'écart-type d'orientation du robot dans le monde (la position du lidar et celle du segment étant constantes).

### 3. Filtrage des segments observables uniquement

Parmi les murs potentiellement détectables dans la ROI, nous ne choisisons que ceux qui sont observables (déjà calculés, voir § V.4.2.1). On notera  $N_c$  (pour amers compatibles) le nombre d'amers autres détectables dans la ROI.

### c. Prise en compte des inliers et outliers de détection dans la ROI

Pareillement pour ce type de triplet, selon les performances du détecteur il est nécessaire de prendre en compte :

- la probabilité de détecter un *outlier* dans la ROI (en plus des murs autres de la carte) : avec  $d_{out}$  la densité d'*outlier* potentielle par unité de surface de la ROI et  $A_{ROI}$  l'aire observable de la ROI
- la probabilité de ne rien détecter dans la ROI (alors qu'il est censé y avoir au moins le mur concerné) donnée par  $\alpha$

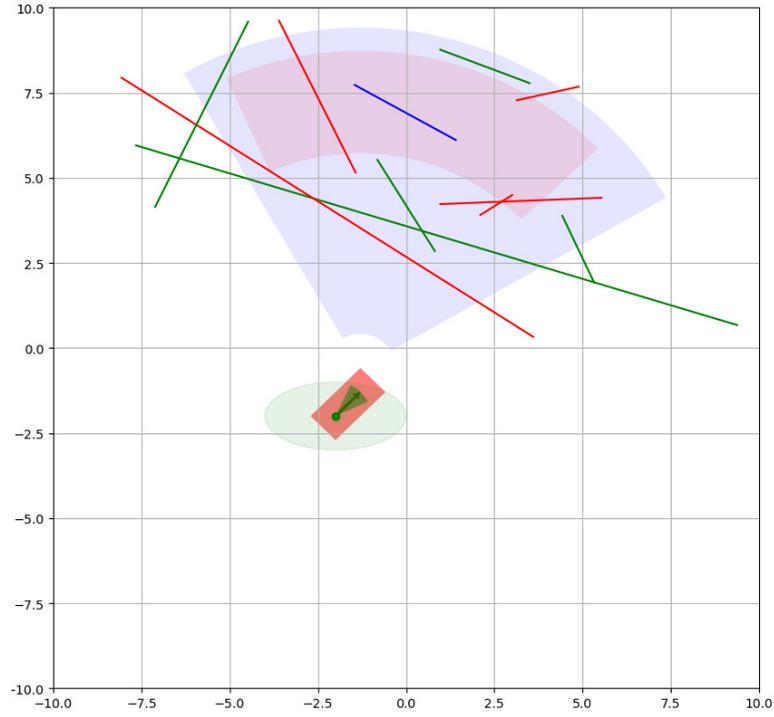


FIGURE V.10 – Exemples de fonctionnement de la procédure de détection de segments : les segments verts sont considérés non détectables et les segments rouges le sont. Le segment bleu est celui sur lequel est centrée la ROI

**d. Calcul de la confiance de détection du bon amer :  $P(db|obs = 1)$**

Ainsi, si nous considérons un triplet observable, alors, la probabilité  $P(db)$  d'une bonne détection sera aussi donnée par :

$$P(db|obs = 1) = \frac{1 - \alpha}{1 + N_c + N_o} \quad \text{et} \quad P(db|obs = 0) = 0 \quad (\text{V.18})$$

Avec :

- $\alpha$  : la probabilité de ne rien détecter dans la ROI alors qu'il est censé y avoir au moins le mur concerné
- $N_c$  : nombre d'amers de la carte se retrouvant dans la ROI
- $N_o$  : le nombre d'amers potentiellement détectables mais ne faisant pas partie de la carte. Ces amers peuvent être vus comme aussi des fausses détections (*outliers*) du détecteur.  $N_o$  sera fonction de la densité  $d_{out}$  d'amers par unité de surface dans l'espace capteur ainsi que de la contrainte de pente imposée par le détecteur. En effet, cette contrainte ne gardera que les amers dont la pente est comprise dans l'intervalle  $\pm\sigma_{\theta_{rw}}$  (voir condition de pente précédente à la page 95). Nous aurons donc :

$$N_o \approx d_{out} \cdot A_{ROI} \cdot \frac{2\sigma_{\theta_{rw}}}{2\pi}$$

Ainsi  $N_c + N_o$  représente le nombre d'amers autres potentiellement détectables à la place du mur considéré.

**V.4.2.3 Calcul de la confiance espérée  $P(O_k^+)$**

Pour ce type de triplet nous utilisons le même réseau bayésien que le triplet lidar-poteau qui est donné par la figure V.11.

Les probabilités conditionnelles  $P(obs = 1|O_k^-)$ ,  $P(db = 1|obs)$ ,  $P(O_k^+ = 1|O_k^-, db)$  et  $P(O_k^+ = 1)$  seront donc calculées de la même manière que dans la partie V.3.2.3.

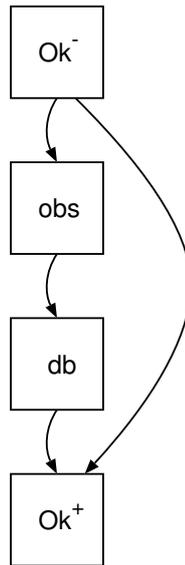


FIGURE V.11 – Réseau bayésien utilisé dans ce triplet lidar-mur

#### V.4.2.4 Calcul de précision apportée par la détection de l’amer

##### a. Objectifs

De la même manière, lorsque nous avons sélectionné un amer donné, en plus de la confiance dans sa détection il faut calculer la précision qu’il va amener. Cela va se faire aussi grâce à un filtre de Kalman avec les particularités suivantes (voir triplet lidar-poteau V.3.2.4) :

- nous utiliserons uniquement la partie mise-à-jour de l’état
- nous ne calculerons que la covariance de l’état mis à jour (pas l’espérance)

##### b. Équation de mesure

Nous supposons connaître la représentation  $(\rho_{lw}, \theta_{lw})$  du mur  $l$ , caractérisé par le segment  $AB$  (voir §VIII.2.1) comme indiqué dans la figure V.12. Nous recherchons les valeurs des mesures théoriques  $(\rho_{ls}, \theta_{ls})$  du segment

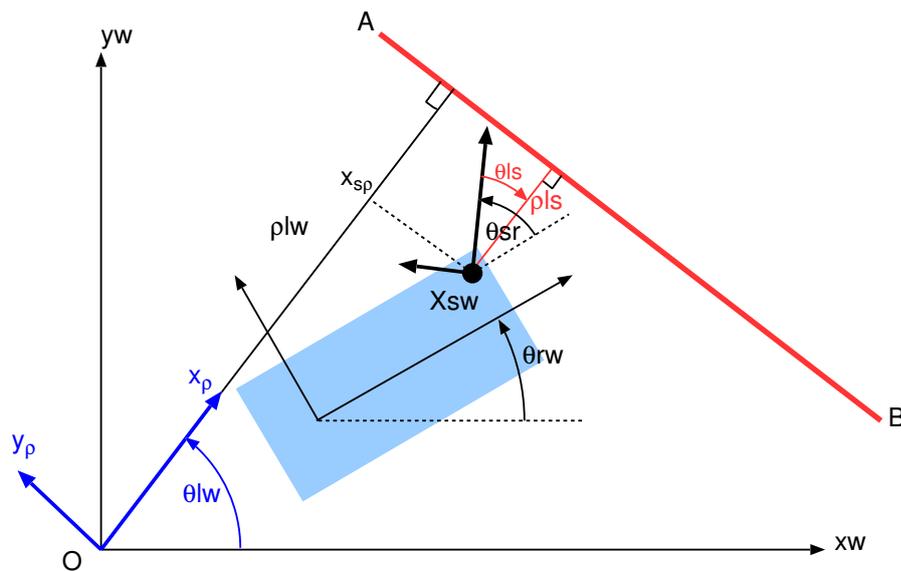


FIGURE V.12 – Paramètres de mesure d’un segment "mur" vu d’un lidar placé en  $X_{sw}$

dans le repère du capteur ( $s$ ). On voit que le point  $X_{sw}$  (coordonnées du centre du lidar dans le monde) aura

pour coordonnées  $X_{s\rho}$  dans le repère  $(x_\rho, O, y_\rho)$  telles que (il s'agit d'une simple rotation de  $\theta_{lw}$ ) :

$$X_{s\rho} = \begin{pmatrix} x_{s\rho} \\ y_{s\rho} \end{pmatrix} = \begin{pmatrix} \cos(\theta_{lw}) & \sin(\theta_{lw}) \\ -\sin(\theta_{lw}) & \cos(\theta_{lw}) \end{pmatrix} \begin{pmatrix} x_{sw} \\ y_{sw} \end{pmatrix}$$

Et que :

$$\rho_{lw} = x_{s\rho} + \rho_{ls}$$

Soit :

$$\rho_{ls} = \rho_{lw} - x_{sw} \cos(\theta_{lw}) - y_{sw} \sin(\theta_{lw})$$

De plus, on a :

$$\theta_{lw} = \theta_{sr} + \theta_{rw} + \theta_{ls} = \theta_{sw} + \theta_{ls}$$

Au aura donc :

$$\boxed{\rho_{ls} = \rho_{lw} - x_{sw} \cos(\theta_{lw}) - y_{sw} \sin(\theta_{lw}) \quad \text{et} \quad \theta_{ls} = \theta_{lw} - \theta_{sw}} \quad (\text{V.19})$$

Considérons la mesure  $z_k = (\hat{\rho}_{ls}, \hat{\theta}_{ls})^\top$  du mur  $l$ . Nous supposons que cette mesure est entachée de bruits blancs gaussiens indépendants  $w_{\rho_{ls}}$  et  $w_{\theta_{ls}}$  sur  $\rho_{ls}$  et  $\theta_{ls}$  d'écart-types  $\sigma_{\rho_{ls}}$  et  $\sigma_{\theta_{ls}}$ , soit :

$$\underline{w}_{\rho\theta_{ls}} = (w_{\rho_{ls}}, w_{\theta_{ls}})^\top \quad \text{avec pour covariance :} \quad \mathbf{R}_{\rho\theta_{ls}} = \begin{pmatrix} \sigma_{\rho_{ls}}^2 & 0 \\ 0 & \sigma_{\theta_{ls}}^2 \end{pmatrix}$$

L'équation de mesure sera donc :

$$z_k = \begin{pmatrix} \hat{\rho}_{ls} \\ \hat{\theta}_{ls} \end{pmatrix} = \mathbf{h}_{lw}(\underline{X}_{rw}) = \begin{pmatrix} h_{ls\rho} \\ h_{ls\theta} \end{pmatrix} = \begin{pmatrix} \rho_{lw} - x_{sw} \cos(\theta_{lw}) - y_{sw} \sin(\theta_{lw}) \\ \theta_{lw} - \theta_{sw} \end{pmatrix} + \begin{pmatrix} w_{\rho_{ls}} \\ w_{\theta_{ls}} \end{pmatrix} \quad (\text{V.20})$$

On sait que la pose  $\underline{X}_{sw} = (x_{sw}, y_{sw})^\top$  est fonction de la pose inconnue du robot  $\underline{X}_{rw}$  et de la pose connue du capteur sur le robot  $\underline{X}_{sr}$  (voir relation (V.7)).

### c. Mise à jour de la covariance $\mathbf{C}_k^+$

L'objectif est de déterminer la covariance  $\mathbf{C}_k^+$  de l'état (qui pour l'instant vaut  $\mathbf{C}_k^-$ ) lorsque le mur aura été détecté. Cette mise à jour sera faite par un filtre de Kalman qui utilisera les matrices jacobiennes de la fonction de mesure  $\mathbf{h}_{lw}$ . L'expression de la matrice jacobienne  $\mathbf{J}_{lw_{X_{rw}}}$  de la fonction  $\mathbf{h}_{lw}$  par rapport à  $\underline{X}_{rw}$  est nécessaire pour les équations du filtre de Kalman.

Ainsi, du fait que  $z_k = \mathbf{h}_{lw}(\underline{X}_{rw})$ , on aura :

$$\mathbf{C}_{zk} = \mathbf{J}_{lw_{X_{rw}}}^\top \mathbf{C}_{X_{rw}} \mathbf{J}_{lw_{X_{rw}}} \quad \text{avec} \quad \mathbf{J}_{lw_{X_{rw}}} = \begin{pmatrix} \frac{\partial h_{lw\rho}}{\partial x_{rw}} & \frac{\partial h_{lw\rho}}{\partial y_{rw}} & \frac{\partial h_{lw\rho}}{\partial \theta_{rw}} \\ \frac{\partial h_{lw\theta}}{\partial x_{rw}} & \frac{\partial h_{lw\theta}}{\partial y_{rw}} & \frac{\partial h_{lw\theta}}{\partial \theta_{rw}} \end{pmatrix}$$

Avec le même raisonnement précédent pour le triplet lidar-poteau, du fait que la fonction  $\mathbf{h}_{lw}(\underline{X}_{rw})$  utilise la grandeur intermédiaire  $\underline{X}_{sw}$  du capteur dans le monde, il est plus simple de calculer séparément la matrice jacobienne  $\mathbf{J}_{lw_{X_{rw}}}$  comme suit :

$$\mathbf{J}_{lw_{X_{rw}}} = \mathbf{J}_{ls_{X_{sw}}} \times \mathbf{J}_{sw_{X_{rw}}}$$

Avec :

$$\mathbf{J}_{ls_{X_{sw}}} = \begin{pmatrix} \frac{\partial h_{ls\rho}}{\partial x_{sw}} & \frac{\partial h_{ls\rho}}{\partial y_{sw}} & \frac{\partial h_{ls\rho}}{\partial \theta_{sw}} \\ \frac{\partial h_{ls\theta}}{\partial x_{sw}} & \frac{\partial h_{ls\theta}}{\partial y_{sw}} & \frac{\partial h_{ls\theta}}{\partial \theta_{sw}} \end{pmatrix} = \begin{pmatrix} -\cos(\theta_{lw}) & -\sin(\theta_{lw}) & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

$$\text{et} \quad \mathbf{J}_{sw_{X_{rw}}} = \begin{pmatrix} \frac{\partial x_{sw}}{\partial x_{rw}} & \frac{\partial x_{sw}}{\partial y_{rw}} & \frac{\partial x_{sw}}{\partial \theta_{rw}} \\ \frac{\partial y_{sw}}{\partial x_{rw}} & \frac{\partial y_{sw}}{\partial y_{rw}} & \frac{\partial y_{sw}}{\partial \theta_{rw}} \\ \frac{\partial \theta_{sw}}{\partial x_{rw}} & \frac{\partial \theta_{sw}}{\partial y_{rw}} & \frac{\partial \theta_{sw}}{\partial \theta_{rw}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -x_{sr} \sin(\theta_{rw}) - y_{sr} \cos(\theta_{rw}) \\ 0 & 1 & x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ 0 & 0 & 1 \end{pmatrix}$$

Nous rappelons que l'on cherche ici uniquement à calculer la covariance de l'état mis à jour (pas l'espérance).

La mise à jour de la covariance s'effectuera par les relations suivantes :

$$\boxed{\begin{cases} \mathbf{K}_k &= \mathbf{C}_k^- \mathbf{J}_{lw_{X_{rw}}}^T (\mathbf{J}_{lw_{X_{rw}}} \mathbf{C}_k^- \mathbf{J}_{lw_{X_{rw}}}^T + \mathbf{R}_{\rho\theta ls})^{-1} \\ \mathbf{C}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{J}_{lw_{X_{rw}}}) \mathbf{C}_k^- \end{cases}} \quad (\text{V.21})$$

avec :

$$\mathbf{R}_{\rho\theta ls} = \begin{pmatrix} \sigma_{\rho ls}^2 & 0 \\ 0 & \sigma_{\theta ls}^2 \end{pmatrix} \quad \text{et} \quad \mathbf{J}_{lw_{X_{rw}}} = \mathbf{J}_{ls_{X_{sw}}} \times \mathbf{J}_{sw_{X_{rw}}}$$

#### V.4.2.5 Calcul de pertinence d'un triplet lidar-mur : récapitulatif

Le calcul de la pertinence d'un triplet lidar-mur se fait par l'analyse de son *gain entropique* qui prendra en compte :

- le gain en confiance (évolution entre  $P(O_k^-)$  et  $P(O_k^+)$ ) expliqué dans le chapitre précédent (IV.3.3) et en utilisant la confiance espérée de la partie V.4.2.3
- le gain en précision (évolution entre  $\mathbf{C}_k^-$  et  $\mathbf{C}_k^+$ ) expliqué dans le chapitre précédent (IV.3.2) et en utilisant la précision espérée de la partie §c. à la page 98

Nous chercherons donc à calculer le gain entropique correspondant à la différence entre l'entropie initiale  $\mathbf{H}^-$  et l'entropie après détection potentielle :  $\mathbf{H}^+$ . Ce gain est donné par la même équation V.12 du triplet lidar-poteau reprise ici par l'équation V.22 :

$$\boxed{\mathbf{G} = [1 - P(d_b = 1)] \mathbf{G}_c + P(d_b = 1) [\mathbf{G}_p + \mathbf{G}_c]} \quad (\text{V.22})$$

### V.4.3 Mise à jour d'un triplet lidar-mur

#### V.4.3.1 Mise à jour de la confiance : $P(O_k^+)$

Lorsqu'une détection a été lancée nous allons avoir ou non un résultat (la détection a donné quelque chose ou non) : il s'agira donc de prendre en compte l'*évidence*  $d_b$  (une détection a été faite) pour mettre à jour la probabilité  $P(O_k^+ | d_b)$ . Pour cela nous réutiliserons le même réseau bayésien de la figure V.2.

Comme nous avons utilisé le même réseau bayésien pour les deux triplets lidar-poteau et lidar-mur, la mise à jour de la confiance  $P(O_k^+)$  sera donc donnée de la même manière comme expliqué dans la partie §V.3.3.1 du triplet lidar-poteau.

#### V.4.3.2 Mise à jour de la pose de l'état

Si la *détection a réussi* (soit  $d_b = 1$ ) alors, l'objectif est de mettre à jour l'état  $\underline{X}_k^-$  et sa covariance  $\mathbf{C}_k^-$  pour donner  $\underline{X}_k^+$  et sa covariance  $\mathbf{C}_k^+$ . On notera que  $\underline{X}_k^-$  et sa covariance  $\mathbf{C}_k^-$  ne seront pas touchés si la détection n'a pas réussi.

Nous ferons comme en section §c. mais ici, il faudra aussi mettre à jour l'état  $\underline{X}_k^-$ . Nous utiliserons les relations suivantes :

$$\boxed{\begin{cases} \mathbf{K}_k &= \mathbf{C}_k^- \mathbf{J}_{lw_{X_{rw}}}^T (\mathbf{J}_{lw_{X_{rw}}} \mathbf{C}_k^- \mathbf{J}_{lw_{X_{rw}}}^T + \mathbf{R}_{\rho\theta ls})^{-1} \\ \underline{X}_k^+ &= \underline{X}_k^- + \mathbf{K}_k (\underline{z}_k - \underline{z}_k^-) \\ \mathbf{C}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{J}_{lw_{X_{rw}}}) \mathbf{C}_k^- \end{cases}} \quad (\text{V.23})$$

La mesure  $\underline{z}_k = (\hat{\rho}_{ls}, \hat{\theta}_{ls})$  sera donnée par le détecteur de segment et la *mesure prédite*  $\underline{z}_k^-$  (voir aussi équation (V.20)) sera donnée par :

$$\underline{z}_k^- = \mathbf{h}_{lw}(\underline{X}_{rw}^-) = \begin{pmatrix} \rho_{lw} - x_{sw} \cos(\theta_{lw}) - y_{sw} \sin(\theta_{lw}) \\ \theta_{lw} - \theta_{sw} \end{pmatrix} \quad (\text{V.24})$$

Ici,  $x_{sw}, y_{sw}, \theta_{sw}$  sont les composantes de la pose (inconnue) du capteur dans le monde et seront liées à celles estimées dans  $\underline{X}_k^- = (x_{rw}, y_{rw}, \theta_{rw})^\top$  de la pose du robot dans le monde. Le lien entre  $\underline{X}_{sw}$  et  $\underline{X}_{rw}$  se fait comme dans la relation (V.5) rappelée en (V.25) :

$$\underline{X}_{sw} = \begin{pmatrix} x_{sw} \\ y_{sw} \\ \theta_{sw} \end{pmatrix} = \begin{pmatrix} x_{rw} + x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ y_{rw} + x_{sr} \sin(\theta_{rw}) + y_{sr} \cos(\theta_{rw}) \\ \theta_{sr} + \theta_{rw} \end{pmatrix} \quad (\text{V.25})$$

On aura aussi :

$$\mathbf{R}_{\rho\theta ls} = \begin{pmatrix} \sigma_{\rho ls}^2 & 0 \\ 0 & \sigma_{\theta ls}^2 \end{pmatrix}, \quad \mathbf{J}_{lwXrw} = \mathbf{J}_{lsXsw} \times \mathbf{J}_{swXrw}$$

et :

$$\mathbf{J}_{lsXsw} = \begin{pmatrix} -\cos(\theta_{lw}) & -\sin(\theta_{lw}) & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad \text{et} \quad \mathbf{J}_{swXrw} = \begin{pmatrix} 1 & 0 & -x_{sr} \sin(\theta_{rw}) - y_{sr} \cos(\theta_{rw}) \\ 0 & 1 & x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ 0 & 0 & 1 \end{pmatrix}$$

## V.5 Triplet UWB

### V.5.1 Notations

La technique UWB (Ultra Wide Band) est utilisée dans le domaine de la géolocalisation (voir II.4.1 sur les types d'amers utilisés pour la localisation). Dans nos travaux nous l'utilisons pour le positionnement d'un véhicule. Le principe de la technique est une trilatération entre un émetteur et un récepteur retournant uniquement une mesure de distance évaluée par une mesure du temps de propagation du signal. Des balises UWB réparties dans l'environnement seront considérées comme des amers de positions connues communiquant avec une balise UWB sur le robot. La même technique pourrait être utilisée dans un cas d'un convoi de véhicules avec des balises mobiles placées sur des robots.

Dans nos travaux, une balise sera assimilée à un amer de type point. Pour cela, nous constatons plusieurs similitudes entre ce triplet et le triplet de type lidar-poteau. Les notations utilisées seront par conséquent les mêmes introduites pour le triplet lidar-poteau dans la partie V.3.1 et données par la figure V.1.

### V.5.2 Analyse de pertinence d'un triplet UWB

Pour analyser la pertinence de ce type de triplet, nous utilisons la même démarche que les triplets précédents. Cela veut dire que nous allons :

1. déterminer la probabilité de l'observabilité  $P(obs)$  de la balise
2. Calculer la confiance qui sera donnée comme suit :
  - (a) Déterminer la ROI
  - (b) Calculer la confiance  $P(O_k^+)$
3. Calculer la précision apportée par la détection de la balise
4. Calculer la pertinence globale de ce triplet

#### V.5.2.1 Observabilité de la balise UWB

Nous recherchons ici quelle est la probabilité que l'amer  $A$  (une balise UWB émettrice placée dans l'environnement) se trouve dans la zone d'observation du capteur (une balise UWB réceptrice placée sur le robot). La probabilité d'observabilité  $P(obs|O_k^- = 1)$  sera donnée de la même manière que pour le triplet lidar-poteau puisqu'il s'agit encore d'un amer de type point (voir partie V.3.2.1). La seule différence entre les deux triplets est au niveau de la nature de la mesure utilisée. En effet, il s'agit pour ce type de triplet d'une mesure de distance uniquement (Range Only) entre la balise émettrice et celle réceptrice (aucune mesure d'angle n'est utilisée pour ce triplet).

### V.5.2.2 Détermination de la confiance dans la détection de la balise : $P(db|obs = 1)$

Pour calculer la confiance que nous aurons suite à une détection lancée d'une balise, nous utiliserons encore un réseau bayésien. En fait, l'aspect ambiguïté ne se pose pas pour une communication UWB. En effet, un identifiant de la balise est communiqué : il n'y donc aucune possibilité d'ambiguïté.

Pour des raisons de clarté et de simplification, nous ne détaillerons que les parties qui se distinguent de ce qui a été présenté pour les triplets précédents. Cela justifie en fait l'aspect générique de notre approche dans la mesure où c'est la même démarche qui est suivie pour tous les types de triplets.

Pour calculer la confiance dans la détection d'un triplet UWB, nous utilisons la même démarche, c'est à dire nous devons :

- Prendre en compte les éventuels inliers et outliers de détection : comme aucune ambiguïté entre les balises n'est considérée, le nombre des inliers sera par conséquent nul ( $N_c = 0$ ). Dès lors, seulement les outliers seront considérés (en fait il est possible d'avoir de faux échos, ce que traduit le terme  $N_o$ ).
- Calculer la confiance en fonction de tous les éléments calculés : si nous considérons un triplet observable, alors, la probabilité  $P(db|obs = 1)$  d'une bonne détection sera donnée par :

$$P(db|obs = 1) = \frac{1}{1 + N_o} \quad \text{et} \quad P(db|obs = 0) = 0 \quad (\text{V.26})$$

### V.5.2.3 Calcul de la confiance espérée $P(O_k^+)$

Ce paragraphe ne suscite pas un intérêt particulier. La confiance espérée  $P(O_k^+)$  sera calculée de la même manière que les triplets précédents, c'est à dire selon l'équation suivante :

$$P(O_k^+ = 1) = P(O_k^- = 1) \cdot P(obs|O_k^- = 1) \cdot P(db|obs = 1)$$

### V.5.2.4 Calcul de précision apportée par la détection de la balise

#### a. Objectifs

De la même manière, en plus de la confiance dans la détection il faut calculer la précision qui sera apportée. Cela va aussi se faire grâce à un filtre de Kalman dans un premier temps puis un UKF dans un second temps.

#### b. Équation de mesure

Nous cherchons ici à modéliser la relation  $\mathbf{h}_{lw}$  décrivant le lien entre la mesure  $\underline{z}_k = \rho$  retournée par le capteur  $s$  lorsqu'il observe la balise  $l$  de position connue  $\underline{X}_{lw}$  dans le monde et de la position du robot dans le monde  $\underline{X}_{rw}$ . En utilisant le même raisonnement de la partie **b.** à la page 86, la mesure théorique  $\underline{z}_k$  sera donnée par :

$$\underline{z}_k = \rho = \mathbf{h}_{ls}(\underline{X}_{lw}, \underline{X}_{sw}) = \sqrt{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2} \quad (\text{V.27})$$

Avec :

$$\underline{X}_{sw} = \begin{pmatrix} x_{sw} \\ y_{sw} \\ \theta_{sw} \end{pmatrix} = \mathbf{h}_{sw}(\underline{X}_{sr}, \underline{X}_{rw}) = \begin{pmatrix} x_{rw} + x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ y_{rw} + x_{sr} \sin(\theta_{rw}) + y_{sr} \cos(\theta_{rw}) \\ \theta_{sr} + \theta_{rw} \end{pmatrix} \quad (\text{V.28})$$

En supposant que cette mesure est entachée d'un bruit blanc gaussien  $\omega_\rho$  sur  $\rho$  avec pour covariance  $\mathbf{R}_\rho$ , nous aurons l'équation de mesure suivante :

$$\underline{z}_k = \rho = \mathbf{h}_{lw}(\underline{X}_{rw}) = \sqrt{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2} + \omega_\rho \quad (\text{V.29})$$

### c. Mise à jour de la covariance $\mathbf{C}_k^+$

L'objectif est de déterminer la covariance  $\mathbf{C}_k^+$  de l'état (qui vaut pour l'instant  $\mathbf{C}_k^-$ ) lorsque la balise aura été détectée. Cette mise à jour sera aussi faite par un filtre de Kalman qui utilisera les matrices jacobiniennes de la fonction de mesure  $\mathbf{h}_{lw}$  (voir partie c. à la page 88).

La mise à jour de la covariance s'effectuera par les relations suivantes :

$$\begin{cases} \mathbf{K}_k &= \mathbf{C}_k^- \mathbf{J}_{lw_{X_{rw}}}^T (\mathbf{J}_{lw_{X_{rw}}} \mathbf{C}_k^- \mathbf{J}_{lw_{X_{rw}}}^T + \mathbf{R}_\rho)^{-1} \\ \mathbf{C}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{J}_{lw_{X_{rw}}}) \mathbf{C}_k^- \end{cases} \quad (\text{V.30})$$

Avec :

$$\mathbf{J}_{lw_{X_{rw}}} = \mathbf{J}_{ls_{X_{sw}}} \times \mathbf{J}_{sw_{X_{rw}}}$$

Et :

$$\mathbf{J}_{ls_{X_{sw}}} = \begin{pmatrix} \frac{x_{sw} - x_{lw}}{\sqrt{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2}} & \frac{y_{sw} - y_{lw}}{\sqrt{(x_{lw} - x_{sw})^2 + (y_{lw} - y_{sw})^2}} & 0 \end{pmatrix}$$

$$\mathbf{J}_{sw_{X_{rw}}} = \begin{pmatrix} 1 & 0 & -x_{sr} \sin(\theta_{rw}) - y_{sr} \cos(\theta_{rw}) \\ 0 & 1 & x_{sr} \cos(\theta_{rw}) - y_{sr} \sin(\theta_{rw}) \\ 0 & 0 & 1 \end{pmatrix}$$

### V.5.2.5 Calcul de pertinence d'un triplet UWB : récapitulatif

Le calcul de la pertinence d'un triplet UWB se fait là encore par l'analyse de son *gain entropique* qui prendra en compte :

- le gain en confiance (évolution entre  $P(O_k^-)$  et  $P(O_k^+)$ ) expliqué dans le chapitre précédent (IV.3.3) et en utilisant la confiance espérée de la partie V.5.2.3
- le gain en précision (évolution entre  $\mathbf{C}_k^-$  et  $\mathbf{C}_k^+$ ) expliqué dans le chapitre précédent (IV.3.2) et en utilisant la précision espérée de la partie §c. précédente.

Nous chercherons donc à calculer le gain entropique correspondant à la différence entre l'entropie initiale  $\mathbf{H}^-$  et l'entropie après détection potentielle :  $\mathbf{H}^+$ . Ce gain est aussi donné par la relation V.22.

### V.5.3 Mise à jour d'un triplet UWB

Une fois que ce triplet est choisi comme le plus pertinent, la détection est lancée et la mise à jour est effectuée sur deux étapes :

1. Mise à jour de la confiance :  $P(O_k^+)$  : le fait que la détection a ou non donné un résultat impacte la confiance dans la pose du robot. La confiance sera donc mise à jour comme montrée précédemment dans la partie V.3.3.1 en considérant l'évidence  $d_b$  (une détection a été faite ou non).
2. Mise à jour de la pose de l'état : si la détection a réussi, la mesure qu'a effectivement donné cette détection précise l'estimation de la pose. L'objectif sera donc de mettre à jour l'état  $\underline{X}_k^-$  et sa covariance  $\mathbf{C}_k^-$  pour donner  $\underline{X}_k^+$  et sa covariance  $\mathbf{C}_k^+$ . On notera que  $\underline{X}_k^-$  et sa covariance  $\mathbf{C}_k^-$  ne seront pas touchés si la détection n'a pas réussi. Nous utiliserons les relations suivantes pour la mise à jour :

$$\begin{cases} \mathbf{K}_k &= \mathbf{C}_k^- \mathbf{J}_{lw_{X_{rw}}}^T (\mathbf{J}_{lw_{X_{rw}}} \mathbf{C}_k^- \mathbf{J}_{lw_{X_{rw}}}^T + \mathbf{R}_\rho)^{-1} \\ \underline{X}_k^+ &= \underline{X}_k^- + \mathbf{K}_k (\underline{z}_k - \underline{z}_k^-) \\ \mathbf{C}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{J}_{lw_{X_{rw}}}) \mathbf{C}_k^- \end{cases} \quad (\text{V.31})$$

La mesure  $z_k = \hat{\rho}$  sera donnée par le détecteur de la balise et la *mesure prédite*  $\bar{z}_k$  sera donnée par l'équation V.29.

La mise à jour selon la relation V.31 mais aussi sur l'espérance en gain de précision V.30 conduit à une mise à jour issue d'une linéarisation qui est particulièrement flagrante dans le cas de mesure de type *range only*. La figure V.13 illustre ce phénomène connu. Ici, l'incertitude initiale est représentée par l'ellipse bleue centrée en  $X_{sw} = (0, 0)$  et la zone de perception de la balise est donnée par le cercle noir. La balise est positionnée en  $X_{lw} = (30, 20)$ . La mise à jour par un filtre de Kalman est l'ellipse verte centrée sur l'axe entre la balise et la pose initiale. La mesure de distance est ici de  $\rho = 26m$  avec un écart-type d'imprécision de  $\sigma_\rho = 0.2m$ .

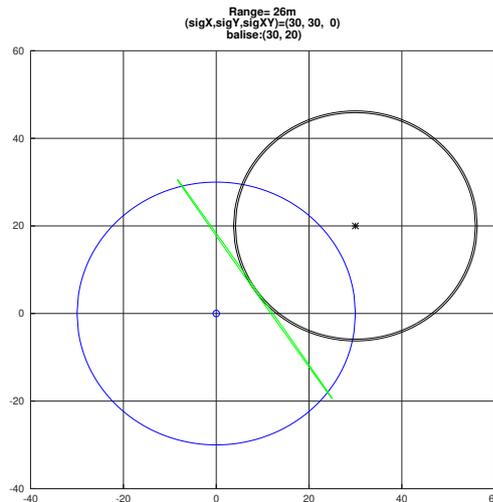


FIGURE V.13 – Illustration de la mise à jour par filtrage de Kalman dans le cas d'une mesure de type *range-only*. La balise, est située à une distance de 26m. L'estimation initiale est l'ellipse bleue et l'estimation de Kalman est l'ellipse verte. Celle-ci est correcte sur le point de linéarisation mais s'éloigne de la forme naturelle de l'anneau.

L'inconvénient de cette approche est que le processus de focalisation va très probablement conduire à une zone d'analyse très fine et finalement fausse du triplet courant. De plus, du fait du gain important en précision apporté par cette mise à jour, ce triplet a de fortes chances d'être sélectionné et d'engendrer un défaut de reconnaissance des triplets suivants. Le processus récursif (retour en arrière décrit en section IV.6) va probablement revenir pour éliminer cette détection mais au prix de quelques itérations supplémentaires.

La solution que nous avons apportée ici consiste à rechercher l'ellipse englobant probablement la partie de l'anneau qui se trouve dans la zone d'incertitude initiale (définie par la matrice de covariance  $C_k^-$ ).

Nous utilisons pour cela un filtre de type UKF (voir (Julier and Uhlmann, 1997)). Pour cela nous procédons de la manière suivante :

- Calcul des sigma points  $P^i$  issus de  $C_k^-$ ,
- Pour chaque sigma point  $P^i$ , définir le segment qui le relie avec la balise et noter les deux points  $Q_1^i$  et  $Q_2^i$  avec chaque bord de l'anneau. Soit  $\underline{V}$  le vecteur entre le sigma point  $P^i$  et la balise  $X_{lw}$  :  $\underline{V} = P^i - X_{lw}$ . On aura donc pour une mesure de distance  $\rho$  d'écart-type  $\sigma_\rho$  les deux points  $X_j^i$  (avec  $j \in \{1, 2\}$ )

$$Q_k = Q_j^i = \frac{\underline{V}}{\|\underline{V}\|} \rho \pm \sigma_\rho + X_{lw} \quad \text{avec} \quad k = 2i + j$$

- Calculer l'espérance et la matrice d'inertie de ces  $N = 2 \times i$  points  $Q_j^i$  :

$$\mu = \frac{1}{N} \sum_{k=1}^N Q_k \quad \text{et} \quad C = \frac{1}{N} \sum_{k=1}^N (Q_k - \mu)(Q_k - \mu)^\top$$

Ici,  $w_k^m$  et  $w_k^c$  sont les poids des sigma points utilisés (voir [Julier and Uhlmann \(1997\)](#)).

La figure V.14 montre l'approche proposée pour diverses configurations. L'ellipse rouge montre l'estimation réalisée. Les sigma points initiaux sont les croix bleues et points  $Q_k$  sont les astérisques en rouge. On note que l'estimation fournie est selon les coordonnées  $(x, y)$ . Pour retrouver l'angle d'orientation  $\theta$  nous utilisons un filtre de Kalman. Naturellement, il est impossible de retrouver avec une gaussienne la forme traditionnelle en "banane" de la fonction de vraisemblance mais la solution produite permet de maintenir l'intégrité de la localisation, un calcul de pertinence de ce triplet de meilleur qualité et au final une meilleure focalisation pour les triplets suivants.

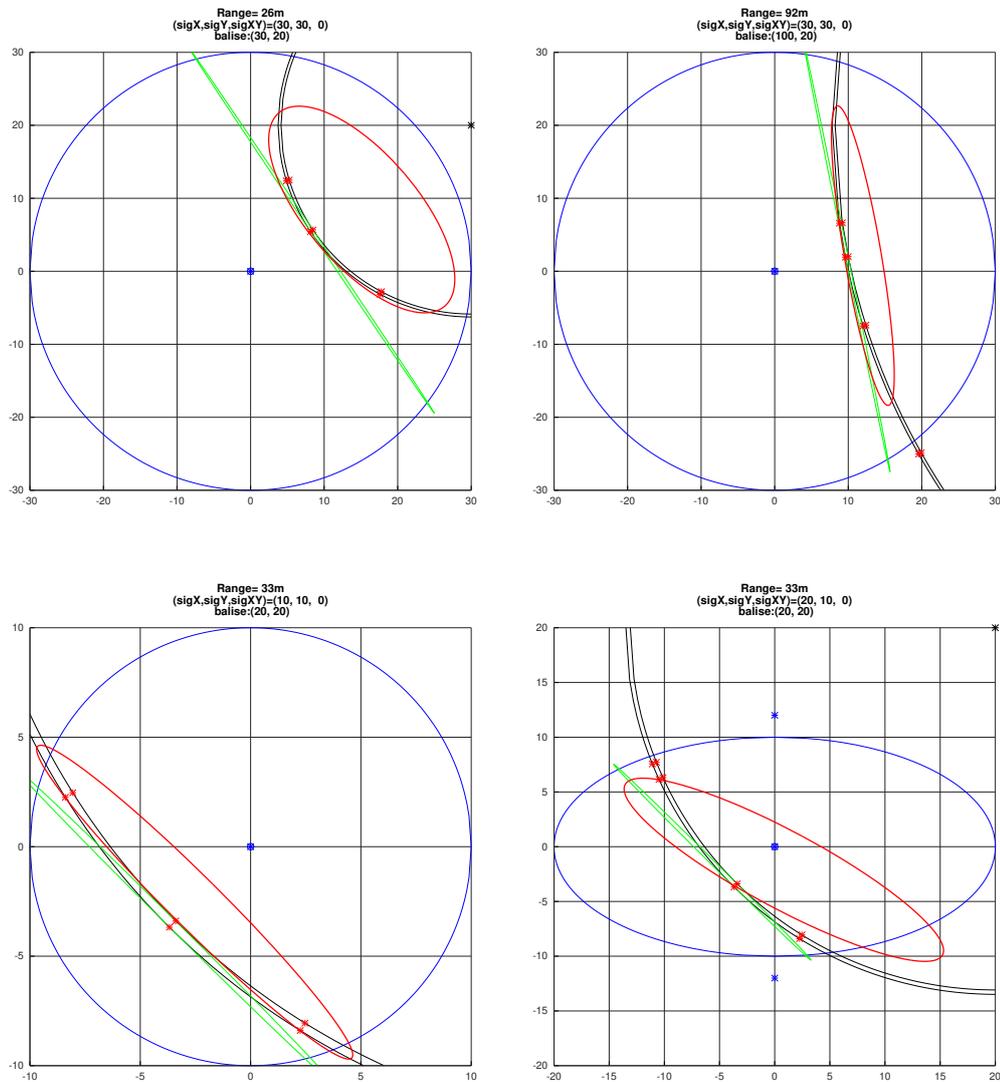


FIGURE V.14 – Illustration du calcul de l'incertitude recalculée avec l'approche proposée pour diverses configurations. L'ellipse rouge montre l'estimation réalisée. Les sigma points initiaux sont les croix bleues et points  $Q_k$  (voir texte) sont les astérisques en rouge.

## V.6 Triplet GPS

### V.6.1 Notations

Une technique très répandue dans le contexte de la localisation, est l'utilisation de données GNSS (nous appellerons ici le type de triplet : triplet GPS). Pour cela, nous allons intégrer dans notre architecture un triplet

de ce type.

L'objectif ici sera de mettre à jour l'état et sa confiance en fonction de l'état courant et de la mesure GPS  $z_k$  (incluant la mesure proprement dite mais aussi une confiance sur cette mesure).

## V.6.2 Analyse de pertinence d'un triplet GPS

Le cas de cet triplet est particulier dans notre approche. En effet, il s'agit d'un triplet dégénéré non associé à un amer. Ceci n'est pas un problème puisque c'est la pertinence globale du triplet qui compte. Les calculs utilisés pour ce type de triplet sont différents mais retournent la même métrique qui est *la pertinence globale du triplet* qui sera utilisée pour la sélection du meilleur triplet. Pour estimer la pertinence globale nous devons :

1. Calculer la précision apportée par le GPS
2. Calculer la confiance escomptée  $P$
3. Calculer la pertinence globale de ce triplet

### V.6.2.1 Calcul de précision apportée par le triplet GPS

Nous supposons que le GPS retourne un couple  $z_k = (x_{gps}, y_{gps})^\top$  avec un bruit de mesure de covariance  $\mathbf{R}_k$  (matrice  $2 \times 2$ ). Nous aurons naturellement le lien entre la mesure GPS et la pose du robot selon la relation :

$$z_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_{rw} \\ y_{rw} \\ \theta_{rw} \end{pmatrix} = \mathbf{H}\underline{X}_{rw}$$

La mise à jour de la covariance consistera à assurer le passage de  $\mathbf{C}_k^-$  à  $\mathbf{C}_k^+$ . Néanmoins, et pour simplifier, nous considérons le vecteur d'état du robot de dimension deux  $\underline{X}_k$  représentant sa pose  $x, y$  :

$$\underline{X}_k = \begin{pmatrix} x_{k_{rw}} \\ y_{k_{rw}} \end{pmatrix}$$

Pour ce faire, on fera simplement un filtrage de Kalman :

$$\begin{aligned} \mathbf{C}_k^+ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{C}_k^- \\ \mathbf{K}_k &= \mathbf{C}_k^- \mathbf{H}^\top (\mathbf{H}\mathbf{C}_k^- \mathbf{H}^\top + \mathbf{R}_k)^{-1} \end{aligned}$$

Avec  $\mathbf{H}$  et  $\mathbf{R}$  : les paramètres d'observation donnés par (avec  $\sigma_{GPS}$  l'écart-type du bruit de mesure) :

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \sigma_{GPS}^2 & 0 \\ 0 & \sigma_{GPS}^2 \end{bmatrix} \quad (\text{V.32})$$

En considérant la configuration du bras de levier pour la position du capteur sur le robot, la mesure utilisée pour la mise à jour  $z_k$  sera donnée par l'expression suivante :

$$z_k = \begin{pmatrix} x_{gps} \\ y_{gps} \end{pmatrix} = \begin{pmatrix} \cos(\theta_{rw}) & -\sin(\theta_{rw}) \\ \sin(\theta_{rw}) & \cos(\theta_{rw}) \end{pmatrix} \begin{pmatrix} x_{sr} \\ y_{sr} \end{pmatrix}$$

### V.6.2.2 Détermination de la confiance escomptée

Néanmoins d'une part nous souhaitons aussi *obtenir la confiance escomptée* de cette mise à jour mais nous souhaitons également savoir si c'est bien la bonne méthode à adopter lorsque par exemple la mesure est peu fiable (confiance faible). Par exemple, si la prédiction de la mesure est très loin de la valeur a priori, nous allons trouver une estimation entre les deux qui sera nécessairement non fiable.

On définit pour cela :

- $P_k^-$  : la probabilité que l'état initial soit bien conforme au modèle gaussien  $\mathcal{N}(X_k^-, \mathbf{C}x_k^-)$ ,
- $P_{zk}$  : la probabilité de la mesure GPS,
- $P_k^+$  : la probabilité que l'état mis à jour soit bien conforme au modèle gaussien  $\mathcal{N}(X_k^+, \mathbf{C}x_k^+)$ ,

La figure V.15 illustre le propos.

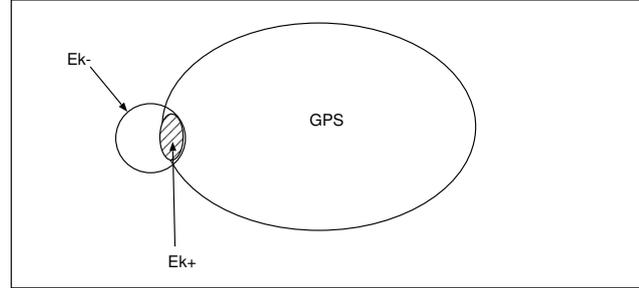


FIGURE V.15 – Détermination de  $P_k^+$  pour un triplet GPS

La difficulté est de gérer les situations suivantes :

- lorsque la prédiction de la mesure et l'état a priori sont compatibles (dans ce cas on fera un filtrage de Kalman par exemple)
- lorsque les deux sont éloignées l'une de l'autre dans ce cas, on ne gardera que la plus pertinente.

Pour cela nous devons remonter à une modélisation autre que gaussienne à la fois sur la mesure mais aussi sur l'estimation *a priori* de l'état. Ainsi, nous considérons la pdf  $p(X_k|z_k)$ . Celle-ci sera donnée dans le cas général par :

$$p(X_k|z_k) = \frac{p(z_k|X_k; X_k) \cdot p(X_k)}{p(z_k)}$$

De manière à gérer la nouvelle confiance  $P_k^+$  dans l'estimation réalisée, il sera nécessaire de gérer aussi un terme de confiance sur les pdf d'entrée, à savoir l'a priori et la mesure. On considère que chacune de ces deux pdf est la somme :

- d'une pdf gaussienne (reflétant l'état a priori ou la fonction de vraisemblance de la mesure) pondérée par une confiance (respectivement  $P_k^-$  et  $P_{zk}$ )
- d'une pdf "plancher"  $p_0^-(X_k)$  permettant de mettre en évidence le fait que les deux pdf précédentes ne recueillent pas toute la probabilité. Pour simplifier les calculs, on considère que la pdf "plancher"  $p_0^-(X_k)$  est une loi uniforme (évidemment normalisée), d'amplitude  $1/A$  et qui englobe l'espace possible de  $X_k$  et de sa mesure. On attribuera à  $A$  une valeur correspondant à l'aire d'évolution possible du robot.

Ainsi on aura :

$$\begin{cases} p(z_k|X_k; X_k) &= (1 - P_{zk}) \cdot p_0^-(X_k) + P_{zk} \mathcal{N}(z_k|X_k, \mathbf{R}_k; X_k) \\ p(X_k) &= (1 - P_k^-) \cdot p_0^-(X_k) + P_k^- \mathcal{N}(X_k^-, \mathbf{C}x_k^-) \end{cases}$$

On cherchera ainsi à caractériser la pdf  $p(X_k|z_k)$  comme étant la somme là encore d'une pdf "plancher"  $p_0^+$  (différente *a priori* de  $p_0^-(X_k)$ ) de poids  $P_k^+$  et d'une gaussienne  $\mathcal{N}(X_k^+, \mathbf{C}x_k^+)$  de poids  $P_k^+$  que l'on devra déterminer.

Ainsi, on essaiera d'identifier  $p(X_k|z_k)$  à l'expression suivante :

$$p(X_k|z_k) = (1 - P_k^+) p_0^+(X_k) + P_k^+ \mathcal{N}(X_k^+, \mathbf{C}x_k^+) \quad (\text{V.33})$$

Lorsqu'on réalise le produit  $p(z_k|X_k; X_k) \cdot p(X_k)$  on aura ainsi :

$$\begin{aligned} p(z_k|X_k; X_k) \cdot p(X_k) &= [(1 - P_{zk}) \cdot p_0^-(X_k) + P_{zk} \mathcal{N}(z_k|X_k, \mathbf{R}_k; X_k)] [(1 - P_k^-) \cdot p_0^-(X_k) + P_k^- \mathcal{N}(X_k^-, \mathbf{C}x_k^-)] \\ &= (1 - P_{zk})(1 - P_k^-) p_0^-(X_k)^2 + \end{aligned} \quad (\text{V.35})$$

$$P_k^- (1 - P_{zk}) p_0^-(X_k) \mathcal{N}(X_k^-, \mathbf{C}x_k^-) + \quad (\text{V.36})$$

$$P_{zk} (1 - P_k^-) p_0^-(X_k) \mathcal{N}(z_k|X_k, \mathbf{R}_k; X_k) + \quad (\text{V.37})$$

$$P_k^- P_{zk} \mathcal{N}(z_k|X_k, \mathbf{R}_k; X_k) \mathcal{N}(X_k^-, \mathbf{C}x_k^-) \quad (\text{V.38})$$

Chacune des relations (V.35) à (V.38) donne une hypothèse  $H_i$  de positionnement du robot. L'objectif sera donc de choisir la meilleure de ces hypothèses parmi les 3 dernières en calculant l'amplitude  $m_i$  de chaque hypothèse  $H_i$ .

De manière à choisir la meilleure, on utilisera un estimateur MAP (Maximum de vraisemblance). La confiance que l'on aura dans cette hypothèse sera l'intégrale  $I_i$  de celle-ci divisée par la somme des intégrales du numérateur, c'est-à-dire aussi celle de  $p(z_k)$  puisque  $p(X_k|z_k)$  est une gaussienne normalisée.

Pour simplifier les calculs, on considère que la pdf "plancher"  $p_0^-(X_k)$  est une loi uniforme (évidemment normalisée), d'amplitude  $1/A$  et qui englobe l'espace possible de  $X_k$  et de sa mesure. Nous ferons donc la somme de l'intégrale de chacune des relations (V.35) à (V.38).

On attribuera à  $A$  une valeur correspondant à l'aire d'évolution possible du robot. Une pondération des hypothèses est donnée comme suit :

#### a. Hypothèse 0 : relation (V.35)

Calcul de l'intégrale :

$$I_0 = \int (1 - P_{zk})(1 - P_k^-) p_0^-(X_k)^2 dX_k = \frac{1}{A} (1 - P_{zk})(1 - P_k^-) \underbrace{\int p_0^-(X_k) dX_k}_1$$

Soit :

$$I_0 = \frac{1}{A} (1 - P_{zk})(1 - P_k^-)$$

**Calcul du maximum :** Comme il s'agit de l'hypothèse plancher, on ne considérera pas celle-ci, il ne sera donc pas nécessaire d'en calculer le maximum qui de toutes façon relève d'une constante.

#### b. Hypothèse 1 : relation (V.36)

Il s'agit ici de l'hypothèse du positionnement *a priori* du robot.

Calcul de l'intégrale :

$$I_1 = \int P_k^- (1 - P_{zk}) p_0^-(X_k) \mathcal{N}(X_k^-, \mathbf{C}x_k^-) dX_k = P_k^- (1 - P_{zk}) \underbrace{p_0^-(X_k)}_{\frac{1}{A}} \underbrace{\int \mathcal{N}(X_k^-, \mathbf{C}x_k^-) dX_k}_1$$

Soit :

$$I_1 = \frac{1}{A} P_k^- (1 - P_{zk})$$

**Calcul du maximum :**

On sait que le maximum d'une gaussienne  $\mathcal{N}(\mu, \mathbf{C})$  vaut :  $\frac{1}{\sqrt{(2\pi)^n |\mathbf{C}|}}$ . Donc le maximum de l'hypothèse vaudra :

$$m_1 = \frac{I_1}{\sqrt{(2\pi)^n |\mathbf{C}x_k^-|}} = \frac{P_k^- (1 - P_{zk})}{A \sqrt{(2\pi)^n |\mathbf{C}x_k^-|}}$$

**c. Hypothèse 2 : relation (V.37)**

Il s'agit ici de l'hypothèse de la mesure elle-même.

**Calcul de l'intégrale :**

$$\int P_{zk}(1 - P_k^-) p_0^-(X_k) \mathcal{N}(z_k | X_k, \mathbf{R}_k; X_k) dX_k = \frac{1}{A} P_{zk}(1 - P_k^-)$$

**Attention :** on est sûr que  $\int p(X_k) dX_k = 1$ , par contre  $\int p(z_k | X_k, \mathbf{R}_k; X_k) dX_k$  ne vaut pas nécessairement 1 dans le cas général (ce n'est pas une pdf mais une fonction de vraisemblance). Dans le cas qui nous intéresse, nous pouvons montrer que cette somme vaut cependant bien 1. Donc :

$$I_2 = \frac{1}{A} P_{zk}(1 - P_k^-)$$

**Calcul du maximum**

Comme précédemment, nous aurons :

$$m_2 = \frac{I_2}{\sqrt{(2\pi)^n |\mathbf{R}|}} = \frac{P_{zk}(1 - P_k^-)}{A \sqrt{(2\pi)^n |\mathbf{R}|}}$$

**d. Hypothèse 3 : relation (V.38)**

Il s'agit ici de l'hypothèse où l'état a priori est compatible avec la mesure. Ce sera donc la fusion par un filtre de Kalman.

**Calcul de l'intégrale :**

Il faut calculer :

$$I_3 = \int P_k^- P_{zk} \mathcal{N}(z_k | X_k, \mathbf{R}_k; X_k) \mathcal{N}(X_k^-, \mathbf{C}x_k^-) dX_k$$

On sait que le produit des deux gaussiennes donnera une gaussienne dénormalisée avec un facteur  $k_{xz}$  :

$$\underbrace{\mathcal{N}(z_k | X_k, \mathbf{R}_k; X_k)}_{g_1(X_k)} \underbrace{\mathcal{N}(X_k^-, \mathbf{C}x_k^-)}_{g_2(X_k)} = k_{xz} \mathcal{N}(X_k^+, \mathbf{C}x_k^+)$$

Avec (voir annexe VIII.3) :

$$k_{xz} = \sqrt{(2\pi)^n |\mathbf{C}x_k^+|} \cdot g_1(X_k^+) \cdot g_2(X_k^+)$$

Ici,  $\mathcal{N}(X_k^+, \mathbf{C}x_k^+)$  sera la gaussienne fournie par le filtre de Kalman ayant pour a priori sur l'état  $\mathcal{N}(X_k^-, \mathbf{C}x_k^-)$  et comme mesure  $z_k$  de covariance  $\mathbf{R}_k$ . Ainsi :

$$I_3 = \int P_k^- P_{zk} \mathcal{N}(z_k | X_k; X_k) \mathcal{N}(X_k^-, \mathbf{C}x_k^-) dX_k = P_k^- P_{zk} k_{xz}$$

Donc :

$$I_3 = P_k^- P_{zk} k_{xz} \quad \text{avec} \quad k_{xz} = \sqrt{(2\pi)^n |\mathbf{C}x_k^+|} \cdot g_1(X_k^+) \cdot g_2(X_k^+)$$

**Calcul du maximum :**

Comme précédemment, le maximum vaudra :

$$m_3 = \frac{I_3}{\sqrt{(2\pi)^n |\mathbf{C}x_k^+|}} = \frac{P_k^- P_{zk} \sqrt{(2\pi)^n |\mathbf{C}x_k^+|} \cdot g_1(X_k^+) \cdot g_2(X_k^+)}{\sqrt{(2\pi)^n |\mathbf{C}x_k^+|}}$$

Soit :

$$m_3 = P_k^- P_{zk} \cdot g_1(X_k^+) \cdot g_2(X_k^+)$$

**e. Calcul du poids de l'hypothèse retenue** L'hypothèse  $H_i$  retenue sera celle qui présentera la meilleure amplitude  $m_i$ . On retiendra ensuite :

- la gaussienne initiale  $\mathcal{N}(X_k^-, \mathbf{C}x_k^-)$  dans le cas de l'hypothèse 1 de l'équation V.36
- la gaussienne de mesure  $\mathcal{N}(z_k | X_k, \mathbf{R}_k; X_k)$  dans le cas de l'hypothèse 2 de l'équation V.37
- la gaussienne résultant du filtrage de Kalman  $\mathcal{N}(X_k^+, \mathbf{C}x_k^+)$  dans le cas de l'hypothèse 3 de l'équation V.38

Après avoir calculé les intégrales  $I_0, I_1, I_2$  et  $I_3$  des hypothèses des équations (V.35) à (V.38), il faut à présent calculer la confiance dans cette hypothèse. Pour cela on utilisera l'intégrale  $I_i$  de l'hypothèse  $i$  considérée que l'on rapportera à la somme globale ( la somme de l'intégrale de chacune des relations (V.35) à (V.38)).

La confiance est ainsi donnée par l'équation V.39.

$$P_k^+ = \frac{I_i}{I_0 + I_1 + I_2 + I_3} \quad (\text{V.39})$$

### V.6.2.3 Calcul de pertinence d'un triplet GPS : récapitulatif

Le calcul de la pertinence d'un triplet GPS se fait par l'analyse de son *gain entropique* qui prendra en compte :

- le gain en confiance (évolution entre  $P_k^-$  et  $P_k^+$ ) en utilisant la confiance espérée donnée par l'équation V.39
- le gain en précision (évolution entre  $\mathbf{C}_k^-$  et  $\mathbf{C}_k^+$ ) : pour ce faire, en utilisant la précision espérée de la partie §V.6.2.1 à la page 105

Nous chercherons donc à calculer le gain entropique correspondant à la différence entre l'entropie initiale  $\mathbf{H}^-$  et l'entropie après détection potentielle :  $\mathbf{H}^+$ . Ce gain est donné dans le chapitre précédent par la relation IV.8 à la page 71.

### V.6.3 Mise à jour d'un triplet GPS

La mise à jour du triplet est effectuée comme expliqué dans la partie e..

Les figures de V.16 à V.18 montrent quelques résultats. Les cercles noir et bleu correspondent respectivement à l'état a priori et à la mesure GPS. L'objectif est de trouver l'état de mise à jour donné par le cercle rouge. La confiance sera la confiance maximale des confiances suivantes :

- $P_{km}$  : la confiance de l'état a-priori (hypothèse 1)
- $P_{gps}$  : la confiance de la mesure GPS (hypothèse 2)
- $P_{kp}$  : la confiance résultante du filtrage de Kalman (hypothèse 3)

Dans les deux premières figures de la première ligne, on remarque que la prédiction de la mesure et l'état a priori sont éloignées. Dans ce cas, on garde la plus pertinente (l'état a priori dans la figure V.16 et la mesure GPS dans la figure V.17). Pour les deux autres (figure V.18), la prédiction de la mesure et l'état a priori sont compatibles. Dans ce cas, on prend la résultante Kalman.

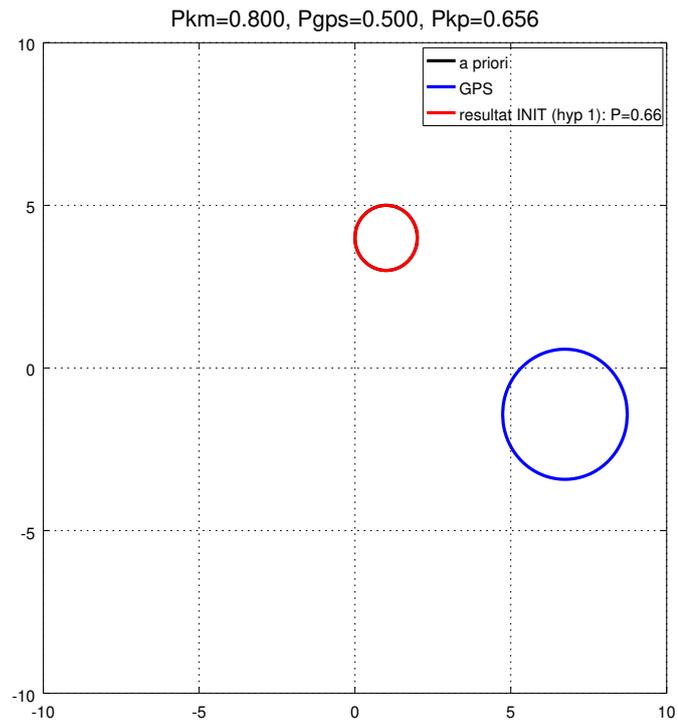


FIGURE V.16 – **Hypothèse 1** : l'estimation finale (cercle rouge) est superposée à l'estimation initiale (cercle noir).

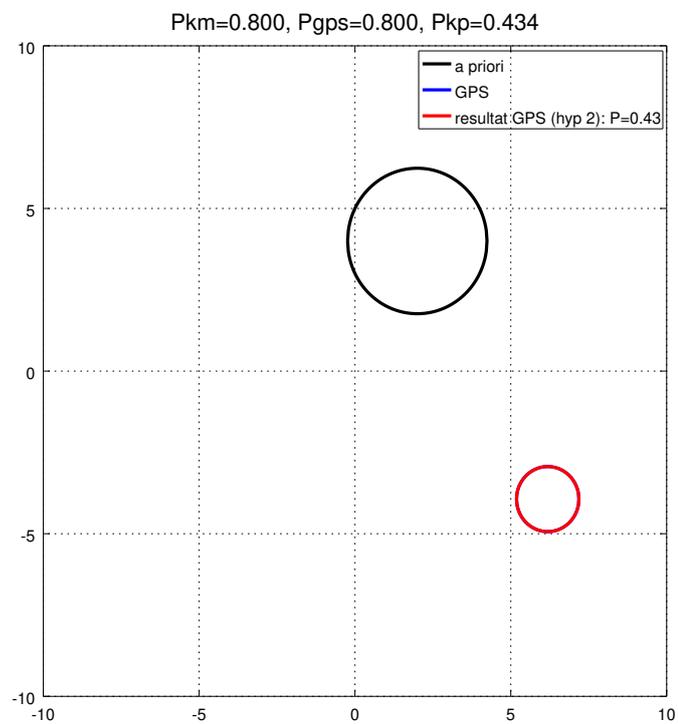


FIGURE V.17 – **Hypothèse 2** : l'estimation finale (cercle rouge) est superposée à la mesure GPS (cercle bleu).

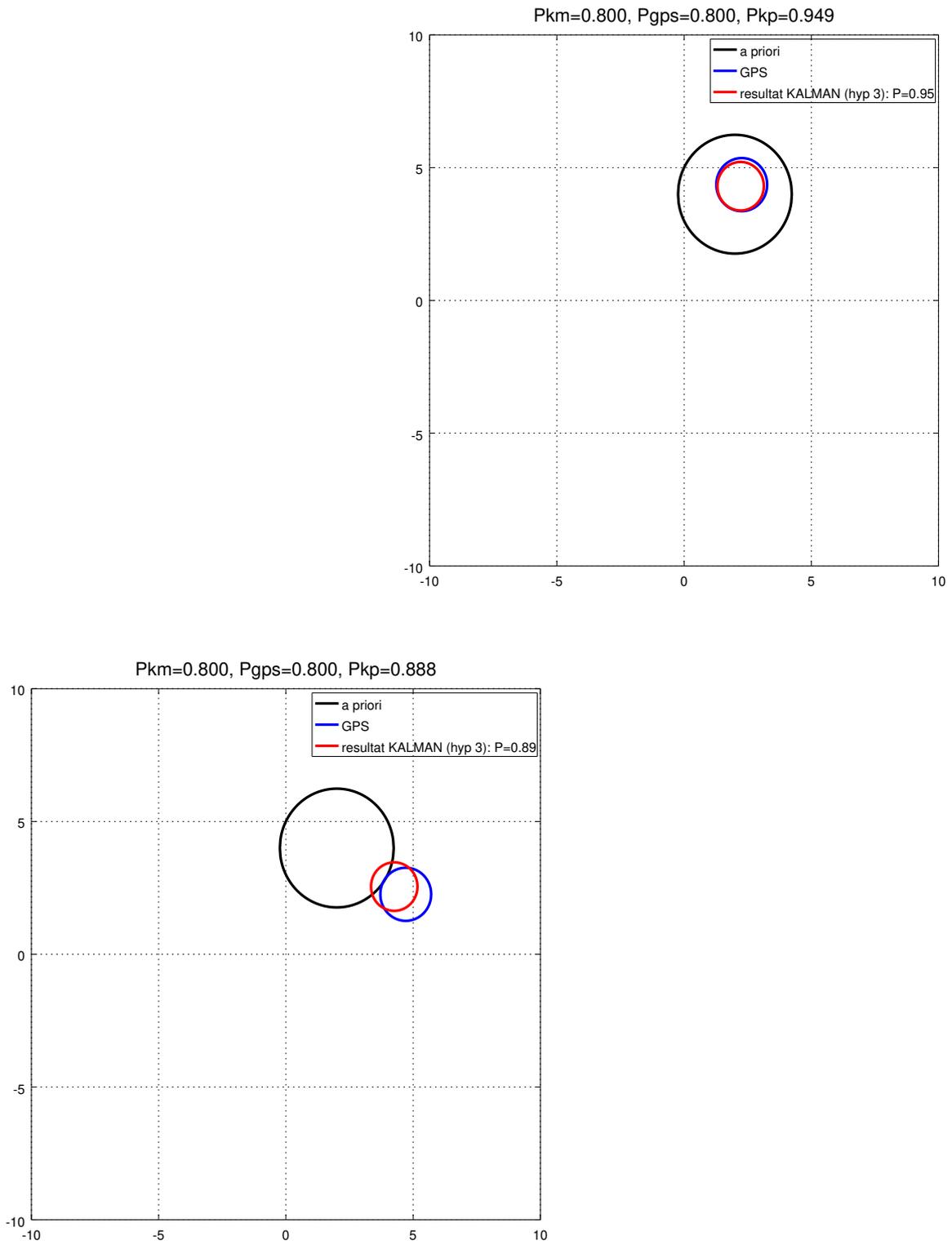


FIGURE V.18 – **Hypothèse 3** : l'estimation finale (cercle rouge) est donnée par un filtrage de Kalman en l'état précédent (cercle noir) et la mesure  $z_k$  (cercle bleu) .

## V.7 Conclusion

Dans ce chapitre, une description des triplets perceptifs que nous avons développés a été décrite. La méthode de présentation des triplets à caractère répétitif est utilisée de manière à montrer l'aspect générique de l'approche. Pour chaque triplet, nous avons introduit les étapes utilisées pour évaluer la pertinence du triplet ensuite nous avons présenté comment l'état du robot (composé de la position du robot, la matrice de covariance et la confiance) est mis à jour. La diversité des triplets justifie l'aspect multi sensoriel de notre approche. Dans le chapitre suivant, nous présentons les résultats de nos travaux.

# Chapitre VI

## Résultats architecture de fusion pour la localisation

### Sommaire

---

VI.1	Introduction	114
VI.2	Conditions de simulation pour les tests à 3 et 4 murs	114
VI.3	Résultats avec 3 murs	115
VI.4	Résultats avec 4 murs dont une ambiguïté	118
VI.5	Résultats avec 15 murs	124
VI.5.1	Cas "simple" avec 15 murs	124
VI.5.2	Tests quantitatifs et qualitatifs	127
VI.6	Résultats sur le site PAVIN	130
VI.6.1	Résultats bruts sur PAVIN	130
VI.6.2	Tests quantitatifs et qualitatifs	132
VI.7	Prise en compte d'un récepteur GNSS	135
VI.7.1	Scénario 1 : grande incertitude initiale et GPS fiable	135
VI.7.2	Scénario 2 : grande incertitude initiale et GPS peu fiable	136
VI.7.3	Scénario 3 : incertitude initiale moyenne et GPS peu fiable	136
VI.7.4	Scénario 4 : incertitude initiale bonne mais peu fiable et GPS fiable	137
VI.7.5	Bilan sur l'ajout du triplet GNSS	137
VI.8	Prise en compte de balises UWB	138
VI.8.1	Situation canonique une balise et un mur avec mise à jour de type Kalman	138
VI.8.2	Situation avec une balise et 15 murs et mise à jour de type Kalman	138
VI.8.3	Illustration des effets de la linéarisation Kalman	138
VI.8.4	Simulation UWB, lidar-mur, GPS avec pose initiale imprécise	140
VI.8.5	Simulation UWB, lidar-mur, GPS avec pose initiale précise mais fausse	140
VI.8.6	Bilan sur l'ajout du triplet UWB	144
VI.9	Conclusion	145

---

## VI.1 Introduction

Dans ce chapitre, nous présenterons les résultats de notre approche descendante appliquée à la localisation d'un robot mobile à partir de mesures issues de plusieurs capteurs et d'une carte de l'environnement. Cette carte contient l'emplacement des amers dans le repère du monde. Trois critères principaux doivent être assurés continuellement par notre algorithme :

- une bonne précision de l'estimation
- une estimation fiable et intègre (la position réelle du véhicule se situe dans la zone d'incertitude de la position estimée)
- une fusion de donnée économe qui, au lieu de fusionner un grand nombre d'information, préfère sélectionner en amont les plus prometteuses au vu d'un objectif donné.

Cette méthode de localisation a été implémentée et testée en situations simulées statiques. Plusieurs situations sont mises en œuvre afin de voir le comportement global de l'application. Des résultats sont montrés afin de conclure sur la pertinence de notre algorithme de localisation.

L'objectif à atteindre est le suivant :

- Probabilité de confiance supérieure à 0.85
- Incertitude sur les axes  $x$  et  $y$  de 10cm
- Incertitude sur l'angle d'orientation  $\theta$  de  $1^\circ$

## VI.2 Conditions de simulation pour les tests à 3 et 4 murs

Dans les différentes simulations suivantes voici les différentes hypothèses ou conditions qui vont être envisagées. Il n'y aura que des triplets de type lidar-mur car il n'y a qu'un seul capteur (Lidar) et un seul détecteur (segment). Les différents triplets seront uniquement fonction du numéro du mur (amer).

Les paramètres caractérisant le capteur (Lidar) sont :

- le capteur est positionné sur le repère de référence du robot (aucun bras de levier)
- angle d'ouverture de  $180^\circ$
- distance de détection comprise entre 10m et 30m
- bruit de mesure :  $\sigma_\rho = 20\text{cm}$  et  $\sigma_\phi = 2^\circ$

Les paramètres caractérisant les qualités du détecteur sont :

- $\alpha = 0.05$  ce qui représente 5% de chance de ne pas détecter alors qu'un amer est présent dans la zone.
- $N_o = 0.15$  ce qui représente 15% de chance de détecter quelque chose alors qu'il n'y a rien dans la zone.

Les conditions initiales sont les suivantes :

- la position réelle du robot est située au vecteur de coordonnées suivantes :  $\underline{X} = (30, 0, 0)^\top$  ce qui signifie que le robot est positionné en  $x = 30$ ,  $y = 0$  et  $\theta = 0$
- la position estimée du robot est initialisée au vecteur de coordonnées suivantes :  $\underline{X} = (15, 0, 0)^\top$  ce qui signifie que la position estimée initiale se situe en  $x = 15$ ,  $y = 0$  et  $\theta = 0$  soit 15m en arrière par rapport à sa position réelle.
- les incertitudes sur la position initiale sont fixées à  $\sigma_x = \sigma_y = 15\text{m}$  et  $\sigma_\theta = 5^\circ$ .
- la probabilité initiale liée à la confiance est fixée à 0.9.

### VI.3 Résultats avec 3 murs

Dans cette section nous présentons les résultats de l'approche proposée dans un cas simple. En effet, dans le contexte envisagé uniquement 3 murs sont présents dans l'environnement. Ainsi, cette configuration initiale est présentée sur la figure VI.1.

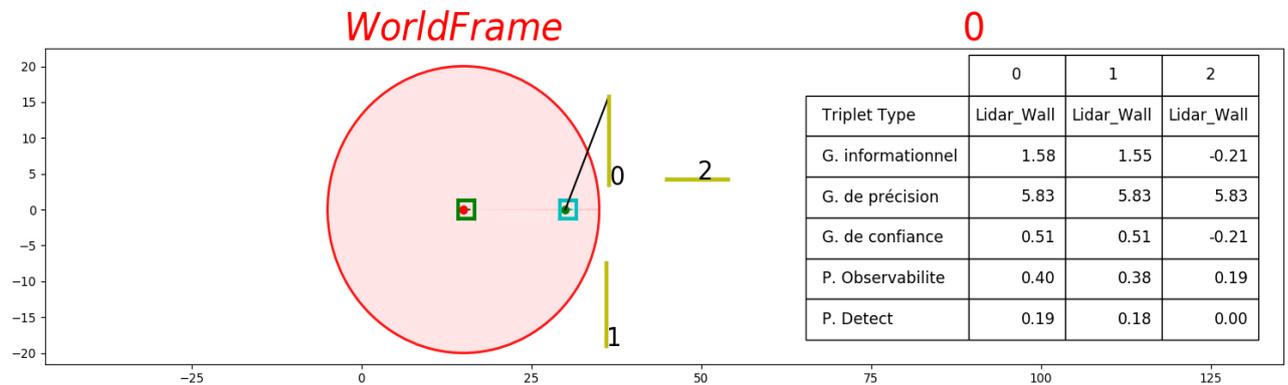


FIGURE VI.1 – Configuration initiale avec 3 murs. Sur la gauche, en cyan, la position réelle du robot, en vert la position initiale estimée du robot et en rouge l'ellipse d'incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités et gains liés à chacun des amers (ici 3 murs) dans la situation donnée.

Dans cette configuration la table des différents gains calculés (confiance, précision et informationnel) est présentée ainsi que les probabilités de détection pour chacun des triplets (ici chacun des murs). Il est important de noter qu'à l'état initial les triplets apportent tous le même gain en précision car l'incertitude initiale est la même dans toutes les directions. Par contre les probabilités d'observabilité et de détecter le bon mur sont différentes en particulier pour le mur 2 car celui-ci n'est que très peu observable dans la configuration donnée et surtout hors de portée du télémètre laser (portée limitée à 30m dans cet exemple). Les murs 0 et 1 ont quasiment les mêmes résultats. Le critère du gain informationnel privilégie le mur 0 car il est un peu plus dans l'axe que le mur 1.

Une fois que le triplet a été sélectionné (le mur 0), la phase de détection est exécutée. Dans un premier temps, la fenêtre de recherche est définie. La position de cette fenêtre est calculée en fonction de la position prévue de l'amer recherché et de l'incertitude courante de position du robot (voir figure VI.2).

Le mur détecté dans cette fenêtre, et surtout compatible avec le mur attendu, est représenté en rouge sur la figure VI.2. Il est également indiqué sur cette figure la probabilité liée à la confiance sur l'estimation de la nouvelle pose par rapport à la vraie position à l'issue de cette détection.

La détection ayant réussi, le processus de reconnaissance va être réitéré après la phase de mise à jour de la position estimée pour la détection d'un nouvel amer (profondeur de l'arbre de recherche égale à 1). Sur la figure VI.3, il est facile d'observer la nouvelle position estimée du robot (très proche de la vérité) et son ellipse d'incertitude. Cette ellipse est cohérente par rapport à la détection précédente car le mur 0 (amer précédent) a donné des informations de position suivant l'axe horizontal mais très peu suivant l'axe vertical. Pour atteindre l'objectif fixé, il faut réduire cette incertitude verticale. Parmi les amers encore observables (murs 1 et 2), la table de droite de la figure VI.3 indique bien que l'amer 2 est le plus pertinent pour atteindre l'objectif en particulier au niveau du gain de précision.

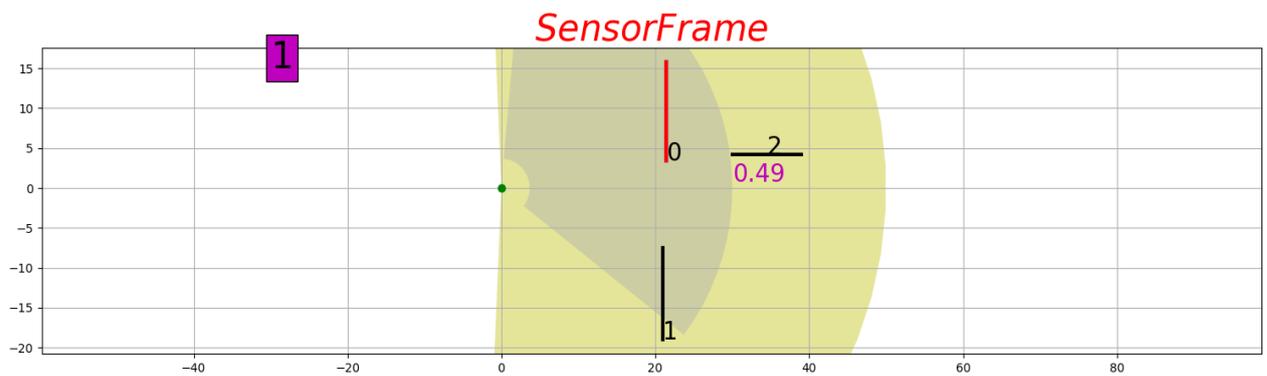


FIGURE VI.2 – Configuration initiale avec 3 murs dans l’espace capteur. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

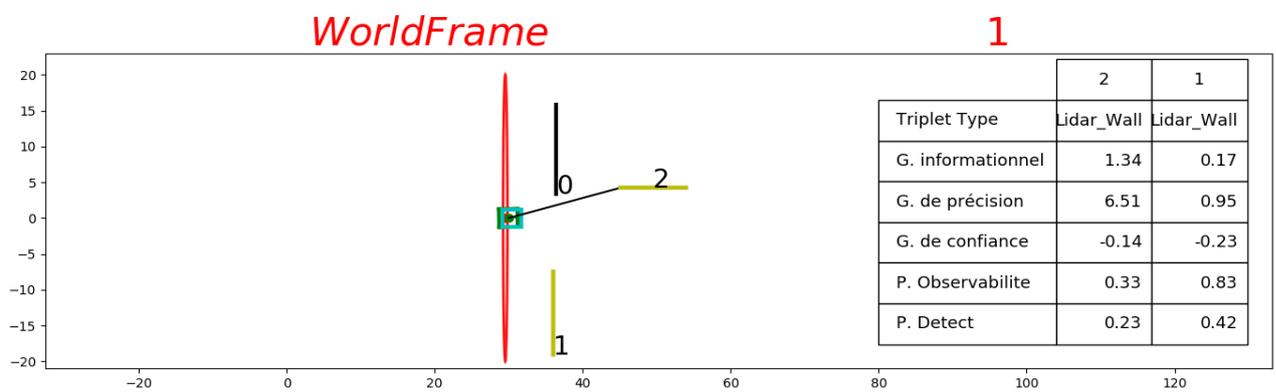


FIGURE VI.3 – Configuration avec 3 murs à la profondeur 1. Sur la gauche, en cyan, la position réelle du robot, en vert la position estimée du robot et en rouge l’ellipse d’incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici 2 murs) dans la situation donnée.

La figure VI.4 représente la même situation dans l’espace capteur. La fenêtre de recherche autour du mur 2 (amer sélectionné) est représentée en gris et le résultat de la détection en rouge. La probabilité liée à la confiance sur l’estimation de la nouvelle pose par rapport à la vraie position à l’issue de cette détection est également donnée.

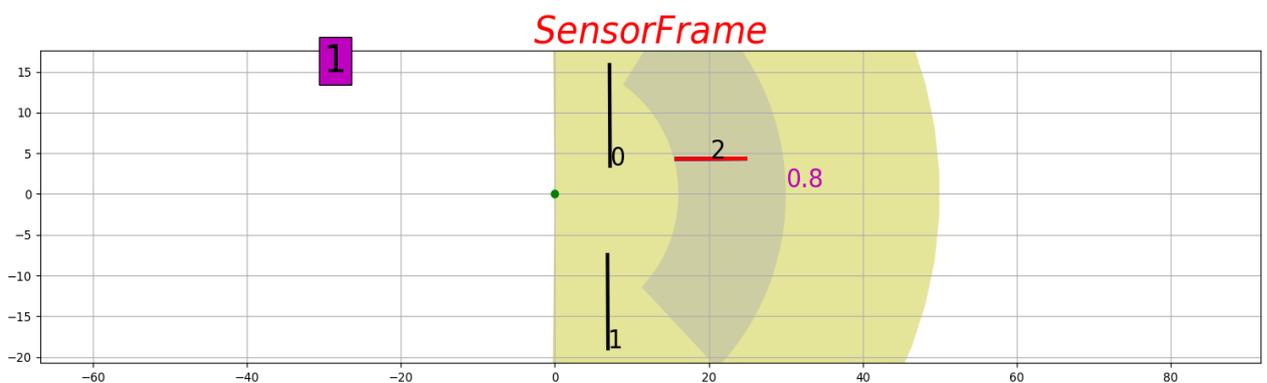


FIGURE VI.4 – Situation dans l’espace capteur après 1 détection. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

Après cette nouvelle bonne détection, le processus est renouvelé (profondeur 2 de l'arbre de recherche). Le résultat de la mise à jour de l'état est présenté sur la figure VI.5. L'ellipse d'incertitude autour de la position estimée est très petite. Il ne reste plus qu'un amer observable (mur 1) dans la situation courante et il va donc être choisi pour atteindre l'objectif final. La table présentée sur la droite montre que cet amer est parfaitement observable de la position courante et surtout qu'il apportera un gain informationnel non négligeable sur les aspects précision et confiance.

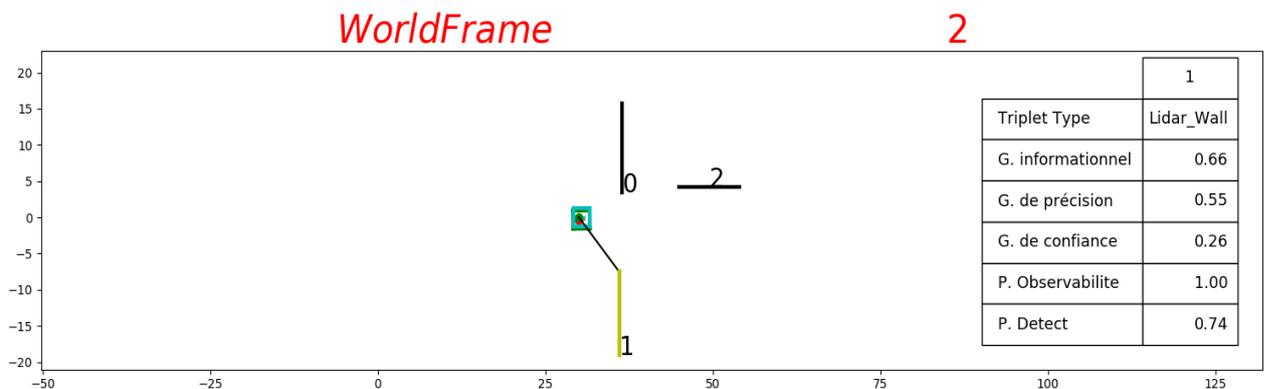


FIGURE VI.5 – Configuration avec 3 murs à la profondeur 2. Sur la gauche, en cyan, la position réelle du robot, en vert la position estimée du robot et en rouge l'ellipse d'incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici 1 mur) dans la situation donnée.

La figure VI.6 représente la même situation dans l'espace capteur. La fenêtre de recherche autour du mur 1 (amer sélectionné) est représentée en gris et le résultat de la détection en rouge. Il est important de noter que d'une part la fenêtre de recherche est largement réduite compte tenu de l'incertitude courante liée à l'estimation de la pose du robot car nous savons ce que nous cherchons et où nous devons le chercher (focalisation spatiale). D'autre part le segment détecté doit être compatible en terme de paramètres ( $\rho$  et  $\phi$ ) par rapport à ceux attendus du mur 1 dans cette configuration (focalisation paramétrique).

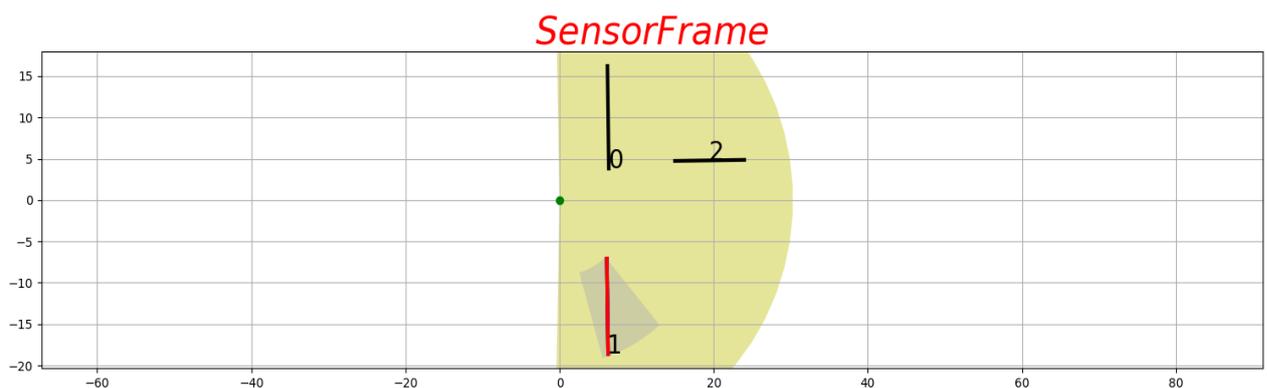


FIGURE VI.6 – Situation dans l'espace capteur après 2 détections. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

La figure VI.7 montre le résultat final du processus de localisation dans cette situation avec 3 murs dans l'environnement. La position initiale et son incertitude initiale sont représentées respectivement en rouge et en bleu. La vraie position et la position estimée finale sont représentées en cyan et en noir. La figure de droite donne un zoom de la situation finale avec les incertitudes à 1 sigma (en rouge) et à 3 sigma (en noir) autour de la position finale estimée. L'estimation finale est parfaitement intègre.

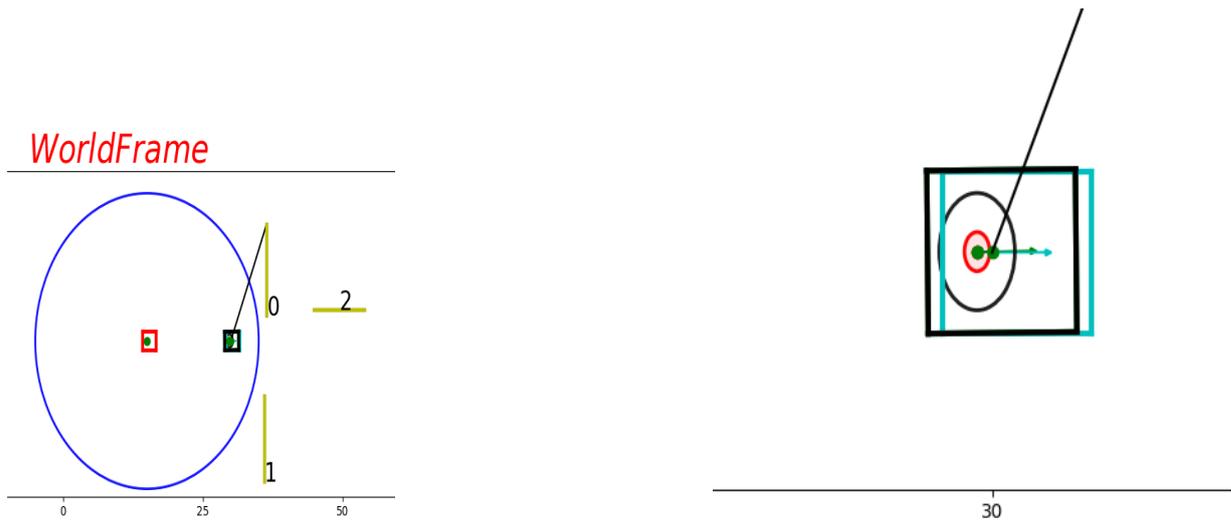


FIGURE VI.7 – Configuration finale avec 3 murs. En cyan, la position réelle du robot, en vert la position estimée du robot et en rouge la situation initiale du robot avec l'ellipse d'incertitude à un sigma autour de la position initiale estimée en bleu. La figure de droite donne la configuration finale zoomée.

#### VI.4 Résultats avec 4 murs dont une ambiguïté

Dans cette section nous présentons les résultats de l'approche proposée dans un cas un peu plus complexe car il y a la présence d'une ambiguïté possible entre deux murs. En effet, dans le contexte envisagé, 4 murs sont présents dans l'environnement mais les murs 0 et 1 peuvent être confondus suivant la pose initiale du robot.

Cette configuration initiale est présentée sur la figure VI.8.

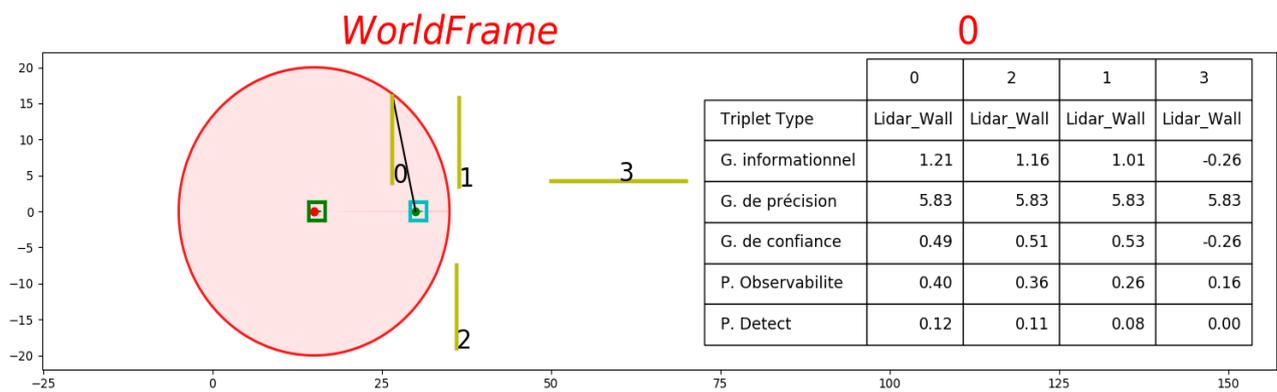


FIGURE VI.8 – Configuration initiale avec 4 murs. Sur la gauche, en cyan, la position réelle du robot, en vert la position initiale estimée du robot et en rouge l'ellipse d'incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici 4 murs) dans la situation donnée.

Dans cette configuration, la table des différents gains calculés (confiance, précision et gain informationnel) est présentée ainsi que les probabilités de détection pour chacun des triplets (ici chacun des murs). Il est important de noter qu'à l'état initial, les triplets apportent tous le même gain en précision car l'incertitude initiale est la même dans toutes les directions. Par contre les probabilités d'observabilité et de détecter le bon mur sont

différentes en particulier pour le mur 3 car celui-ci n'est que très peu observable dans la configuration donnée et surtout hors de portée du télémètre laser (portée limitée à 30m dans cet exemple). Dans cette configuration le mur 0 est sélectionné de façon préférentielle car il a une meilleure probabilité d'être observé que les autres murs. L'aspect ambiguïté est tout relatif dans l'état actuel car la zone de recherche du mur 0 est tellement grande (grande incertitude initiale) que les amers 0, 1 et 2 sont présents dans cette zone. La situation est la même pour les fenêtres de recherche associées au mur 1 et 2. La figure VI.9 présente en gris la zone de recherche associée au mur 0 (mur présentant le meilleur gain informationnel).

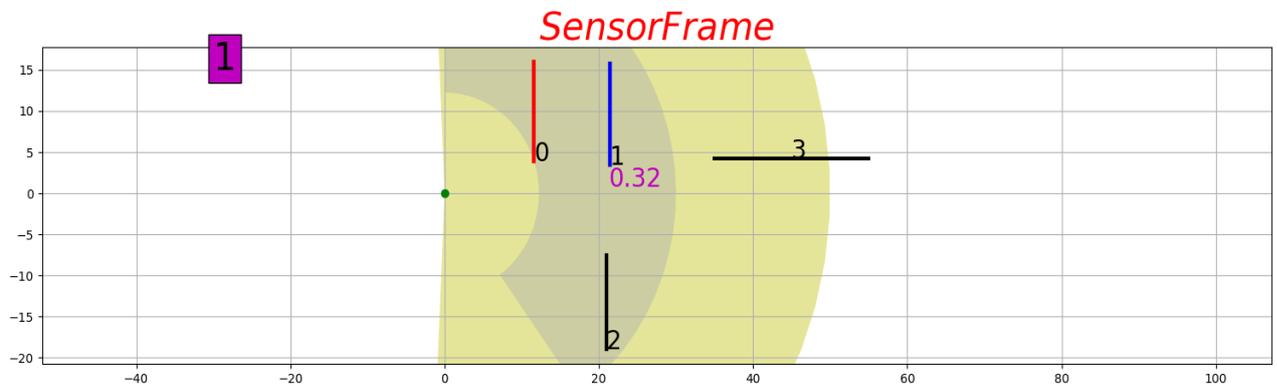


FIGURE VI.9 – Configuration initiale avec 4 murs dans l'espace capteur. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

À l'issue de la phase de détection, une ambiguïté se présente. En effet, le processus a prévu de détecter le mur 0 (en rouge sur la figure VI.9) mais le détecteur retourne le mur 1 (en bleu sur la figure VI.9) comme étant le mur 0. Dans la situation actuelle (grande incertitude de position), les caractéristiques du mur 1 sont parfaitement compatibles avec celles du mur 0. Par contre la probabilité liée à la confiance n'est que de 0.32 car il y avait 3 murs compatibles avec le mur 0 dans cette zone de recherche.

L'algorithme considère tout de même que cette détection est correcte (aucune autre information à ce niveau pour infirmer cette hypothèse) et le processus va être réitéré pour une profondeur de l'arbre de recherche égale à 1. Sur la figure VI.10, il est facile d'observer la nouvelle position estimée du robot (éloignée de la vérité en raison de l'ambiguïté) et son ellipse d'incertitude. Cette ellipse est tout de même cohérente par rapport à la détection précédente car le mur 0 (qui était en fait le mur 1) a donné des informations de position suivant l'axe horizontal mais très peu suivant l'axe vertical. Par contre l'erreur de positionnement en horizontal est liée à la confusion entre les murs 0 et 1. Pour atteindre l'objectif fixé, l'algorithme cherche à réduire l'incertitude verticale tout en augmentant la probabilité liée à la confiance qui reste encore faible pour le moment. Parmi les amers encore observables (murs 1 et 2), la table de droite de la figure VI.10 indique que l'amer 1 est le plus pertinent pour atteindre l'objectif sachant que le mur 3 n'est pas observable dans cette configuration compte tenu de la portée du Lidar. D'après le calcul du critère établi, le mur 1 est l'amer qui apporte le plus d'information. En effet ce mur va sans doute permettre de lever l'ambiguïté potentielle rencontrée à la première itération.

La zone d'incertitude sur la position du véhicule étant encore très grande la zone de recherche pour le mur 1 est également très grande (voir figure VI.11). Le processus permet bien de détecter le mur 1. Il est important de noter que nous n'avons pas géré le fait que le processus ne peut pas détecter deux fois le même amer réel (ce qui est le cas ici). Cependant la confiance liée à cette nouvelle estimation reste faible 0.36 ce qui n'est pas suffisant pour atteindre l'objectif en terme de confiance et c'est la même chose pour le critère de précision.

Après cette nouvelle bonne détection, le processus est renouvelé (profondeur 2 de l'arbre de recherche). Le résultat de la mise à jour de l'état est présenté sur la figure VI.12. L'ellipse d'incertitude autour de la position estimée reste très grande suivant l'axe vertical. Il ne reste plus qu'un amer observable (mur 2) dans la situation

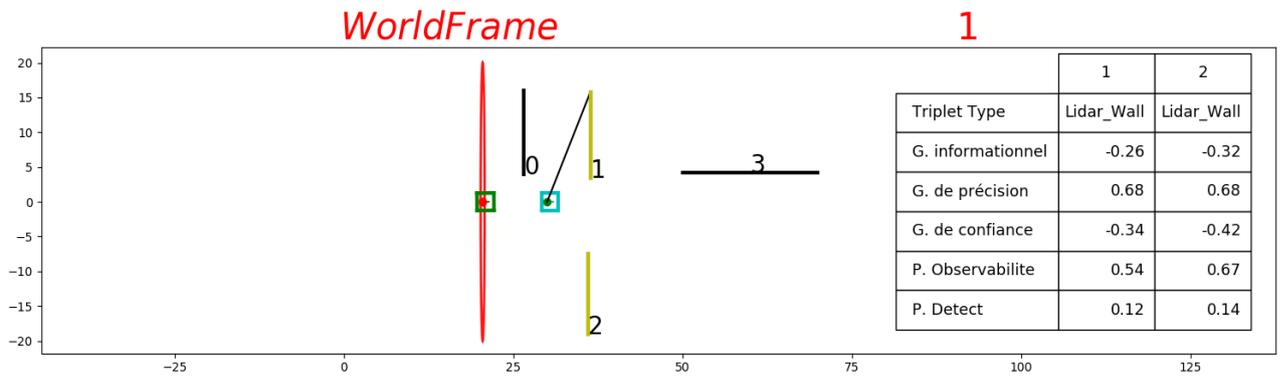


FIGURE VI.10 – Configuration avec 4 murs à la profondeur 1 avec ambiguïté. Sur la gauche, en cyan, la position réelle du robot, en vert la position estimée du robot et en rouge l’ellipse d’incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici 2 murs) dans la situation donnée.

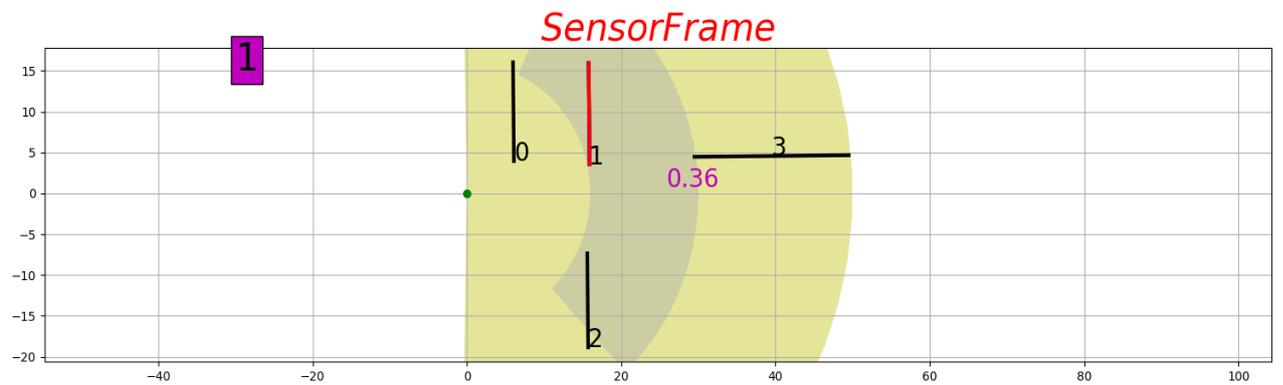


FIGURE VI.11 – Situation dans l’espace capteur après 1 détection. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

courante et il va donc être choisi pour essayer d’atteindre l’objectif fixé.

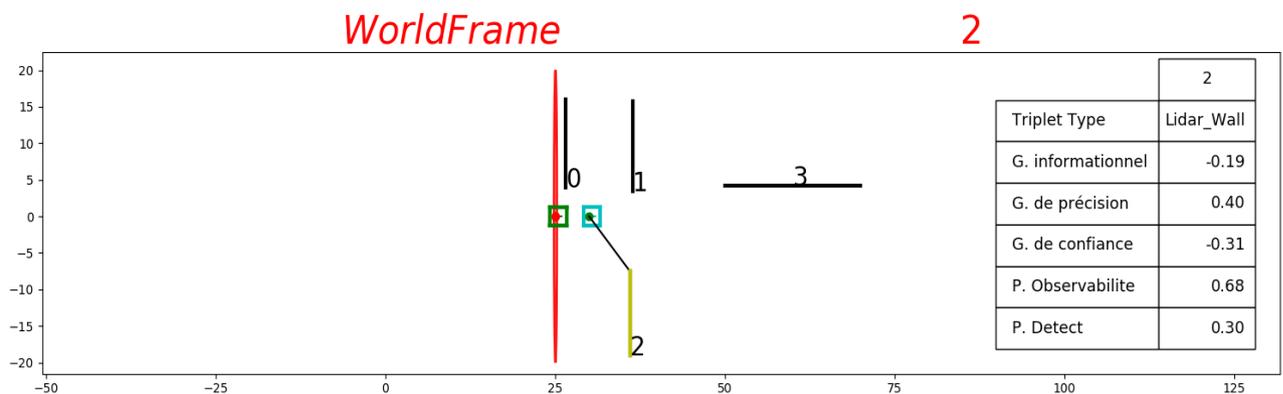


FIGURE VI.12 – Configuration avec 4 murs à la profondeur 2 avec ambiguïté. Sur la gauche, en cyan, la position réelle du robot, en vert la position estimée du robot et en rouge l’ellipse d’incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici 1 mur) dans la situation donnée.

La figure VI.13 représente la même situation dans l'espace capteur. La fenêtre de recherche autour du mur 2 (amer sélectionné) est représentée en gris et le résultat de la détection en rouge. Il est important de noter que la fenêtre de recherche est toujours très grande car l'incertitude sur la pose est encore très grande.

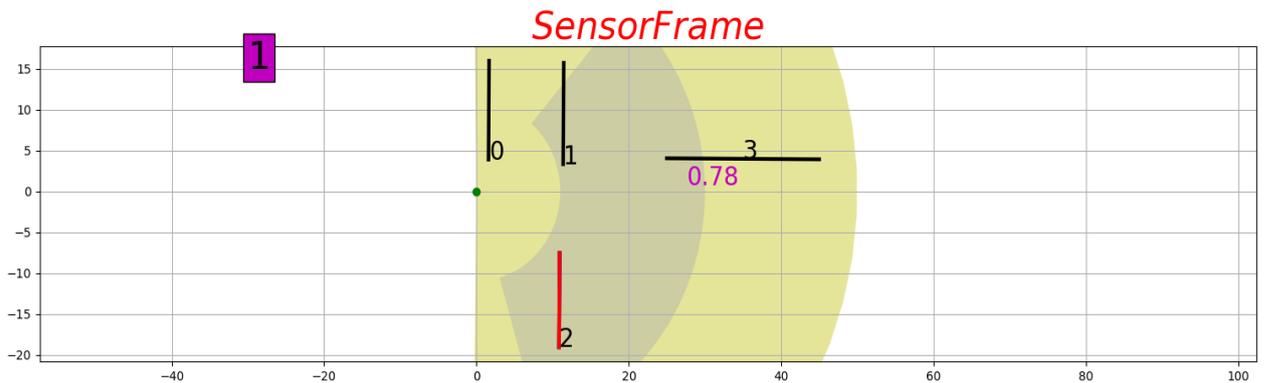


FIGURE VI.13 – Situation dans l'espace capteur après 2 détections avec ambiguïté. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

À la suite de cette nouvelle détection, l'algorithme a permis de détecter 3 amers dans l'environnement mais les objectifs en termes de confiance et de précision ne sont pas atteints. L'algorithme va donc remettre en cause de façon récursive chacune des détections précédentes jusqu'à revenir à la situation initiale. Dans cette configuration (voir figure VI.14), l'algorithme sélectionne le deuxième triplet le plus pertinent (le mur 2 ici) car le premier triplet (mur 0) n'a pas conduit à une situation satisfaisante.

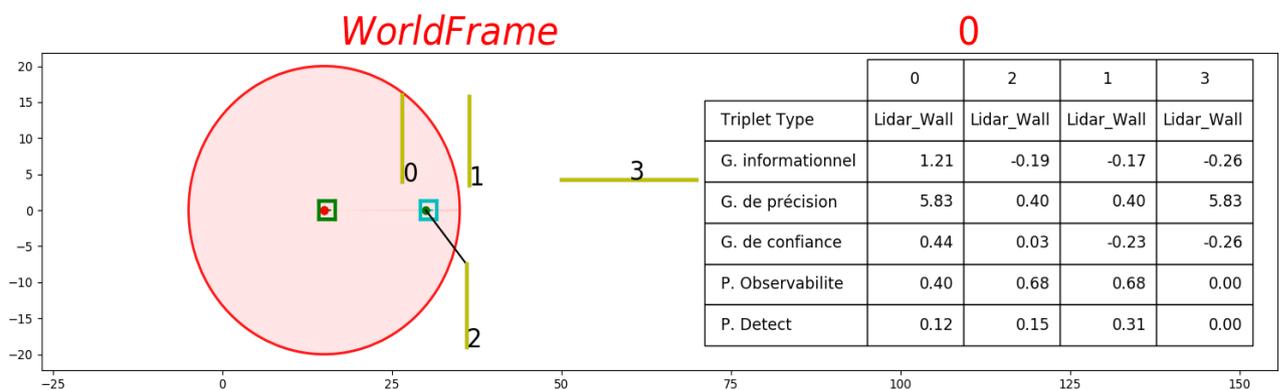


FIGURE VI.14 – Configuration avec 4 murs à la profondeur 0 après remise en cause. Sur la gauche, en cyan, la position réelle du robot, en vert la position estimée du robot et en rouge l'ellipse d'incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici 4 murs) dans la situation donnée.

La figure VI.15 présente en gris la zone de recherche associée au mur 2 et le résultat de la détection en rouge. Dans cette situation l'association est bonne entre le mur prévu et le mur détecté. Cependant la probabilité de confiance liée à cette nouvelle configuration n'est pas très élevée (0.25)

À l'issue de cette détection, la mise à jour de la pose du robot est réalisée et nous pouvons observer sur la figure VI.16 que la position estimée est parfaitement sur la pose réelle du robot. L'ellipse d'incertitude est encore grande selon l'axe vertical mais le prochain triplet sélectionné doit permettre de réduire cette incertitude et également confirmer la vraisemblance de l'estimation. Le mur 0 n'est plus observable dans cette configuration

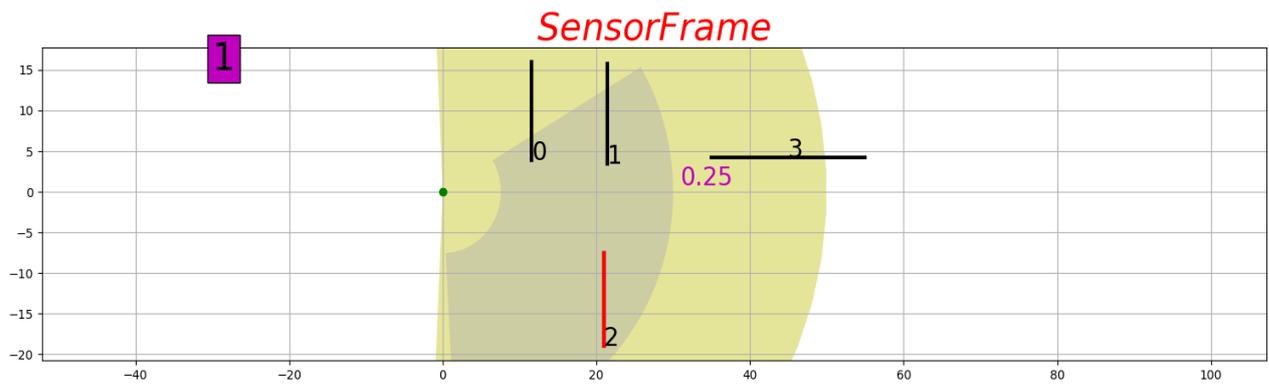


FIGURE VI.15 – Nouvelle configuration avec 4 murs dans l’espace capteur après remise en cause. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

car il se trouve derrière le robot et que le capteur utilisé ne peut voir que vers l’avant. Le mur le plus pertinent pour atteindre les objectifs est le mur 3 car il apporte de l’information à la fois pour la partie précision mais également sur la confiance.

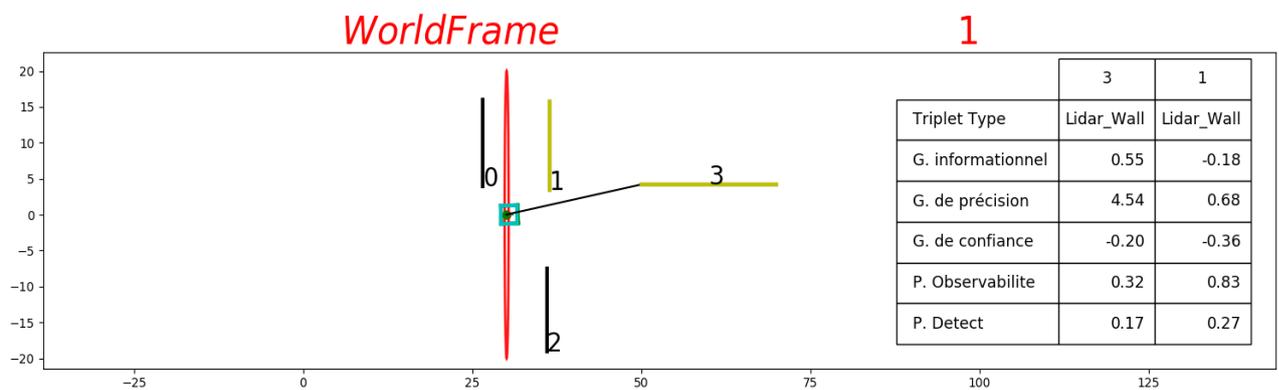


FIGURE VI.16 – Configuration avec 4 murs à la profondeur 1 après remise en cause. Sur la gauche, en cyan, la position réelle du robot, en vert la position estimée du robot et en rouge l’ellipse d’incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici 2 murs) dans la situation donnée.

La zone de recherche du mur 3 est présentée sur la figure VI.17 et le résultat de la détection sur la figure VI.18.

Le segment (mur 3) est parfaitement détecté et la probabilité liée à la confiance a augmenté. L’algorithme va évoluer afin de chercher une nouvelle détection (nouvel amer) permettant d’atteindre les objectifs. Il ne reste plus que le mur 1 qui est le seul observable dans cette configuration (voir figure VI.18).

La fenêtre de recherche est fortement réduite par rapport aux itérations précédentes et comme dans le cas avec les 3 murs, les processus de focalisation spatiales et géométriques permettent de minimiser les chances de ne pas trouver le bon mur (augmentation du rapport signal sur bruit) (voir figure VI.19).

La figure VI.20 montre le résultat final du processus de localisation dans cette situation avec 4 murs dans l’environnement présentant une ambiguïté entre les murs 0 et 1. La position initiale et son incertitude initiale sont représentées respectivement en rouge et en bleu. La vraie position et la position estimée finale sont repré-

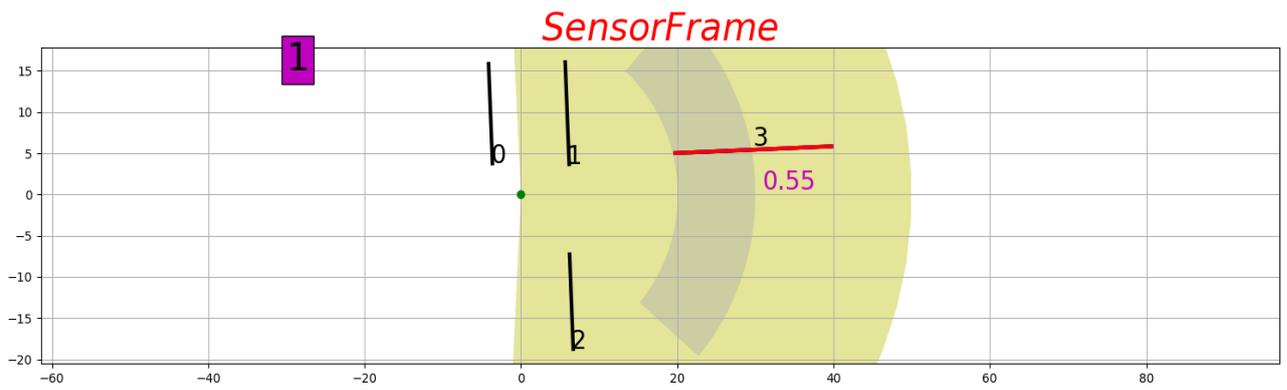


FIGURE VI.17 – Situation dans l'espace capteur après 1 détection après remise en cause. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

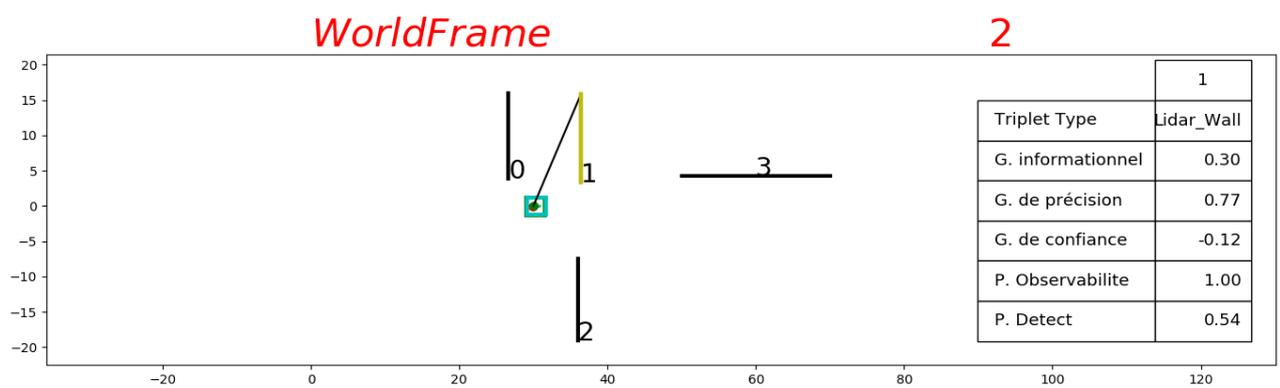


FIGURE VI.18 – Configuration avec 4 murs à la profondeur 2 après remise en cause. Sur la gauche, en cyan, la position réelle du robot, en vert la position estimée du robot et en rouge l'ellipse d'incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici 1 mur) dans la situation donnée.

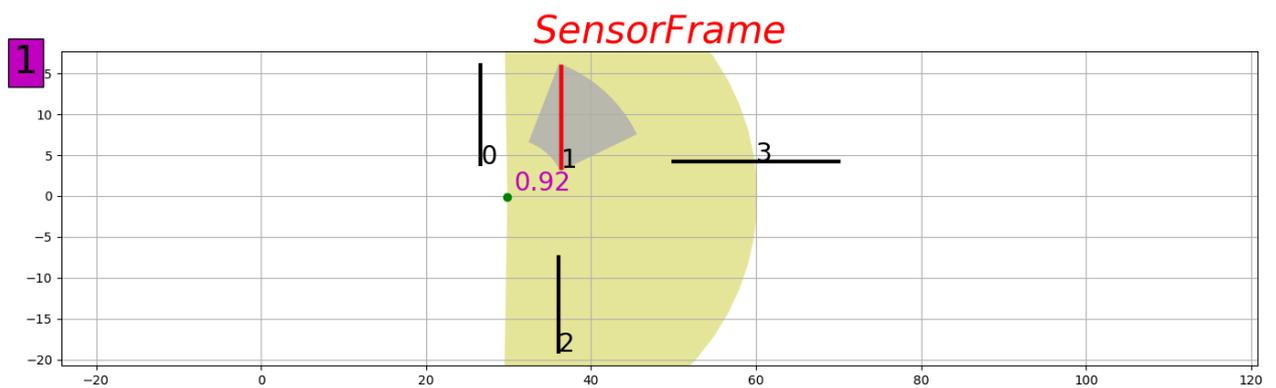


FIGURE VI.19 – Situation dans l'espace capteur après 2 détection après remise en cause. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

sentées en cyan et en noir. La figure de droite donne un zoom de la situation finale avec les incertitudes à 1 sigma (en rouge) et à 3 sigma (en noir) autour de la position finale estimée. L'estimation finale est parfaitement intègre.

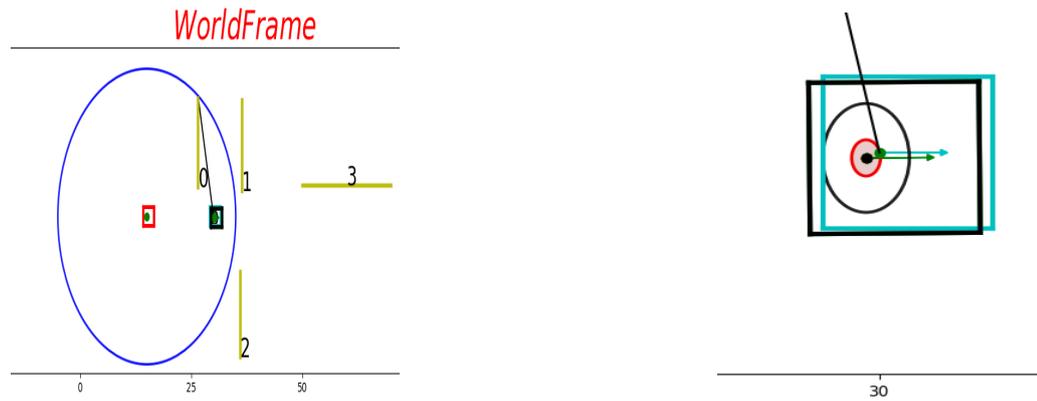


FIGURE VI.20 – Configuration finale avec 4 murs. En cyan, la position réelle du robot, en vert la position estimée du robot et en rouge la situation initiale du robot avec l'ellipse d'incertitude à un sigma autour de la position initiale estimée en bleu. La figure de droite donne la finale zoomée avec 4 murs. configuration

## VI.5 Résultats avec 15 murs

Par rapport aux simulations précédentes les seuls paramètres qui changent concernent la configuration initiale du robot. Les nouveaux paramètres sont :

- la position réelle du robot est située au vecteur de coordonnées suivantes :  $\underline{X} = (0, 0, 0)^\top$  ce qui signifie que le robot est positionné en  $x = 0$ ,  $y = 0$  et  $\theta = 0$
- la position estimée du robot est initialisée de façon aléatoire autour de la position réelle avec une incertitude de 20m suivant les axes  $x$  et  $y$  et les incertitudes sur la position initiale sont donc fixées à  $\sigma_x = \sigma_y = 20m$  et  $\sigma_\theta = 10^\circ$ .
- la probabilité initiale liée à la confiance est fixée à 0.9.

Dans une première partie, nous allons présenter les résultats d'une simulation dans un cas relativement simple avec 15 murs positionnés dans l'environnement. Dans une seconde partie une étude quantitative et qualitative sera présentée.

### VI.5.1 Cas "simple" avec 15 murs

Dans cette section nous présentons les résultats de l'approche proposée dans un cas où il y a la présence de 15 murs dans l'environnement. Suivant la position initiale donnée du robot, de nombreuses situations ambiguës peuvent être rencontrées. La configuration initiale pour cette simulation est présentée sur la figure VI.21.

Compte tenu de la situation courante, le mur 6 est sélectionné par notre processus. La différence en terme d'apport informationnel est très faible entre les différents premiers amers car la situation initiale est très incertaine. La seule différence se trouve au niveau de la probabilité d'observabilité des amers car dans certaines parties de la zone d'incertitude des amers ne peuvent être observés.

La figure VI.22 présente en gris la zone de recherche associée au mur sélectionné (mur 6) sachant que cet amer présente le meilleur gain informationnel.

Le mur détecté dans cette fenêtre, et surtout compatible avec le mur attendu, est représenté en rouge sur la figure VI.22. Il est également indiqué sur cette figure la probabilité liée à la confiance sur l'estimation de la nouvelle pose par rapport à la vraie position à l'issue de cette détection. Cette probabilité a augmenté par rapport à la probabilité initiale (0.9) car il n'y a pas d'autre amer dans la zone de recherche (aucune ambiguïté).

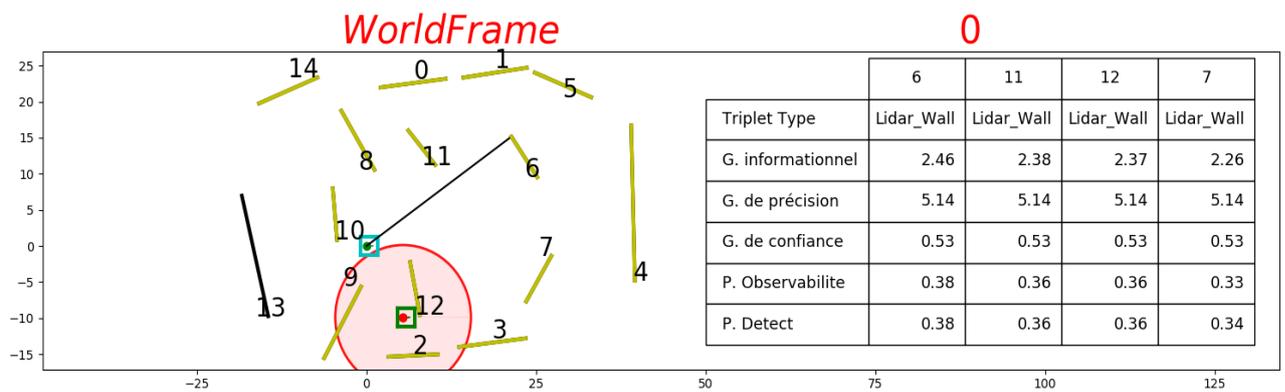


FIGURE VI.21 – Configuration initiale avec 15 murs. Sur la gauche, en cyan, la position réelle du robot, en vert la position initiale estimée du robot et en rouge l’ellipse d’incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici limitée à 4 murs) dans la situation donnée.

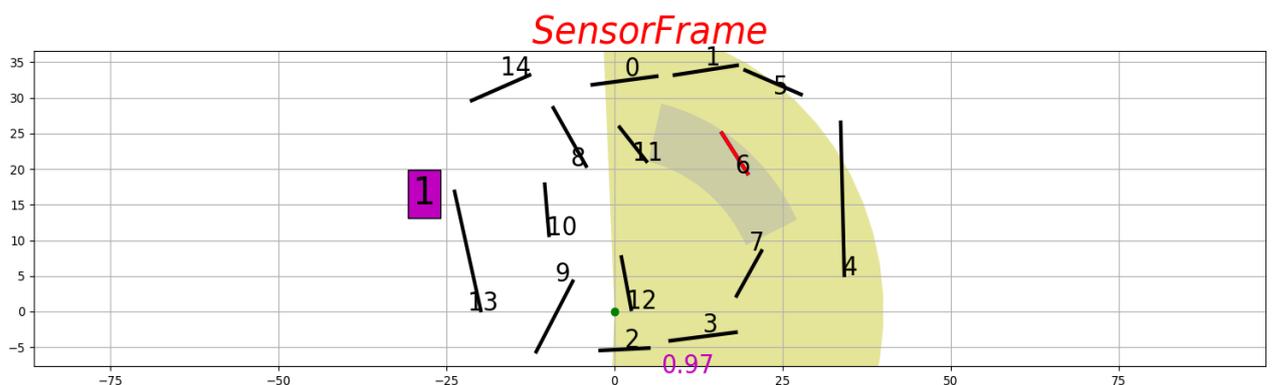


FIGURE VI.22 – Configuration initiale avec 15 murs dans l’espace capteur. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

Cependant l’algorithme ne peut pas garantir une bonne association ce qui est pourtant le cas ici.

À l’issue de cette détection, l’état estimé du robot est remis à jour et la nouvelle configuration est présentée sur la figure VI.23. La mise à jour est cohérente avec la détection du mur 6 réalisée. L’algorithme va sélectionner un nouvel amer pour atteindre les objectifs fixés. Dans cette situation données, le mur 2 est choisi car il apporte le meilleur gain informationnel en particulier au niveau de la précision.

Compte tenu de l’incertitude courante la zone de recherche du mur 2 dans l’espace capteur reste très grande mais grâce aux contraintes paramétriques, le seul mur viable dans cette zone reste le mur 2. La détection est donc réussie (voir figure VI.24). La probabilité de confiance dans cette estimation est maximale. Cependant l’objectif de précision n’est pas complètement atteint l’algorithme renouvelle donc son processus de sélection pour trouver un nouvel amer compatible avec la solution actuelle.

Dans cette situation, il reste 6 amers observables par le capteur et c’est le mur 7 qui est sélectionné par l’algorithme comme étant celui apportant le plus d’informations pour atteindre l’objectif (voir figure VI.25).

Comme pour les simulations avec 3 et 4 murs, à cette profondeur (2), la zone de recherche devient très petite et le processus de détection est très efficace pour retrouver le bon amer (voir figure VI.26).

Après la mise à jour de l’état du robot, les objectifs de précision et de confiance sont atteints. L’algorithme

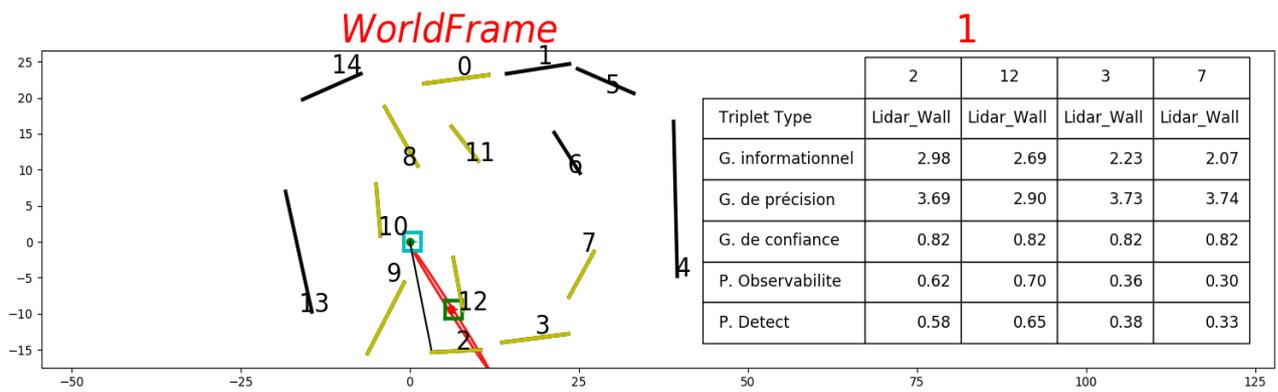


FIGURE VI.23 – Configuration avec 15 murs à la profondeur 1. Sur la gauche, en cyan, la position réelle du robot, en vert la position initiale estimée du robot et en rouge l’ellipse d’incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici limitée à 4 murs) dans la situation donnée.

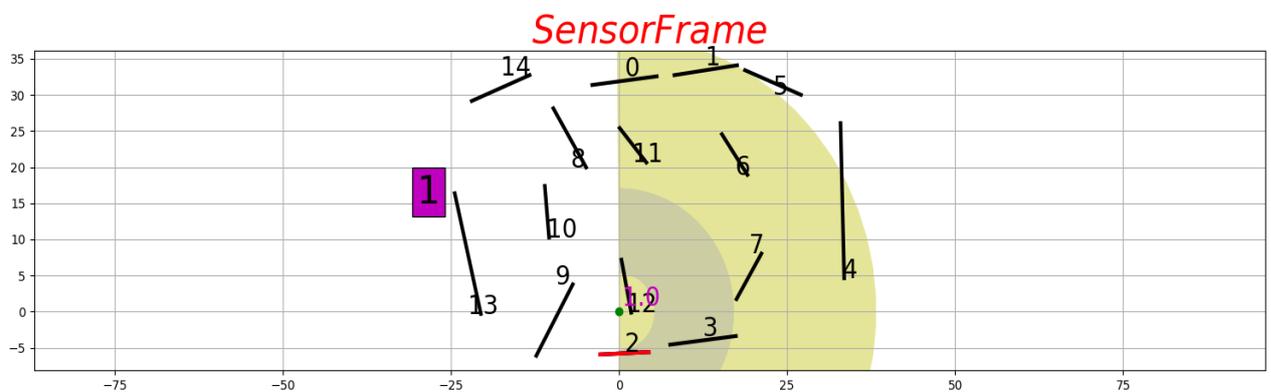


FIGURE VI.24 – Configuration avec 15 murs dans l’espace capteur à la profondeur 1. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

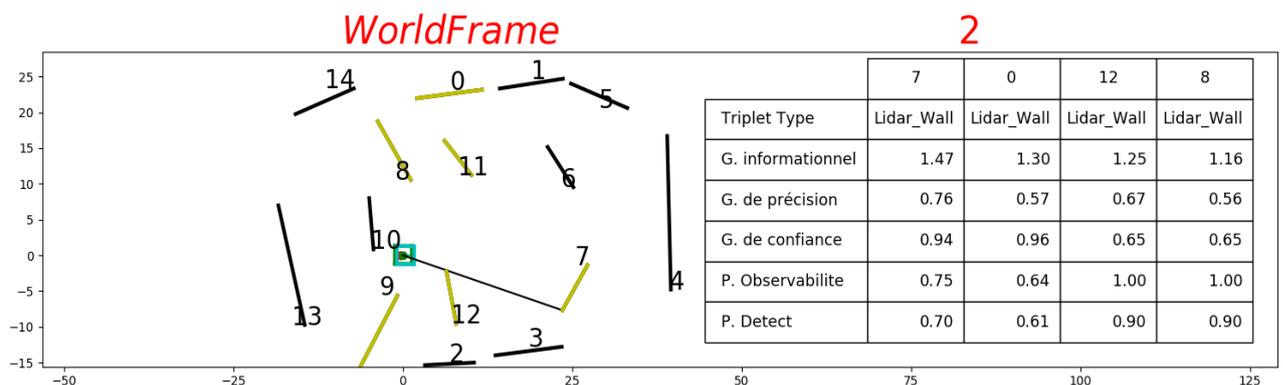


FIGURE VI.25 – Configuration avec 15 murs à la profondeur 2. Sur la gauche, en cyan, la position réelle du robot, en vert la position initiale estimée du robot et en rouge l’ellipse d’incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici limitée à 4 murs) dans la situation donnée.

renouvelle ses processus de sélection et de détection afin d’obtenir la solution optimale dans la configuration donnée. Ces étapes sont exécutées très rapidement car les zones de recherche sont très petites. La figure VI.27

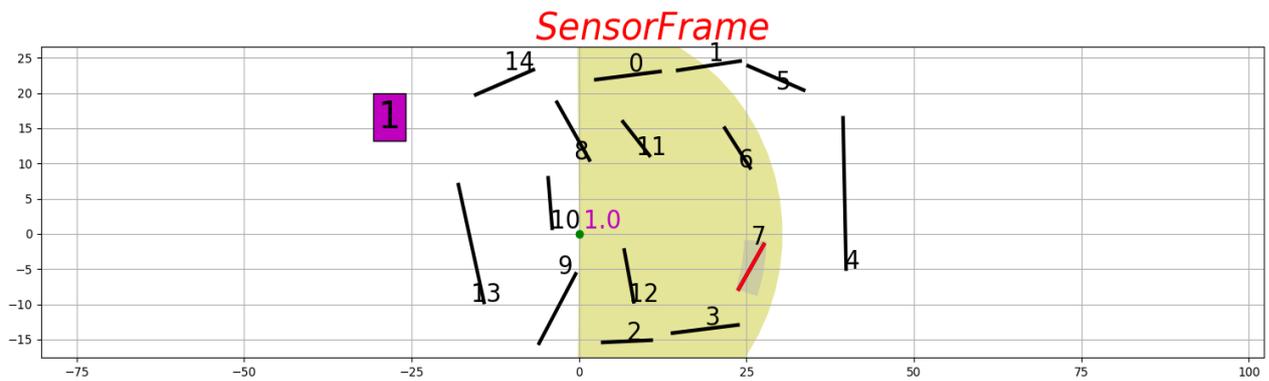


FIGURE VI.26 – Configuration avec 15 murs dans l’espace capteur à la profondeur 2. La fenêtre de recherche est dessinée en gris et le segment détecté en rouge.

présente l’état de la situation à la profondeur 4 de l’arbre de recherche.

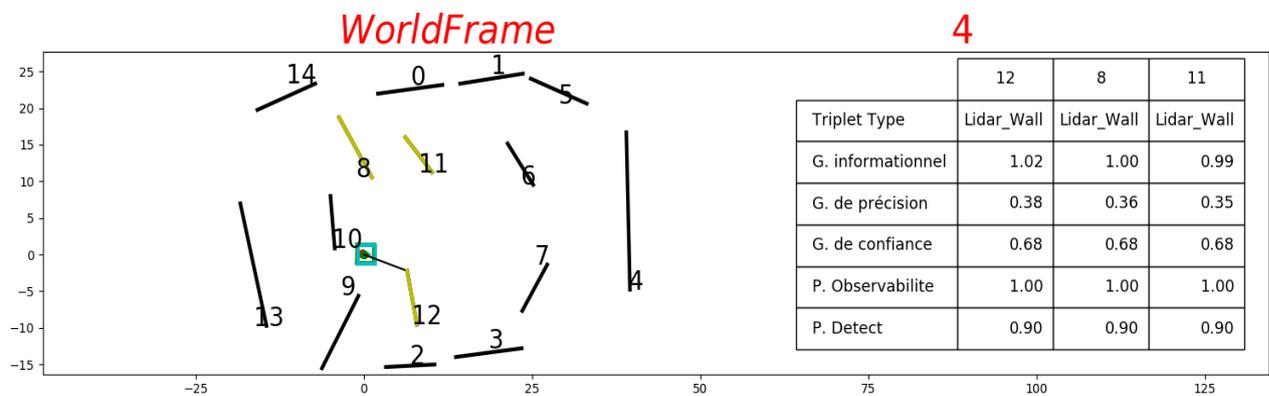


FIGURE VI.27 – Configuration avec 15 murs à la profondeur 4. Sur la gauche, en cyan, la position réelle du robot, en vert la position initiale estimée du robot et en rouge l’ellipse d’incertitude à un sigma autour de la position estimée. Sur la droite, la table des probabilités liées à chacun des amers (ici 3 murs) dans la situation donnée.

La résultat final de l’algorithme est présenté sur les figures VI.28. Le processus s’est arrêté à la profondeur 7 ce qui signifie que 7 détections de différents murs sont compatibles entre elles pour atteindre la solution finale. La position initiale et son incertitude initiale sont représentées respectivement en rouge et en bleu. La vraie position et la position estimée finale sont représentées en cyan et en noir. La figure de droite donne un zoom de la situation finale avec les incertitudes à 1 sigma (en rouge) et à 3 sigma (en noir) autour de la position finale estimée. L’estimation finale est parfaitement intègre alors que la position initiale était très éloignée de la position réelle.

## VI.5.2 Tests quantitatifs et qualitatifs

Afin de mieux caractériser l’approche proposée dans cet environnement de 15 murs, l’algorithme a été exécuté 1000 fois dans ce même environnement avec une position initiale aléatoire centrée sur la position réelle du robot mais avec des variations possibles de 20m selon les axes  $x$  et  $y$  et  $10^\circ$  selon  $\theta$  (loi de tirage uniforme).

Comme le montre la figure VI.29, l’algorithme a atteint l’objectif dans 996 cas sur 1000 c’est à dire que dans tous ces cas l’algorithme atteint une probabilité de confiance minimale de 0.85 et une incertitude de posi-

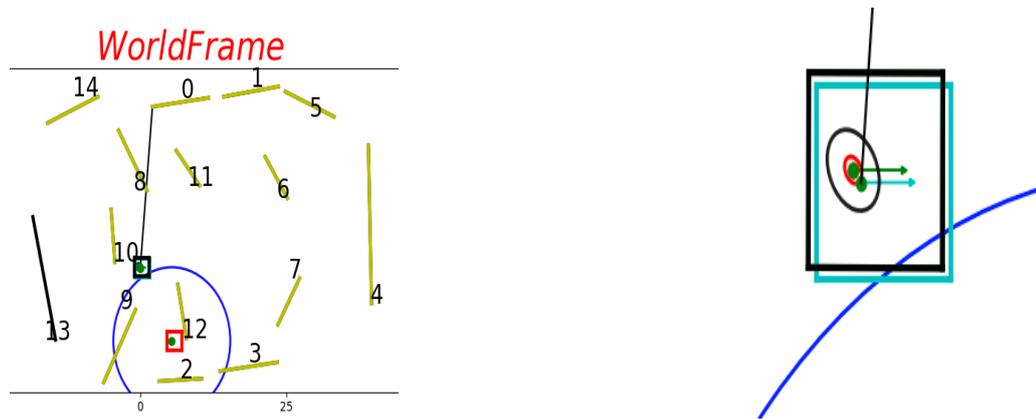


FIGURE VI.28 – Configuration finale avec 15 murs. En cyan, la position réelle du robot, en vert la position estimée du robot et en rouge la situation initiale du robot avec l'ellipse d'incertitude à un sigma autour de la position initiale estimée en bleu. La figure de droite donne la configuration finale zoomée.

tion de  $10\text{cm}^2$ .

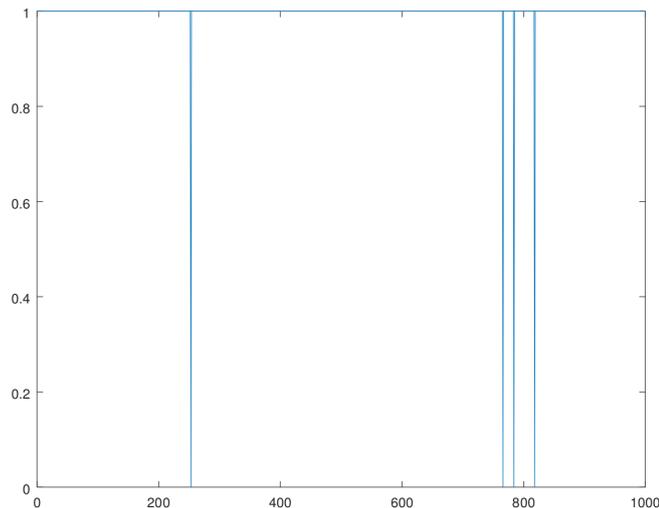


FIGURE VI.29 – Résultat de la reconnaissance sur 1000 situations. Si les objectifs ont été atteints, reconnaissance vaut 1 sinon 0

La figure VI.30 présente le nombre d'itérations réalisées par l'algorithme pour atteindre la meilleure solution. Ce nombre varie en fonction des situations rencontrées. En effet, avec cet environnement de 15 murs de nombreuses situations ambiguës peuvent être rencontrées ce qui entraîne l'exécution du processus de remise en cause. Cependant dans 85% des cas, le nombre d'itérations est inférieur à 20.

La figure VI.31 donne la probabilité de confiance en l'estimation réalisée sur les 996 cas où l'algorithme considère avoir atteint la solution optimale. Cette confiance est en permanence très proche de 1.

La figure VI.32 présente la valeur de la distance de Mahalanobis entre la position estimée et son incertitude associée par l'algorithme et la position réelle du robot. La solution proposée peut être considérée comme intégrale si la distance de Mahalanobis est inférieure, dans 95% des cas, à la limite de la loi du  $\chi^2$  pour 3 degrés de liberté qui est fixée à  $\sqrt{\chi_3^2(0.95)} = 2.79$ . Les distances présentées sur la figure VI.32 sont toujours inférieures à cette valeur ce qui prouve bien l'intégrité de l'estimation réalisée.

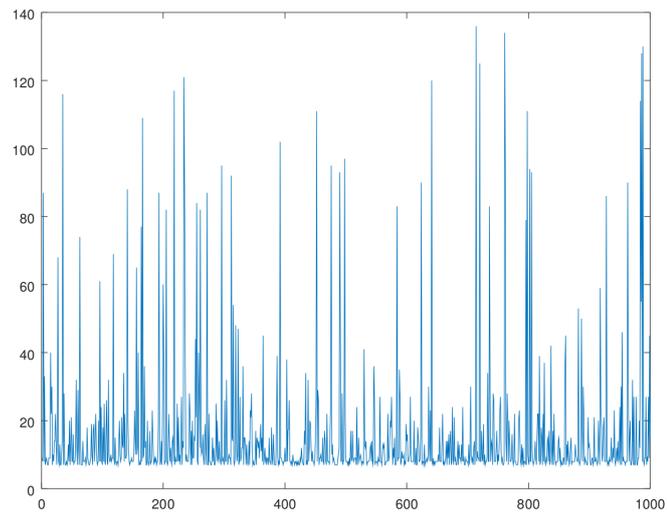


FIGURE VI.30 – Nombre d'itérations de l'algorithme sur 1000 situations.

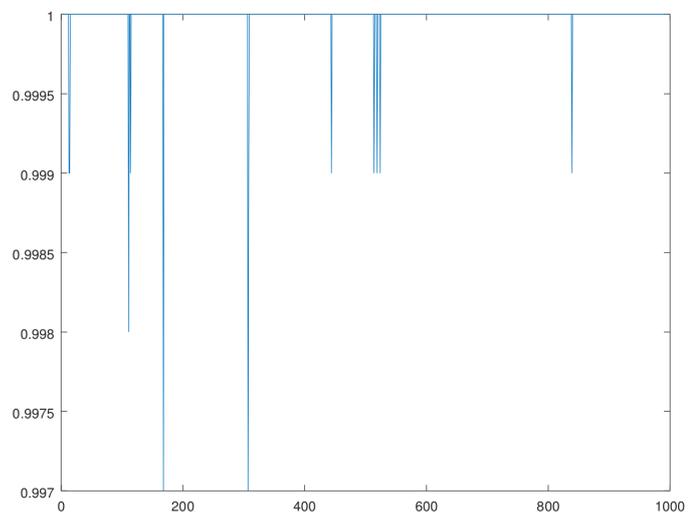


FIGURE VI.31 – Probabilité de confiance sur 996 situations.

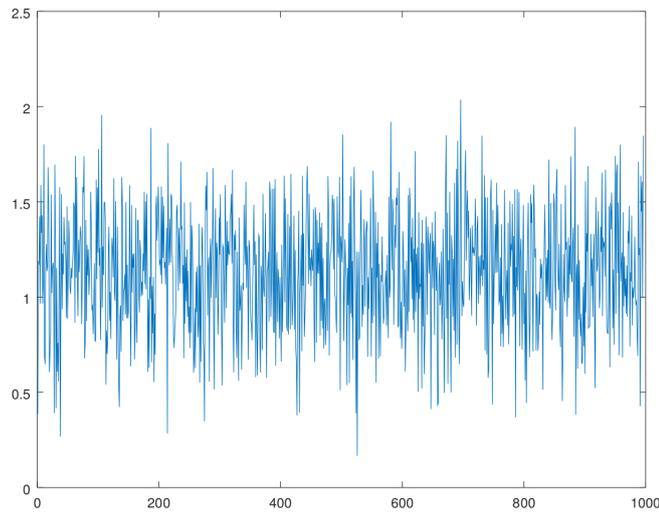


FIGURE VI.32 – Distance de Mahalanobis sur 996 situations.

## VI.6 Résultats sur le site PAVIN

Par rapport aux simulations précédentes les seuls paramètres qui changent concernent la configuration initiale du robot. Les nouveaux paramètres sont :

- la position réelle du robot est initialisée de façon aléatoire autour de la position :  $\underline{X} = (-10, 0, 0)^\top$  avec une variation uniforme possible de 15m selon l'axe  $x$ , de 20m selon l'axe  $y$  ce qui signifie que le robot peut être positionné aléatoirement dans un carré de 30m de large et de 40m de haut autour de la position  $\underline{X} = (-10, 0, 0)^\top$ . Ces valeurs ont été choisies en fonction de la configuration des murs du site PAVIN avec 56 murs (voir figure VI.33).
- la position estimée du robot est initialisée de façon aléatoire autour de la position réelle fixée précédemment avec une incertitude de 15m suivant les axes  $x$  et  $y$
- les incertitudes sur la position initiale sont donc fixées à  $\sigma_x = \sigma_y = 20\text{m}$  et  $\sigma_\theta = 10^\circ$ .
- la probabilité initiale liée à la confiance est fixée à 0.9.

Dans un premier temps nous allons présenter des résultats bruts de l'approche proposée dans ce manuscrit sur deux cas distincts et ensuite nous proposerons une étude un peu plus quantitative et qualitative sur plusieurs itérations de l'algorithme.

### VI.6.1 Résultats bruts sur PAVIN

Dans la première configuration, la situation initiale est présentée sur la figure VI.34. Les murs observables depuis un point quelconque situé dans l'ellipse d'incertitude en rouge sont représentés en jaune sur la figure VI.34 les autres sont en noir.

Le résultat final de l'algorithme est présenté sur la figure VI.35. Le résultat obtenu est précis et intègre (distance de Mahalanobis de 1.79) alors que la situation rencontrée est relativement complexe (une quarantaine de murs potentiels dans l'environnement). L'algorithme a convergé au bout de 13 itérations avec une profondeur de recherche de 12 ce qui correspond à 12 détections cohérentes avec la solution trouvée.

Pour la seconde configuration, la situation initiale est présentée sur la figure VI.36. Les murs observables depuis un point quelconque situé dans l'ellipse d'incertitude en rouge sont toujours représentés en jaune sur la figure VI.36 les autres sont en noir.

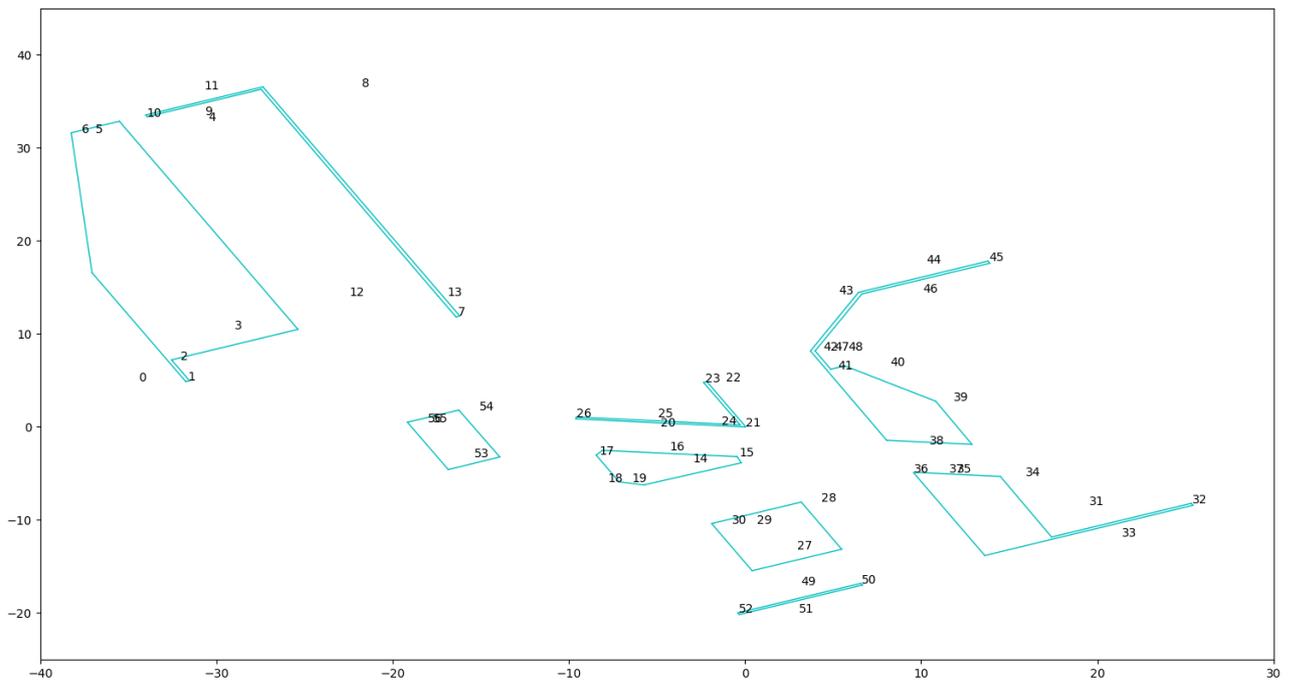


FIGURE VI.33 – Cartographie de PAVIN avec 56 murs.

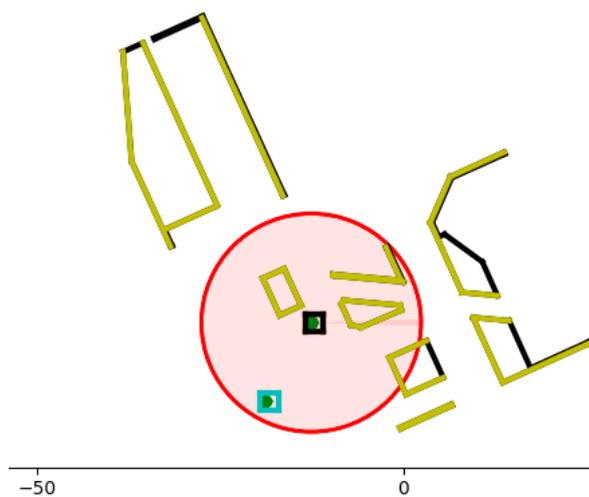


FIGURE VI.34 – Configuration initiale avec les murs de PAVIN (Situation 1). En cyan, la position réelle du robot, en noir la position initiale estimée du robot et en rouge l'ellipse d'incertitude à un sigma autour de la position estimée. Les murs observables depuis la position estimée sont représentés en jaune et les non observables sont en noir.

Le résultat final de l'algorithme est présenté sur la figure VI.37. Une nouvelle fois le résultat est correct. L'algorithme a convergé au bout de 206 itérations avec une profondeur de recherche de 5 ce qui correspond à 5 détections cohérentes avec la solution trouvée. L'algorithme a itéré beaucoup plus de fois car compte tenu de la position réelle du robot il y avait plusieurs situations ambiguës. Le processus de remise en cause a été souvent activé. Le résultat reste intègre (distance de Mahalanobis de 1.24) avec une probabilité de confiance de 0.91.

Afin de mieux caractériser notre approche nous allons réaliser plusieurs itérations de celle-ci sur le site PAVIN.

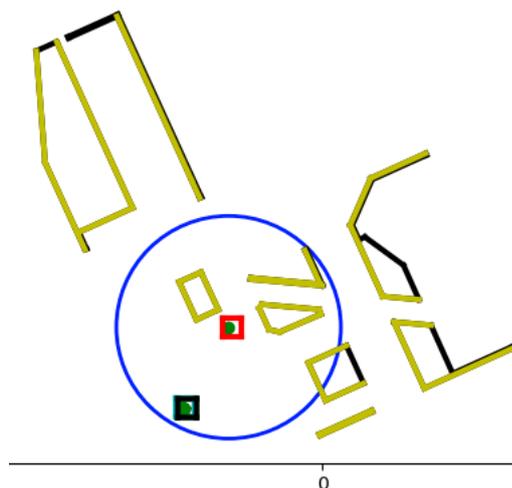


FIGURE VI.35 – Résultat final avec les murs de PAVIN (Situation 1). En cyan, la position réelle du robot, en vert la position estimée du robot et en rouge la situation initiale du robot avec l’ellipse d’incertitude à un sigma autour de la position initiale estimée en bleu.

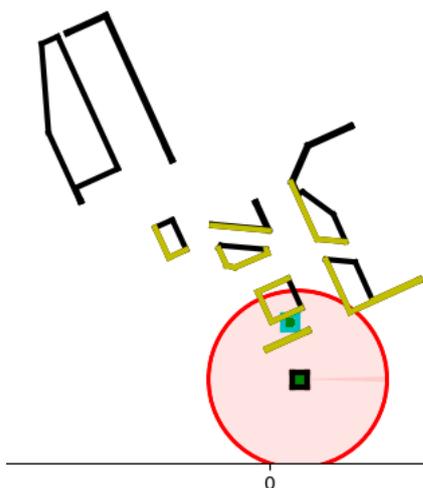


FIGURE VI.36 – Configuration initiale avec les murs de PAVIN (Situation 2). En cyan, la position réelle du robot, en noir la position initiale estimée du robot et en rouge l’ellipse d’incertitude à un sigma autour de la position estimée.

## VI.6.2 Tests quantitatifs et qualitatifs

Dans cette partie, nous allons présenter les résultats de notre approche qui a été exécutée 200 fois dans l’environnement PAVIN avec une position initiale aléatoire (uniforme) centrée sur la position réelle du robot mais avec des variations possibles de 15m selon les axes  $x$  et  $y$ .

Comme le montre la figure VI.38, l’algorithme a atteint l’objectif dans 192 cas sur 200 c’est à dire que dans tous ces cas l’algorithme atteint une probabilité de confiance minimale de 0.85 et une incertitude de position de  $10\text{cm}^2$ .

La figure VI.39 présente le nombre d’itérations réalisées par l’algorithme pour atteindre la meilleure solution. Ce nombre varie en fonction des situations rencontrées. En effet, avec cet environnement de 57 murs de nombreuses situations ambiguës peuvent être rencontrées ce qui entraîne l’exécution du processus de remise en cause. Cependant dans 80% des cas, le nombre d’itérations est inférieur à 20. Nous pouvons constater une

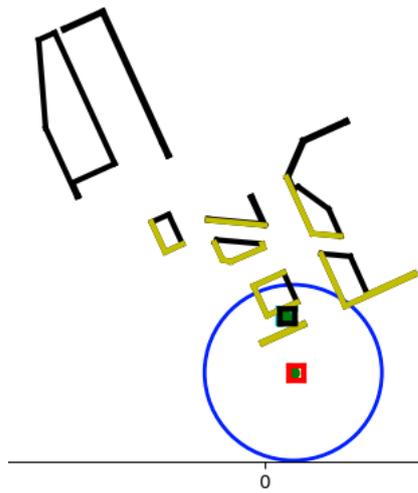


FIGURE VI.37 – Résultat final avec les murs de PAVIN (Situation 2). En cyan, la position réelle du robot, en vert la position estimée du robot et en rouge la situation initiale du robot avec l’ellipse d’incertitude à un sigma autour de la position initiale estimée en bleu.

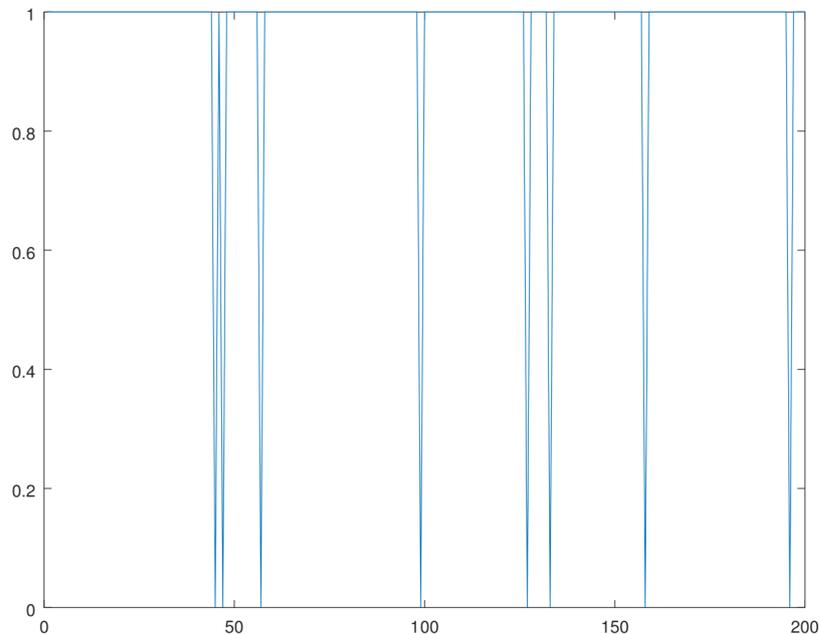


FIGURE VI.38 – Résultat de la reconnaissance sur 200 situations sur le site PAVIN. Si les objectifs ont été atteints, reconnaissance vaut 1 sinon 0

itération de l’algorithme avec environ 1600 itérations. Cette situation correspond à un cas très ambigu où quasiment tous les murs (50/57) étaient observables depuis l’ellipse d’incertitude initiale. L’algorithme a tout de même réussi à converger vers l’objectif dans cette situation.

La figure VI.40 donne la probabilité de confiance en l’estimation réalisée sur les 192 cas où l’algorithme considère avoir atteint la solution optimale. Cette confiance est en permanence très proche de 1 et elle ne descend que très rarement en dessous de 0.9.

La figure VI.41 présente la valeur de la distance de Mahalanobis entre la position estimée et son incertitude associée par l’algorithme et la position réelle du robot. Les distances présentées sur la figure VI.41 sont qua-

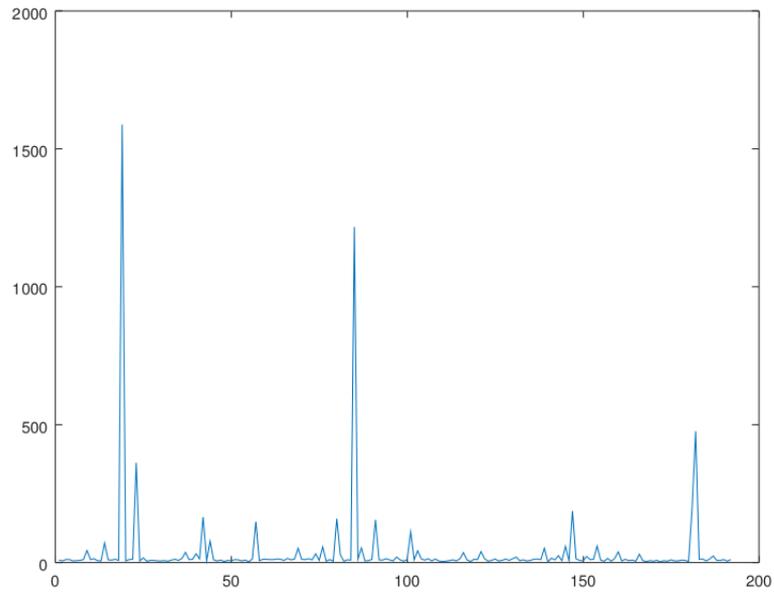


FIGURE VI.39 – Nombre d’itérations de l’algorithme sur 200 situations sur le site PAVIN.

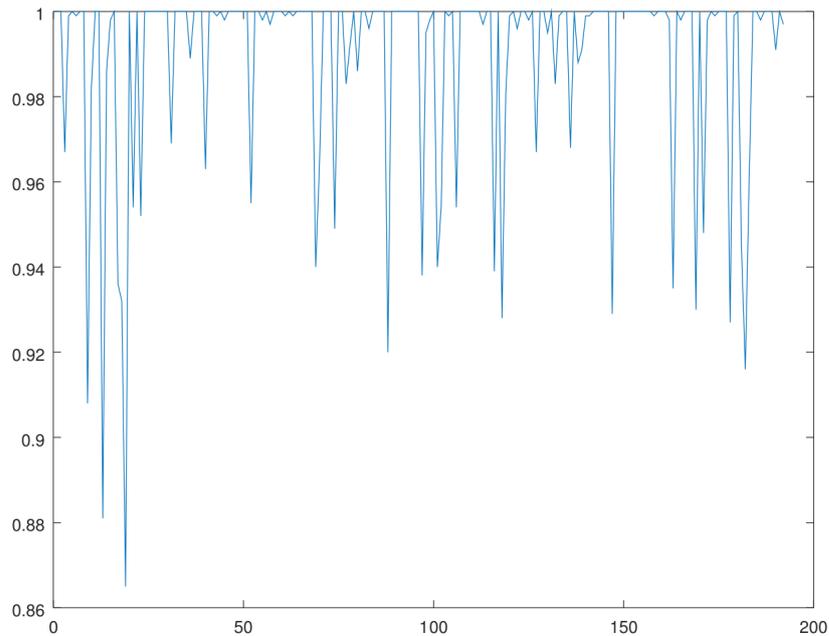


FIGURE VI.40 – Probabilité de confiance sur 192 situations sur le site PAVIN.

siment toujours inférieures à la valeur 2.79 (voir section VI.5.2) ce qui prouve bien l’intégrité de l’estimation réalisée. Dans quelques cas le résultat n’est pas intègre car la position estimée est légèrement éloignée de la position réelle et en plus l’incertitude associée à cette estimation est très faible ce qui fait augmenter la valeur de la distance de Mahalanobis.

Les seules situations où l’algorithme n’arrive pas à retrouver la position réelle du robot sont des situations bien particulières comme celle présentée sur la figure VI.42. Cela peut être expliqué par le fait que la position estimée est très éloignée de la position réelle du robot.

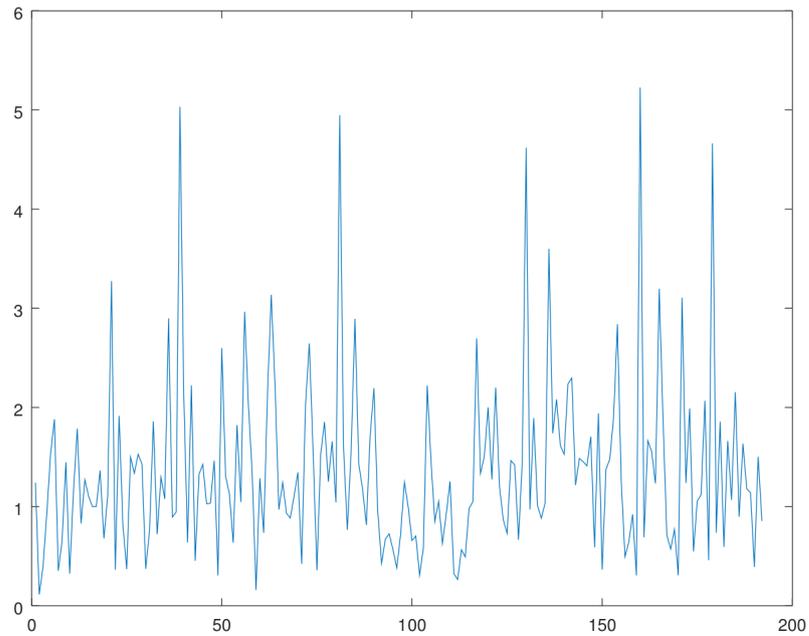


FIGURE VI.41 – Distance de Mahalanobis sur 192 situations sur le site PAVIN.

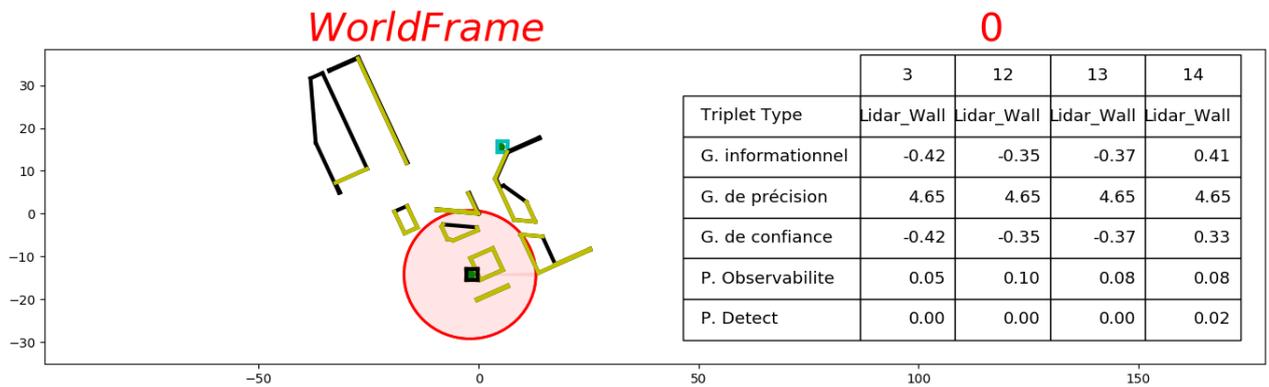


FIGURE VI.42 – Exemple de situation où l’algorithme n’atteint pas les objectifs sur le site PAVIN.

## VI.7 Prise en compte d’un récepteur GNSS

Nous allons nous intéresser à présent à l’apport d’un triplet GNSS (appelé ici pour simplifier *triplet GPS*) comme vu en section V.6.2.

Les scénarios présentés ont pour éléments communs :

- position de référence du robot en  $\underline{X} = (0, 0, 0)^T$ ,
- environnement composé de 15 murs,
- la mesure GPS est entachée d’un bruit de 3m d’écart-type.
- observabilité du signal GPS : 100%

Les scénarios suivants montrent la prise en compte par l’algorithme du triplet GPS.

### VI.7.1 Scénario 1 : grande incertitude initiale et GPS fiable

Nous considérons ici les points suivants :

- initialisation de la pose estimée du robot en  $\underline{X}_0 = (-25m, 15m, 20^\circ)^T$

- grande incertitude initiale sur cette pose d'écart-type 100m en  $x$  et  $y$  et  $30^\circ$  en angle d'orientation du robot,
- confiance initiale dans cette pose estimée du robot : 50%.
- confiance dans la mesure GPS : 95%.

Lorsque le triplet GPS n'est pas ajouté dans la liste des triplets, le système sélectionne naturellement le mur le plus prometteur (mur 9, comme le montre la figure VI.43). La reconnaissance de la pose s'opèrera ensuite en 16 itérations de détections dans ce cas. La confiance finale sera très proche de 1 (99.8%).

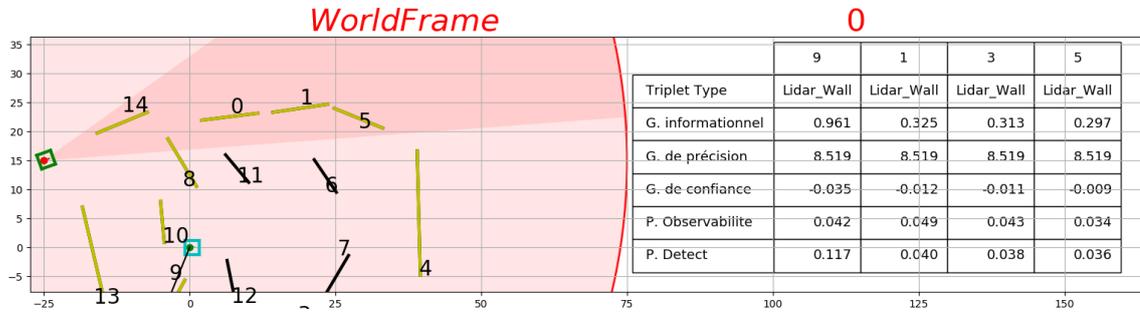


FIGURE VI.43 – Sur la gauche, en cyan, la position réelle du robot, en vert la position initiale estimée du robot et en rouge l'ellipse d'incertitude à un sigma autour de la position estimée. Le système choisit le triplet "mur 9" dès le début du processus. Seuls des triplets murs sont présents ici. La reconnaissance de la pose s'opèrera ensuite en 16 itérations de détections dans ce cas. La confiance finale sera de 0.9979

Lorsque le triplet GPS est ajouté, le processus le sélectionne d'emblée du fait qu'il représente un gain de confiance significatif (mais apporte moins de précision que la détection du mur 9) comme le montre la figure VI.44. Le système a sélectionné logiquement l'hypothèse 3 (fusion Kalman, voir section V.6.2.2).

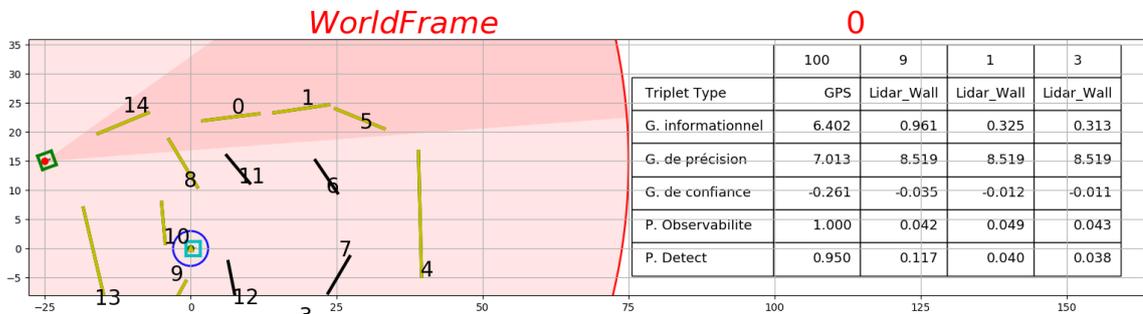


FIGURE VI.44 – Choix du triplet GPS de par son apport informationnel comparativement aux autres triplets. La zone d'incertitude autour de la mesure GPS est donnée par le cercle bleu. La reconnaissance de la pose s'opèrera ensuite en 7 itérations de détections ici. La confiance finale sera quasiment 100%.

## VI.7.2 Scénario 2 : grande incertitude initiale et GPS peu fiable

Nous gardons les conditions précédentes mais la confiance estimée sur la pose du robot passe à 95% et la confiance dans la pose fournie par le GPS passe à 50%. Nous observons les résultats de la figure VI.45. Le système préfère encore choisir le triplet GPS. En effet au vu de la taille de la zone de recherche (100m d'écart-type et  $30^\circ$  en angle) il est encore préférable de choisir le GPS. La convergence est assurée en 7 détections.

## VI.7.3 Scénario 3 : incertitude initiale moyenne et GPS peu fiable

Nous gardons les conditions précédentes (la confiance estimée sur la pose du robot vaut 95% et la confiance dans la pose fournie par le GPS vaut 50%) mais ici l'incertitude initiale est réduite à un écart-type de 20m en

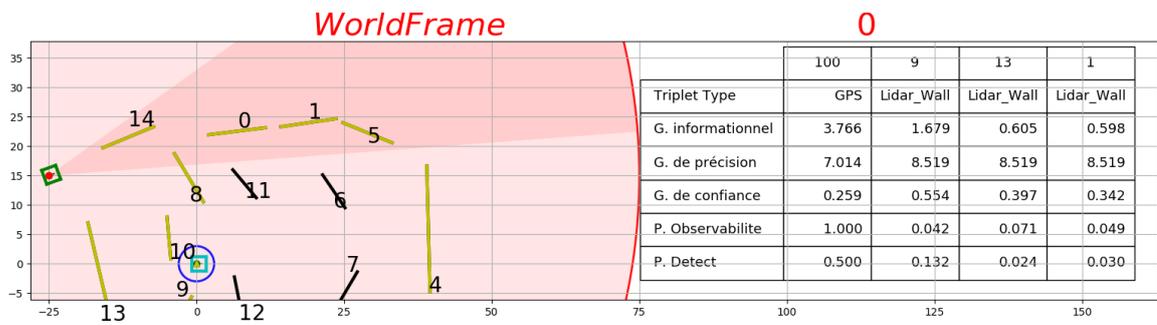


FIGURE VI.45 – Choix du triplet GPS (malgré sa confiance de seulement 50%) de par son apport informationnel comparativement aux autres triplets. La reconnaissance de la pose s’opèrera ensuite en 7 itérations de détections ici. La confiance finale sera de quasiment 100%

$x$  et  $y$  et de  $20^\circ$  en angle). Par ailleurs la pose initiale estimée est placée en  $\underline{X}_0 = (-15m, 15m, 20^\circ)^\top$  (pour garder l’intégrité).

Nous observons les résultats de la figure VI.46. Le système préfère dans ce cas choisir le triplet mur 9 car il est plus observable que précédemment. La convergence est assurée dans ce cas en 37 itérations.

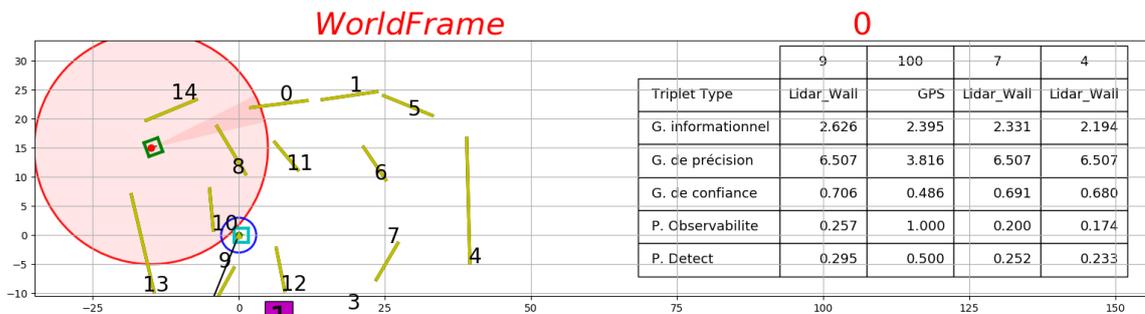


FIGURE VI.46 – Choix du triplet mur 9 dans cette configuration où l’incertitude initiale plus faible rend plus pertinente la sélection d’un triplet mur. La reconnaissance de la pose s’opèrera ensuite en 37 itérations de détections ici. La confiance finale sera de 100%

#### VI.7.4 Scénario 4 : incertitude initiale bonne mais peu fiable et GPS fiable

Dans ce cas, nous supposons que l’incertitude initiale est plutôt bonne (écart-type de 5m en  $x$  et  $y$  et de  $20^\circ$ ) mais centrée sur une pose fausse  $\underline{X}_0 = (-35m, 15m, 20^\circ)^\top$  : l’intégrité est donc perdue ici. Nous supposons que la confiance dans cette estimation est de 20% mais que la confiance dans la donnée GPS est de 90%.

Nous observons les résultats de la figure VI.47. Le système préfère dans ce cas choisir le triplet GPS. Le processus décrit au chapitre précédent (voir section V.6.2.2 ) privilégiera ensuite l’hypothèse 2 (mesure GPS seule, à la page 108). Ici, plutôt que de faire la fusion de Kalman avec la pose initiale, la mesure GPS sera choisie comme étant meilleure ce qui permet de rétablir une pose plus acceptable et d’assurer la convergence finale (dans ce cas en 17 détections).

#### VI.7.5 Bilan sur l’ajout du triplet GNSS

Dans cette partie, nous avons montré l’influence de la précision et de la confiance du capteur GNSS sur le critère de sélection ainsi que sur la mise à jour de l’état du robot. En particulier, étant donné la gestion de confiance, le système est en mesure de sélectionner l’hypothèse de fusion la plus adaptée plutôt que de réaliser une fusion de type Kalman systématiquement (qui aurait conduit à une estimation fautive car à mi-chemin entre l’estimée et la mesure dans le scénario 4)

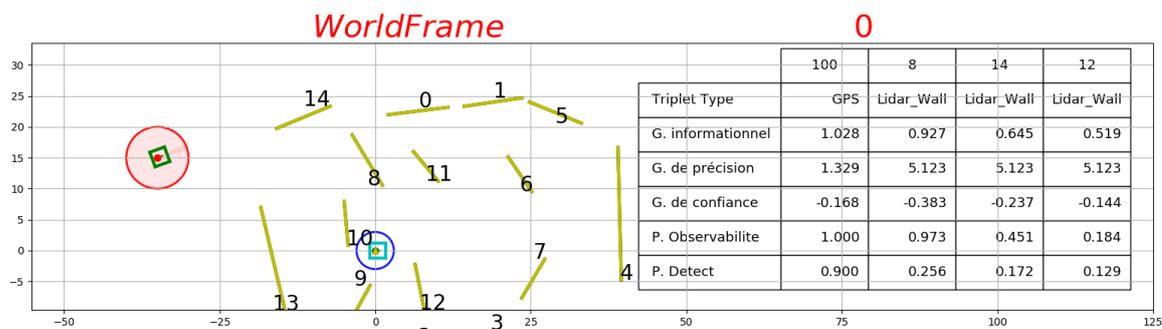


FIGURE VI.47 – Choix du triplet GPS de par son apport informationnel. La fusion ne sera pas faite avec l'état précédent du fait que l'hypothèse initiale est de faible probabilité. La reconnaissance de la pose s'opèrera ensuite en 17 itérations de détections ici. La confiance finale sera de 100%

## VI.8 Prise en compte de balises UWB

Dans cette section nous nous intéressons au cas de balises UWB. Comme nous l'avons vu en section §V.5.2.4, il s'agit d'un capteur *range-only* capable de donner uniquement l'information de distance mais pas l'angle d'orientation comme le ferait un capteur lidar par exemple sur un mur.

### VI.8.1 Situation canonique une balise et un mur avec mise à jour de type Kalman

Voyons tout d'abord une situation canonique simple avec une mise à jour de type Kalman (voir V.5.3). Dans cette situation, nous disposons d'un mur et d'une balise UWB numéro 200 (positionnée en (20, 10)). La précision de mesure dans le cas de ce triplet est supposée de 10cm. La pose initiale est (-10,0,0) et l'écart-type de son incertitude est (30m ,30m ,20°).

Le système sélectionne tout d'abord le mur du fait que la probabilité d'observabilité de la balise est moyenne mais le gain en confiance est faible comme le montre le tableau de la figure VI.48-a-. Le mur est donc choisi en premier ce qui fournit une incertitude très faible selon  $x$  mais qui reste grande selon  $y$ . La balise est ensuite choisie pour faire converger l'algorithme (figure VI.48-b-).

### VI.8.2 Situation avec une balise et 15 murs et mise à jour de type Kalman

La situation est ici un peu plus complexe :

- 15 murs sont présents dans l'environnement
- l'incertitude initiale est de 20m en  $x$  et  $y$  avec 20° en angle. La position initiale est (-10m, 10m, 10°).

Le mur 5 est choisi en premier (figure VI.49-a-). La balise elle, est en seconde position. Après la première mise à jour, l'ordre de pertinence a changé, la balise n'est plus dans les triplets les plus pertinents, elle ne sera choisie qu'en 9ème itération.

### VI.8.3 Illustration des effets de la linéarisation Kalman

Nous nous proposons ici de vérifier les effets de la linéarisation inhérente à la mise à jour de Kalman décrite en section V.5.3. Pour cela, nous disposons de seulement deux balises (200 et 201) comme l'indique la figure VI.50-a-. Après la première mise à jour, nous voyons (figure VI.50-b-) que l'ellipse de la mise à jour ne contient pas la vraie pose (en cyan dans la figure).

A présent, nous reprenons le même scénario mais avec la mise à jour proposée en section V.5.3 exploitant un filtre UKF. Après la première mise à jour, nous voyons (figure VI.51-b-) que l'ellipse de la mise à jour (à un écart-type) ne contient toujours pas la vraie pose (en cyan) mais reste proche (elle la contient à deux

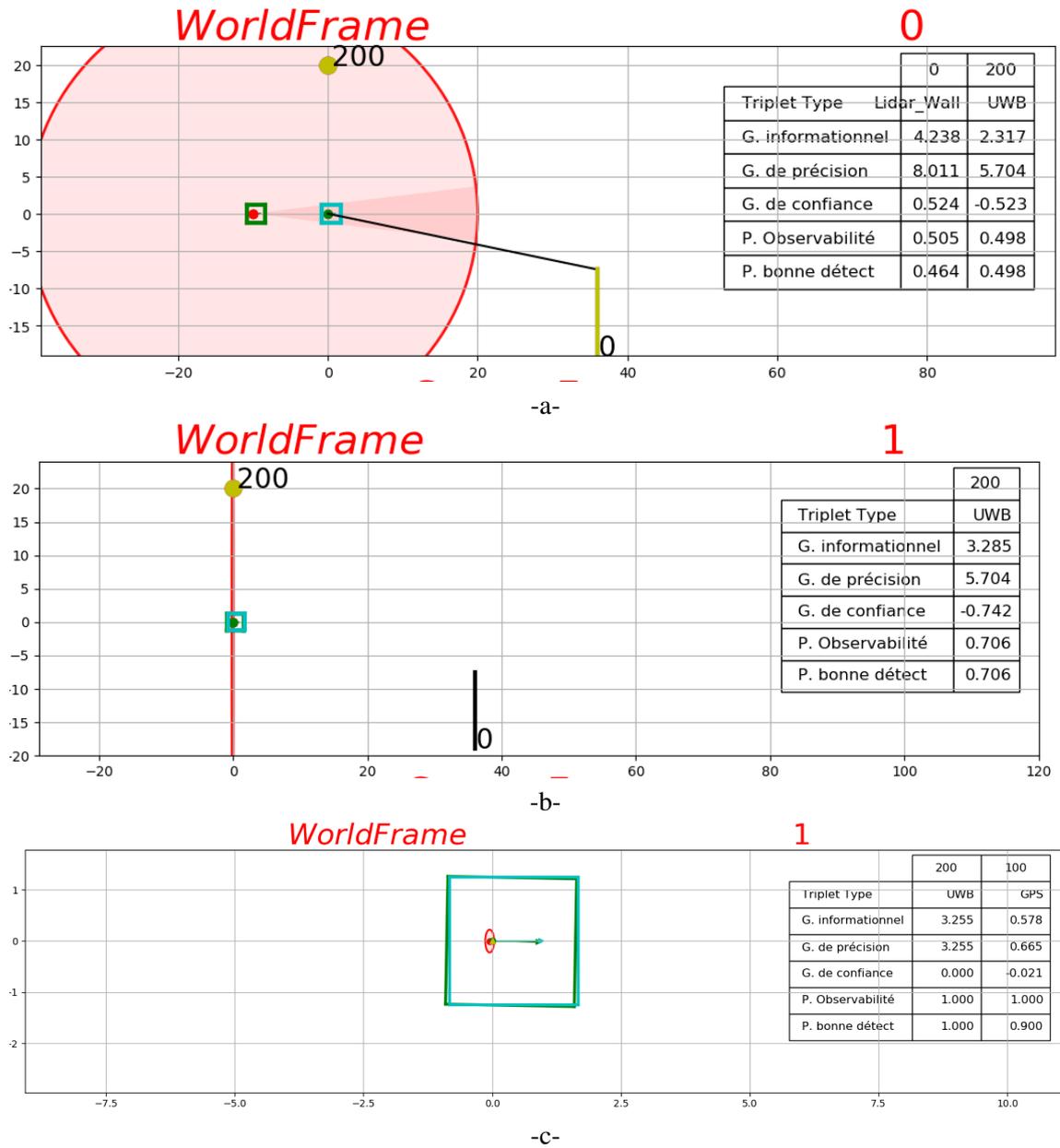


FIGURE VI.48 – Simulation avec un mur et une balise (numéro 200) positionnée en (20, 10). La précision de mesure est supposée de 10cm. La pose initiale (carré vert) est (-10,0,0) et l'écart-type de son incertitude est (30m, 30m, 20°) : ellipse rouge. La position réelle est en (0,0,0) : carré cyan. Le système sélectionne tout d'abord le mur (figure -a-) puis la balise (figure -b-). La figure -c- montre la convergence finale.

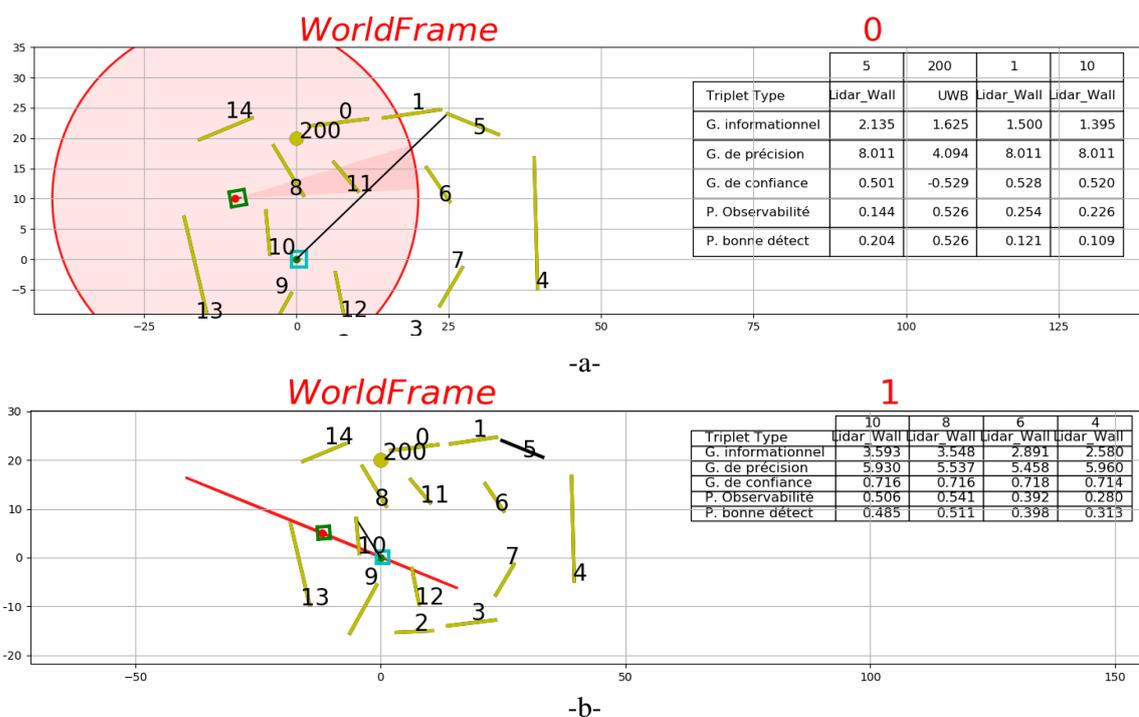


FIGURE VI.49 – Simulation avec 15 murs et une balise. Le système sélectionne tout d’abord le mur 5 (figure -a-), la balise arrive en seconde place (figure -a-). Après la mise à jour (figure -b-), l’ordre de pertinence des triplets a changé, ce sera encore un mur qui sera choisi. La balise ne sera choisie qu’à la 9<sup>ème</sup> itération.

écarts-types). La seconde mesure est faite par la balise 201 qui met à jour cet état de manière moins précise que précédemment mais qui est cette fois juste (figure VI.51-c-).

L’algorithme considère cependant que la reconnaissance de pose n’a pas abouti au vu du critère de précision non atteint. En effet, nous considérons que lorsqu’un triplet a été sollicité il ne peut pas l’être à nouveau. Ici, il aurait été intéressant de solliciter à nouveau la mesure de la balise 200. Cette possibilité a cependant été exclue car les filtres utilisés étant optimistes, plusieurs mises à jour consécutives pourraient conduire à une perte d’intégrité.

#### VI.8.4 Simulation UWB, lidar-mur, GPS avec pose initiale imprécise

Ici, figure VI.52, nous réalisons une simulation avec 2 balises UWB, 3 murs, 1 récepteur GPS, le tout avec une très grande incertitude de pose initiale. La pose réelle est en  $(30m, 30m, 0^\circ)$ . La position estimée est en  $(15m, 0m, 10^\circ)$  avec un écart-type de  $(100m, 100m, 20^\circ)$ . La pose finale est en  $(30m, 30m, 0^\circ)$ .

#### VI.8.5 Simulation UWB, lidar-mur, GPS avec pose initiale précise mais fausse

Ici, nous réalisons une simulation similaire mais la pose initiale est fausse : centrée sur  $(-50m, 20m, 10^\circ)$  avec une incertitude  $(5m, 5m, 20^\circ)$ . La confiance dans cette pose est cependant initialisée à 50%, celle du GSP à 90%. La pose réelle est toujours en  $(30m, 30m, 0^\circ)$ .

On voit dans la figure VI.53-a- la situation initiale : faible incertitude sur la pose estimée pourtant fausse. Le prochain triplet à utiliser sera le GPS. Après mise à jour (fig. VI.53-b-), le processus du triplet GPS a conduit à privilégier l’hypothèse 2 (mesure GPS sans fusion). Ensuite ce sera un triplet mur qui sera privilégié par l’algorithme. En effet la configuration initiale (notamment l’orientation initiale) a conduit à ce choix. La mise

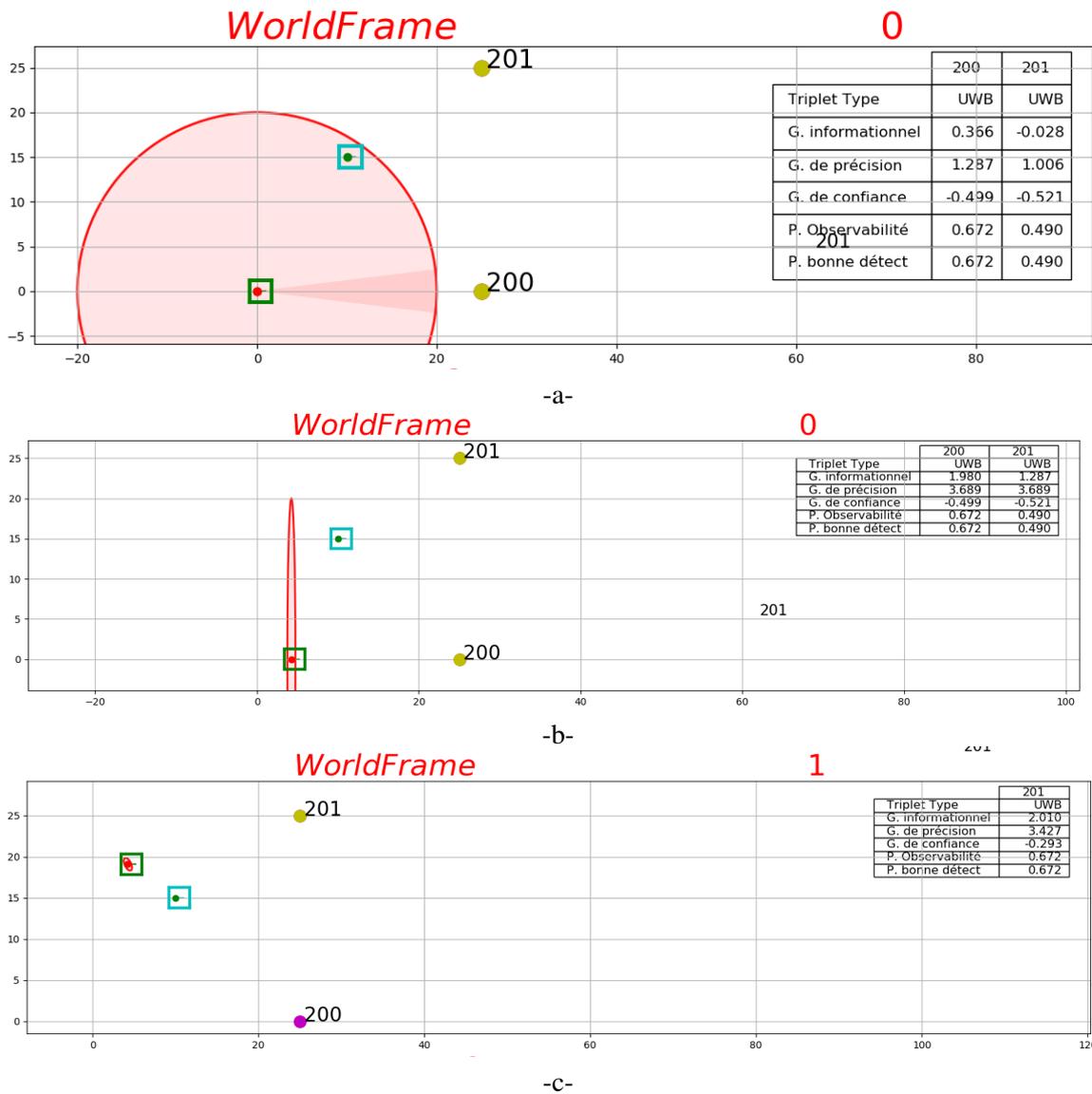


FIGURE VI.50 – Simulation avec 2 balises et une *mise à jour de type Kalman*. La pose réelle est donnée encore par le rectangle cyan en (10m, 15m, 0°). La pose estimée est en (0,0,0) : carré vert. Le première mesure (balise 200) contraint l'estimation initiale (figure -a-) à une ellipse fine et allongée qui ne contient pas la vraie pose (figure -b-). La seconde mesure est faite par la balise 201 qui met à jour cet l'état de manière précise mais malheureusement erronée (figure -c-).

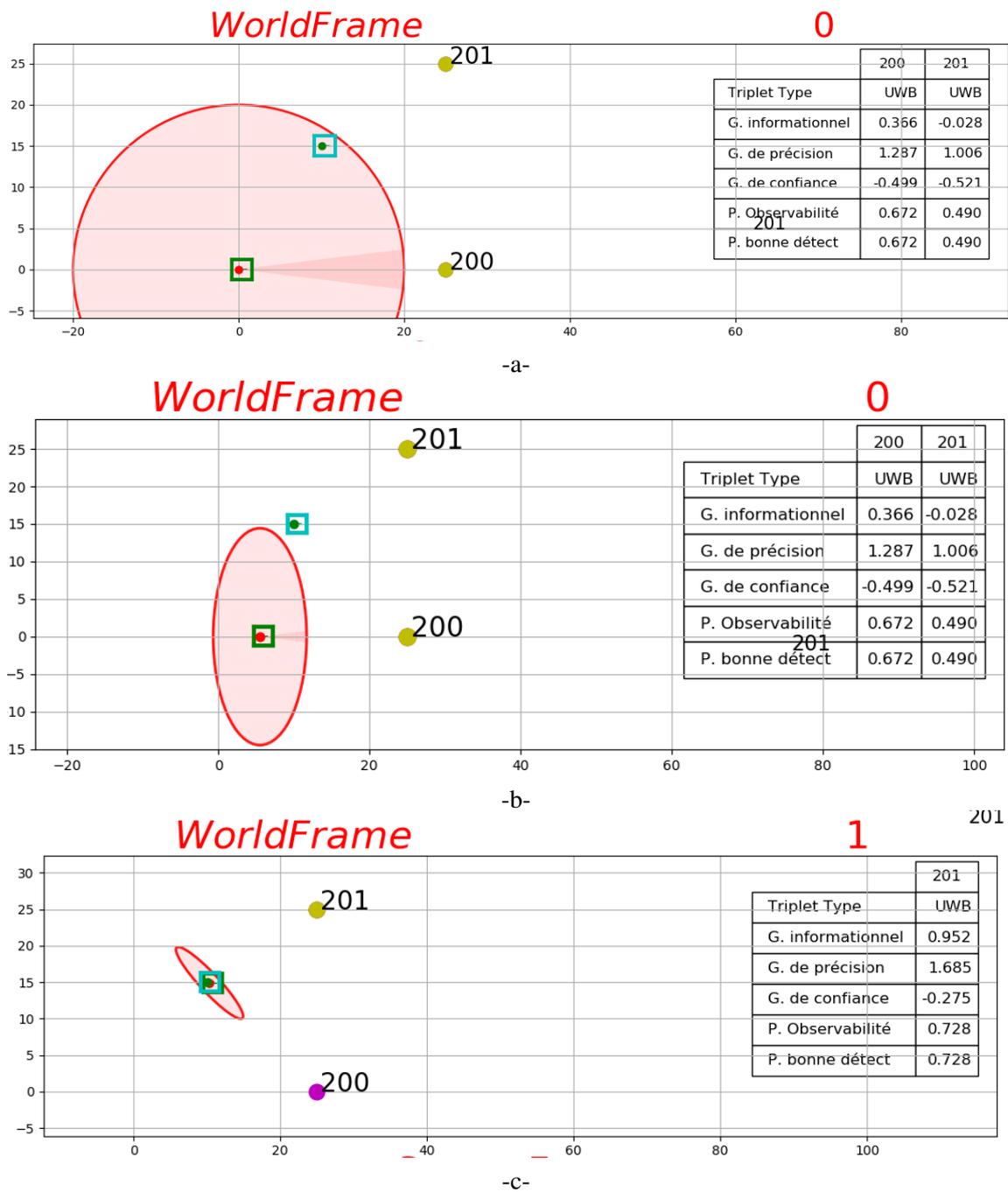


FIGURE VI.51 – Simulation avec 2 balises et une **mise à jour de type UKF adaptée**. La pose réelle est donnée encore par le rectangle cyan en  $(10m, 15m, 0^\circ)$ . La pose estimée est en  $(0,0,0)$  : carré vert. Le première mesure (balise 200) contraint l'estimation initiale (figure -a-) à une ellipse qui n'englobe pas encore la vraie pose à un écart-type mais qui en est proche (figure -b-). La seconde mesure est faite par la balise 201 qui met à jour cet l'état de manière moins précise que précédemment mais qui est cette fois juste (figure -c-)

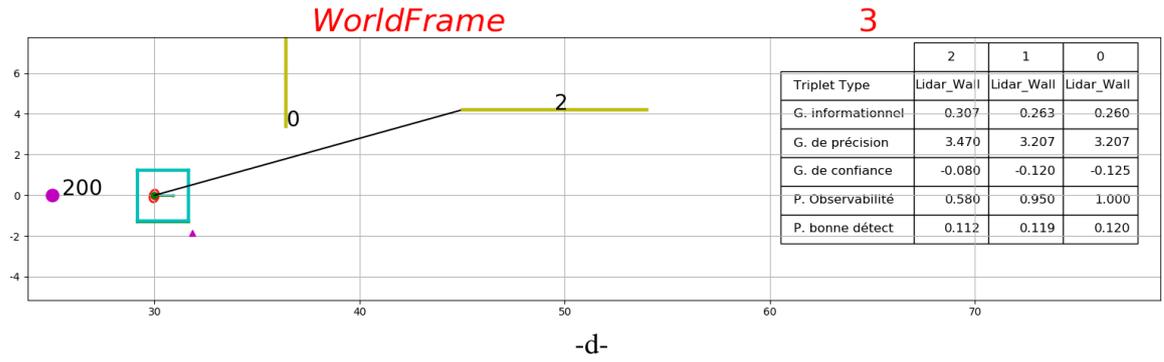
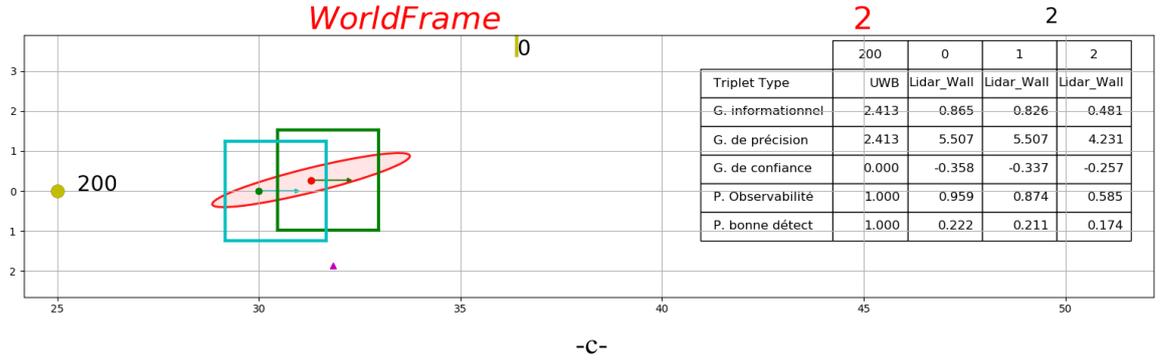
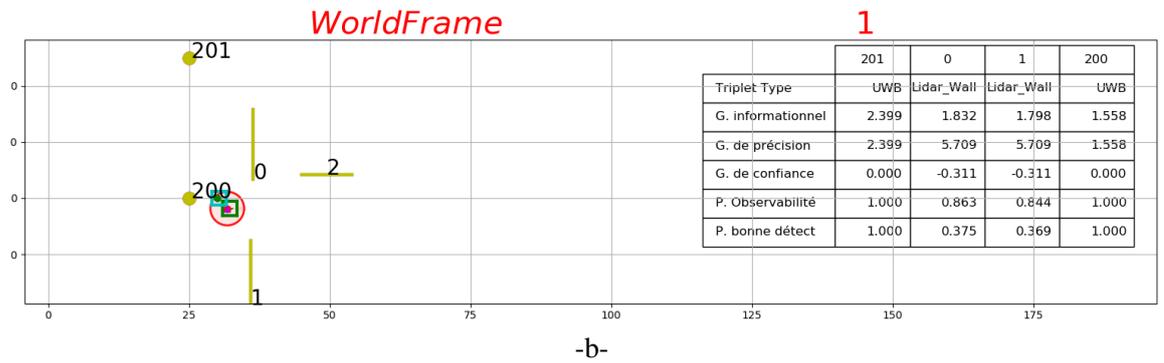
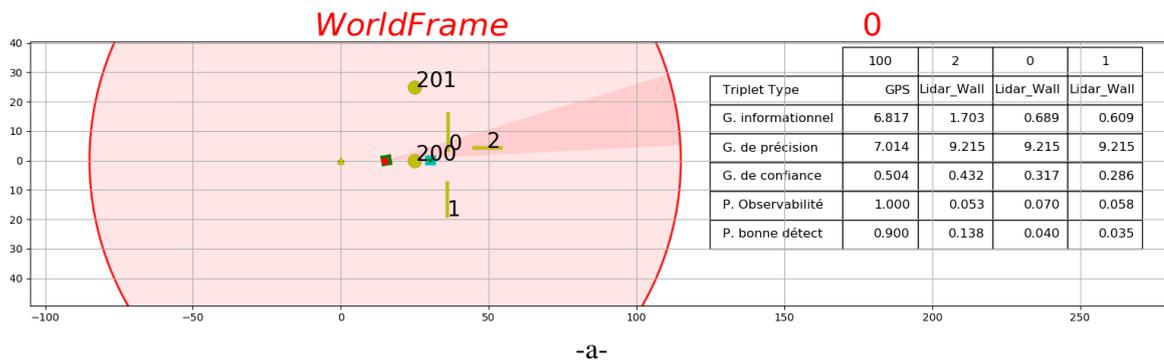


FIGURE VI.52 – Simulation avec 2 balises, 3 mur et un récepteur GPS. La pose initiale est très imprécise (figure -a-). Le récepteur GPS est choisi logiquement en premier (de par son observabilité) et conduit à une mise à jour d'écart-type de l'ordre de 3m (celui du récepteur GPS). La balise 201 est ensuite sélectionnée pour conduire à l'estimation montrée en figure -c-. On voit dans cette figure que ce sera la balise 200 qui sera la plus pertinente ensuite. Logiquement, le mur 2 est choisi ensuite (figure -d-), ce qui conduira à atteindre le critère.

à jour se poursuit ensuite comme à l'habitude ( figure VI.53-c-) et la reconnaissance est réalisée en 4 itérations sans retour arrière.

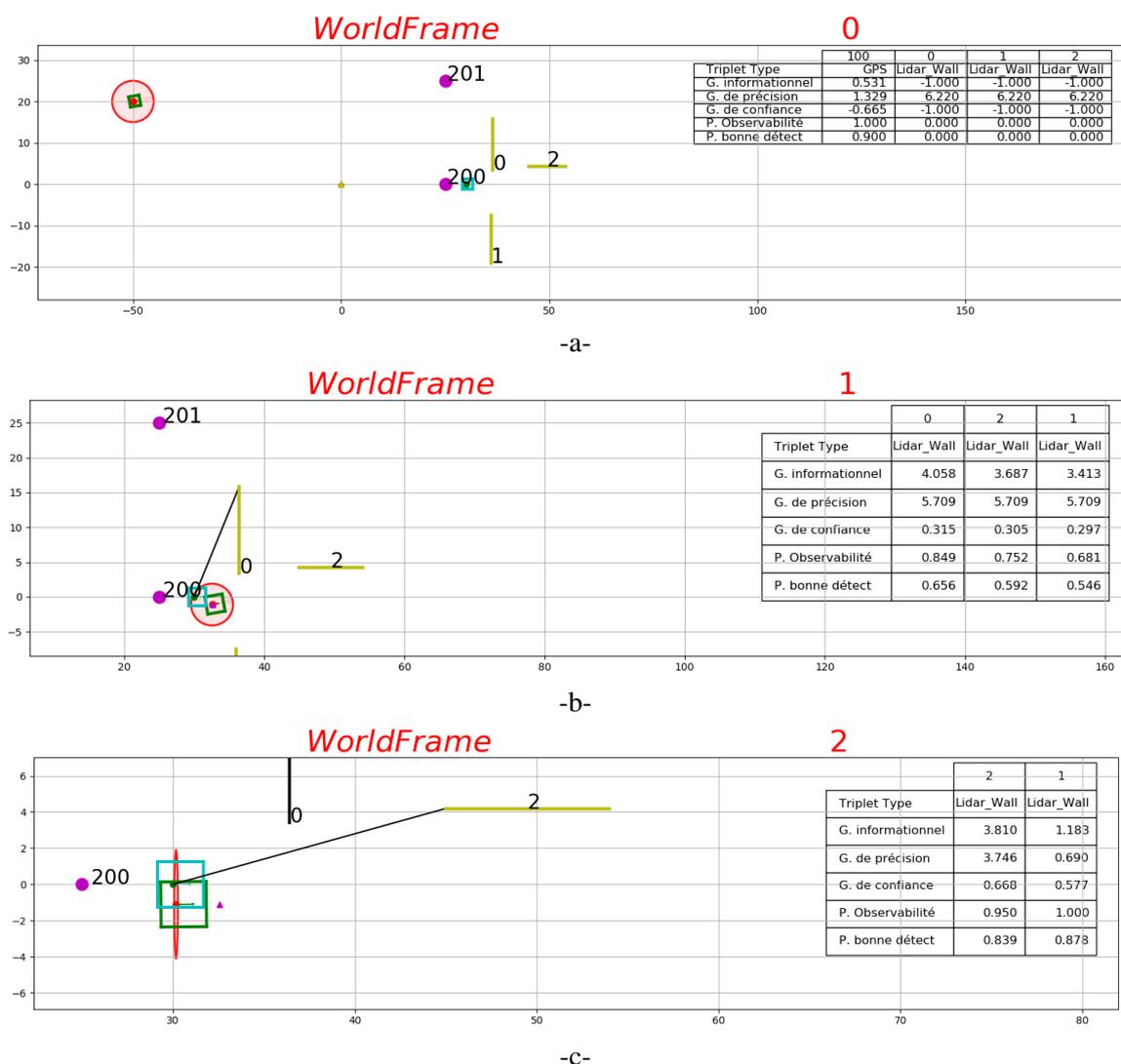


FIGURE VI.53 – Simulation avec 2 balises, 3 mur et un récepteur GPS. La pose initiale est assez précise **mais fausse** avec une confiance de 50% (figure -a-). Le récepteur GPS est choisi logiquement en premier et conduit à une mise à jour d'écart-type de l'ordre de 3m mais avec un choix de l'hypothèse de fusion 2 (voir V.6.2.2) qui conduit à ne pas réaliser la fusion avec l'état à priori mais à choisir la mesure GPS pour réinitialiser cet état (celui du récepteur GPS). Ensuite ce sera un triplet mur qui sera privilégié par l'algorithme (figure -c-) et la reconnaissance sera réalisée en 4 itérations sans retour arrière.

## VI.8.6 Bilan sur l'ajout du triplet UWB

Dans cette section nous avons montré l'intérêt d'ajouter un triplet de type UWB. En effet, ce dernier présente l'avantage de ne pas présenter des ambiguïtés. Cependant, l'inconvénient de ce triplet utilisé avec un filtrage de type Kalaman étendu est que le processus de focalisation va très probablement conduire à une zone d'analyse très fine et finalement fausse du triplet courant suite aux problèmes de non linéarité caractéristiques de ce type de capteur. La solution sera d'utiliser plutôt un filtrage de type UKF. Une fois ces problèmes de non linéarité s'amenuisent (puisque les incertitudes correspondantes deviennent plus petites), la sélection d'un triplet UWB permettra par la suite d'éviter des situations ambiguës et améliorer la performance de l'estimation retournée.

## VI.9 Conclusion

Les résultats de cette méthode sont très prometteurs et valident selon nous la pertinence d'une approche top-down pour l'application de localisation de véhicules dans un milieu présentant des ambiguïtés. L'approche proposée est basée sur l'utilisation d'un critère informationnel pour la sélection de la donnée la plus pertinente.

L'application des travaux montrés dans ce rapport est de proposer une méthode permettant d'assurer une estimation précise et fiable de la pose initiale du robot ainsi que donner une solution au problème des situations ambiguës. Le processus de retour réagit de manière efficace en fonction des objectifs de précision et de confiance fixés préalablement.

Par ailleurs, le fait de gérer la confiance à chaque instant permet de sélectionner une hypothèse de fusion pertinente dans le cas du GPS, ce qui est impossible sans cela.

Nous avons montré aussi la prise en compte de mesures de type range-only avec des capteurs de type UWB. La mise à jour par un filtre de type UKF permet d'éviter les convergences dues aux non linéarités caractéristiques de type ce capteur. Le fait de disposer là aussi d'une analyse de pertinence des triplets avec cette approche de fusion conduit à leur choix lorsque ces problèmes de non linéarité s'amenuisent puisque les incertitudes correspondantes deviennent plus petites.

Enfin dans un cadre plus général nous avons montré que l'intégration de plusieurs types de triplet, de par le formalisme unifié proposé est relativement simple dès lors qu'un triplet donné peut être considéré comme un agent informationnel doté de méthode de calcul de pertinence, de détection et de mise à jour.

# Chapitre VII

## Conclusion et perspectives

### Sommaire

---

VII.1 Conclusion . . . . .	147
VII.2 Perspectives . . . . .	148
VII.2.1 Autres types de triplets . . . . .	148
VII.2.2 Prise en compte du temps d'arrivée de la donnée . . . . .	148
VII.2.3 Correction et enrichissement de la carte . . . . .	148
VII.2.4 Triplet de construction de carte . . . . .	149

---

## VII.1 Conclusion

Dans ce manuscrit, nous nous sommes intéressés à la localisation d'un robot mobile à partir d'une carte. Nous avons vu dans le [chapitre II](#) que différentes méthodes sont utilisées dans le domaine de la localisation de robots mobiles à savoir les méthodes de localisation relative et absolue. Un bilan comparatif conclut sur la pertinence de l'utilisation de techniques de fusion afin de tirer profit de la richesse des données et exploiter au mieux les capteurs embarqués sur le robot. Pour cela, il nous semblait pertinent de s'orienter sur une approche multi-sensorielle descendante (top-down) afin d'exploiter au mieux la diversité des informations disponibles.

Dans le [chapitre III](#) nous avons conclu sur la pertinence d'utiliser une carte sémantique afin de répondre à notre objectif d'architecture générique et modulaire. En effet, ce type de carte présente l'avantage de contenir une grande quantité d'informations fournie par les primitives cartographiées. Ce caractère multi-amer permet une gestion unifiée et générique des données. Cependant, l'utilisation de carte requiert une gestion robuste de différentes situations présentant des ambiguïtés, d'où l'intérêt du double objectif que nous nous sommes fixés. En effet, l'objectif de notre application de localisation doit répondre à deux critères principaux qui sont la précision et la confiance. Cela veut dire qu'en plus d'assurer une estimation précise de la pose du robot, il faut en plus caractériser la confiance que l'on a dans cette estimation.

Dans le [chapitre IV](#), nous avons donné une description générique de notre approche. Celle-ci est basée sur un formalisme de nature probabiliste qui se prête bien à différentes approches pour gérer différents types de données. Elle est aussi basée sur un cadre de triplet perceptif introduit par les travaux antérieurs ([Aynaoud, 2015](#)) et ([Delobel, 2018](#)). Les différents triplets perceptifs mis en œuvre dans nos travaux ont pour objectif d'exploiter les différentes modalités capteurs dont dispose le robot en détectant des éléments variés de son environnement. Afin de sélectionner la meilleure information à utiliser pour estimer la position du robot, nous utilisons un critère de pertinence basé sur la théorie de l'information qui unifie les deux objectifs de précision et de confiance.

Dans le [chapitre V](#), nous avons montré comment décrire divers types de triplets de manière à les rendre compatibles avec notre approche. Nous avons aussi optimisé la technique de mise à jour des données par UWB par un filtre de type UKF limitant les effets de non linéarité et conduisant à une baisse de précision propice à une sélection plus pertinente de ce type de triplet. Grâce à cette approche reposant sur l'analyse de confiance nous avons proposé, pour les capteurs de type GNSS, une technique de fusion plus pertinente qu'un simple filtrage bayésien.

Dans nos travaux, la gestion des situations ambiguës est gérée par différentes techniques. D'une part, le critère informationnel de sélection permet de sélectionner la donnée la moins ambiguë. Ceci est assurée par un réseau bayésien qui prend en compte l'ensemble des amers compatibles avec l'amer recherché. D'autre part, lors de l'étape de détection, la focalisation spatiale et celle paramétrique permettent de réduire le problème de mauvaises détections en réduisant la taille de la zone de recherche de l'amer à détecter. Néanmoins dans certains cas, cela s'avère insuffisant lorsque les situations rencontrées sont fortement ambiguës. Par conséquent, des échecs de détection peuvent avoir lieu ou bien des mauvaises associations de données sont établies. Dans les deux cas, la performance de notre application sera dégradée et les objectifs ne seront pas atteints. Pour cela, un processus de retour en arrière a été mis en place afin de remettre en cause l'échec de détection d'une part et d'autre part d'améliorer la qualité de l'estimation dans le cas où cette dernière ne répond pas aux objectifs de précision et/ou de confiance fixés préalablement.

Dans le [chapitre VI](#), nous avons montré que cette technique permet de résoudre des scénarios critiques notamment l'estimation de la pose du robot dans des situations très ambiguës. Elle permet aussi d'éviter une phase d'initialisation spécifique pour l'estimation de la pose lorsque l'incertitude initiale est très grande. L'aspect générique (approche insensible au type du capteur utilisé) et multisensoriel (plusieurs modalités capteurs peuvent être exploitées) de l'approche confirme son grand apport dans le domaine de la localisation mobile. Les résultats montrés ont permis de mettre en avant son efficacité. Ce travail a fait l'objet de la publication ([Ladhari et al., 2019](#)) et d'un article journal en cours d'écriture.

## VII.2 Perspectives

Les résultats de cette méthode sont très prometteurs et valident la pertinence d'une approche top-down pour l'application de localisation sur carte de robots mobiles dans un milieu présentant des ambiguïtés. Nous présentons dans la suite des perspectives que ces travaux permettent d'envisager et d'apporter une plus value potentielle.

### VII.2.1 Autres types de triplets

À l'issue du formalisme du triplet perceptif utilisé dans nos travaux, il est tout à fait judicieux d'ajouter d'autres types de triplets exploitant différentes modalités capteurs et d'autres types d'amers. Par exemple en ajoutant un capteur caméra, nous pourrions imaginer les triplets suivants :

- {Capteur caméra, détecteur de panneaux de signalisation, panneau de signalisation}
- {Capteur caméra, détecteur d'enseigne, enseigne}
- {Capteur caméra, détecteur de segment vertical, poteau}
- {Capteur caméra, détecteur de ligne blanche, marquage au sol}

En effet, l'ajout d'un capteur caméra permet de mieux exploiter la richesse des données stockées dans la carte en détectant des indices visuels saillant de la scène comme par exemple les enseignes lumineuses ou le château d'eau.

### VII.2.2 Prise en compte du temps d'arrivée de la donnée

Les travaux présentés jusqu'à présent sont réalisés en mode statique sans utilisation d'un modèle d'évolution. L'estimation dynamique peut se faire avec notre approche de manière très simple en exploitant un modèle dynamique du robot entre chaque itération. La confiance dans l'estimation ne changera pas, seule la covariance de l'estimation de pose va évoluer.

Pour un système de localisation basé sur la fusion de données issues de plusieurs sources, la prise en compte du caractère désynchronisé des mesures capteurs du système est cependant primordiale. En effet, dans la pratique, les mesures issues des capteurs n'arrivent pas au même instant d'où le problème d'association asynchrone de données qui arrivent de manière disjointe dans le temps. Pour cela, la considération des délais d'acheminement des données ainsi que de la latence due au traitement de la donnée sont nécessaires.

La prise en compte du temps d'arrivée de la donnée importe ainsi sur le critère de sélection du meilleur triplet que nous avons introduit dans nos travaux. En effet, l'utilisation de plusieurs détecteurs peut engendrer un temps d'exécution non négligeable, ce qui affectera par la suite la disponibilité du triplet à utiliser. D'autre part, la prise en compte du temps d'arrivée doit être aussi considérée lors de l'exécution du processus de retour en arrière. En effet, pour corriger une mauvaise association de données, une sauvegarde des temps d'arrivée des données est ainsi nécessaire afin d'être capable de revenir sur une telle association et de modifier le choix effectué.

### VII.2.3 Correction et enrichissement de la carte

Bien que la carte que nous avons utilisée présente des ambiguïtés, elle est toutefois très précise. Cependant, un inconvénient lié à ce type de carte est sa disponibilité en tout lieu et en tout temps. C'est pourquoi il sera intéressant d'appliquer le principe de notre méthode avec des cartes déjà fournies de type Open Street Map. Ce

type de carte permettra d'assurer une localisation à peu près n'importe où du robot.

Toutefois, comme nous l'avons évoqué dans le [chapitre III](#), ce type de carte présente des informations cartographiques qui sont plutôt imprécises et potentiellement erronées. Il peut arriver que la position estimée du robot soit plus précise que celles des amers de la carte. L'idée sera donc de tirer profit, d'une part, de la bonne précision de l'estimation retournée et améliorer la précision de la carte et d'autre part de la confiance que l'on a dans cette estimation pour améliorer la fiabilité des données de la carte. En effet, outre l'imprécision des données de la carte, il se peut très bien qu'un amer de la carte soit absent ou bien qu'il ne soit pas cartographié.

L'objectif ici serait d'intégrer des travaux antérieurs de l'équipe ([Delobel, 2018](#)) visant à :

1. corriger la position des amers de la carte
2. supprimer des amers qui n'existent plus dans la carte
3. ajouter des amers détectables et non encore insérés dans la carte

#### **VII.2.4 Triplet de construction de carte**

La carte sémantique que nous avons utilisée est supposée connue et définie dans un référentiel absolu. Toutefois, il est judicieux d'utiliser une carte dans un référentiel relatif local comme celle retournée par un algorithme SLAM par ajustement de faisceaux. En effet, l'utilisation des deux référentiels permettra une meilleure exploitation des informations de position relative et absolue.

D'autre part, dans le cas où la carte n'est pas fournie, un triplet de type SLAM permettra de créer une carte et de l'utiliser pour se localiser en temps réel. Ceci peut être facilement intégré dans nos travaux grâce à l'aspect générique et multi sensoriel de la méthode.

# Chapitre VIII

## Annexes

### VIII.1 Introduction à la Théorie de l'Information

Développée par C. E. Shannon depuis son premier exposé (Shannon, 1948), cette théorie faisant largement appel à la théorie des probabilités décrit les aspects fondamentaux des systèmes de communications. En revanche, le domaine d'application va bien au delà de celui-ci comme : l'informatique théorique, la compression des données, la thermodynamique, la génétique, etc. Il s'agit en fait d'une théorie mathématique de la communication de l'information comme l'a introduite Claude Shannon et Warren Weaver. Une des questions à laquelle répond cette théorie est : *Quelle est la quantité d'information et de redondance apportée par une source de données ?*

Une source d'information est définie par un ensemble d'événements possibles pouvant apparaître d'une manière aléatoire. Nous nous intéressons ici aux sources discrètes (un espace probabilisé discret) et sans mémoire (la probabilité que l'information arrive à un instant n'est pas dépendante de l'instant précédent). Bien que la notion d'information d'une source est difficile à quantifier, elle représente toutefois une mesure de son imprévisibilité. Cette notion est ainsi liée à celle de probabilité. Pour cela, une source est modélisée par une expérience aléatoire décrite par la donnée d'un espace des épreuves  $A = \{a_1, \dots, a_n\}$  (appelé alphabet de la source et représentant l'ensemble des résultats possibles de l'expérience). On considère que pour chaque résultat possible  $a_i$  (aussi appelé issue), on dispose de sa probabilité d'apparition  $P(a_i)$ . On aura donc :

$$\sum_{i=1}^n P(a_i) = 1$$

#### VIII.1.1 Incertitude et information

##### VIII.1.1.1 Description qualitative de l'information

*Qualitativement, l'information caractérise la diminution de l'incertitude sur l'issue d'une expérience aléatoire.* Prenons un exemple simple. Imaginons un système produisant 3 symboles A, B et C de manière équiprobable. Dans ce cas, on pourrait dire que nous avons une incertitude de 3 symboles. Cette incertitude est donc importante. Néanmoins, l'observation d'un symbole A émis par le système augmente la probabilité qu'un A ait été émis et diminue la probabilité qu'un B ou C aient été émis. Par conséquent, une fois qu'on a reçu un symbole, cette incertitude devient nulle : c'est une *information*.

D'une manière générale et pour deux événements  $X = x$  et  $Y = y$ , la probabilité conditionnelle  $p(x|y)$  caractérise la modification apportée par l'information de la réalisation de l'événement  $y$  (l'information 'y est réalisé') sur la probabilité de l'événement  $x$ , c'est à dire l'incertitude de la réalisation de l'événement  $x$ , de  $p(x)$  à  $p(x|y)$ .

La diminution de l'incertitude d'un événement sera par conséquent interprétée comme un gain d'information. En effet :

- si  $p(x|y) \geq p(x)$  : il s'agit d'un gain d'information (une diminution de l'incertitude)
- si  $p(x|y) \leq p(x)$  : il s'agit d'une perte d'information (une augmentation de l'incertitude)

Nous désignons par  $I(x)$  la quantité d'information apportée par un événement  $x$ . Cette fonctionnelle appelée information propre est fonction de la probabilité  $I(x) = f(p(x))$ . Dans le paragraphe suivant, nous donnerons une quantification de cette information.

### VIII.1.1.2 Description quantitative de l'information

Dans le paragraphe précédent nous avons montré que l'information s'écrit sous la forme  $I(x) = f(p(x))$  avec :

- la fonction  $f$  est une fonction décroissante de  $p(x)$  dans la mesure où un événement très probable n'apporte pas d'information et inversement un événement incertain apporte de l'information. Par conséquent  $I(x) = f(\frac{1}{p(x)})$
- la fonction  $f$  est une fonction additive c'est à dire pour deux événements  $x$  et  $y$  statistiquement indépendants :  $f(p(x, y)) = f(p(x)) + f(p(y))$ . Pour affirmer cela, reprenons l'exemple du paragraphe précédent et supposons qu'un deuxième système produit 2 symboles 1 ou 2 en parallèle avec le premier système. Dans ce cas, nous aurons une incertitude de 6 symboles ( $A1, A2, B1, B2, C1, C2$ ) et nous pourrions par conséquent penser que nous avons reçu plus d'information. Afin de faire en sorte que l'information soit additive, elle sera par conséquent caractérisée par un logarithme :  $I(x) = \lambda \log p(x)$  avec  $\lambda < 0$  pour assurer la décroissance par rapport à  $p(x)$ . Le choix de  $\lambda$  dépend de l'unité d'information. Nous choisissons un logarithme en base 2 pour obtenir une unité en bits.
- $f(1) = 0$  : en effet l'information apportée par un événement certain est nulle.

Par conséquent, la quantité d'information associée à la réalisation d'un événement  $x$  pourra être écrite sous la forme donnée par l'équation VIII.1. Dès lors, l'information est donnée par l'incertitude qui est calculée à partir de la probabilité. Plus l'information est incertaine plus elle est intéressante. Dans cette théorie, c'est un raisonnement par probabilité qui est utilisé pour calculer l'apport informationnel d'une source de données.

Nous précisons qu'afin d'assurer la cohérence des calculs,  $I(x)$  doit être exprimée dans la même base. En effet cette fonctionnelle peut être donnée sous différentes unités :

- en bits : pour un logarithme en base 2
- en nats : pour un logarithme en base naturelle  $e$
- en bans : pour un logarithme en base décimale

$$I(x) = -\log_2(P(x)) \quad (bits) \tag{VIII.1}$$

**Exemple :** Soit  $A = a_0, a_1, \dots, a_{15}$  un alphabet de  $n = 16$  lettres qui sont *équiprobables*. L'information propre d'une lettre quelconque  $a_i$  sera donnée par :

$$I(a_i) = -\log_2\left(\frac{1}{16}\right) = 4 \text{ bits}$$

### VIII.1.2 Entropie d'une source

À chaque symbole d'un alphabet  $A$  issu d'une source donnée est associée une quantité d'information moyenne. Elle est donnée par l'équation VIII.2 où  $p_i$  représente une loi de probabilité *discrète* (la probabilité associée à l'apparition de l'événement  $i$ ). L'entropie est toujours positive ou nulle. Elle sera nulle si et

seulement si l'un des événement est certain (de probabilité  $p_i = 1$ ) et les autres sont de probabilités nulles. Dès lors, l'entropie mesure une quantité liée à l'incertitude ou à la méconnaissance du système à partir d'une loi de probabilité donnée. Pour un alphabet de cardinal  $n$ , cette fonctionnelle est bornée au niveau supérieur comme montré par l'équation VIII.3 avec égalité si et seulement si la loi de probabilité de  $A$  est uniforme. Une démonstration de cette relation peut être trouvée dans (Sendrier, 2007).

$$H(A) = \mathbb{E}[I(A)] = \sum_{i \in A} p_i \log_2(p_i) \geq 0 \quad (\text{VIII.2})$$

$$H(A) \leq \log_2(n) \quad (\text{VIII.3})$$

**Exemple d'une source binaire :** Dans le cas de  $n = 2$  symboles :  $A = a, b$  dont la probabilité serait  $P(a)$  et  $P(b)$  respectivement, l'allure de l'entropie  $H(A)$ , donnée par l'équation VIII.4, sera représentée par la figure VIII.1. Nous remarquons que l'entropie est maximale lorsque les deux événements  $a$  et  $b$  sont équiprobables et elle est nulle lorsque la probabilité de  $a$  ou de  $b$  est nulle.

$$H(A) = -(p \log_2(p) + (1 - p) \log_2(1 - p)) \quad (\text{VIII.4})$$

avec  $p = P(a) = 1 - P(b)$

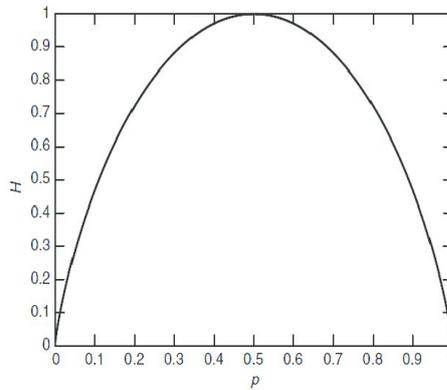


FIGURE VIII.1 – Évolution de l'entropie  $H(A)$  en fonction de  $P(a) = 1 - P(b)$  (source : (Liu et al., 2016))

### VIII.1.3 Entropie différentielle

C'est une généralisation de l'entropie de Shannon du paragraphe précédent aux cas de distributions de probabilités *continues*. Pour une variable continue de densité de probabilité  $p$ , il s'agit de remplacer la sommation dans l'équation VIII.2 par une intégration. La formule générale de l'entropie différentielle sera ainsi donnée par l'équation VIII.5. Toutefois, cette fonctionnelle n'est ni positive ni bornée.

$$H_p(X) = \int p(x) \log_2(p(x)) dx \quad (\text{VIII.5})$$

#### Exemple : Cas d'une distribution normale multidimensionnelle

Soit une distribution normale  $\mathcal{N}(X_0, \mathbf{C}_X)$  de dimension  $N$  et de matrice de covariance  $\mathbf{C}_X$ . La densité de probabilité est donnée par l'équation VIII.6. En utilisant la formule de l'équation VIII.5, l'entropie sera donnée selon les équations de VIII.7 à VIII.9. La démonstration peut être trouvée dans (Ahmed and Gokhale, 1989).

$$p(X; X_0, \mathbf{C}_X) = \frac{1}{\sqrt{(2\pi)^N |\mathbf{C}_X|}} e^{-\frac{1}{2}(X-X_0)^T \mathbf{C}_X^{-1} (X-X_0)} \quad (\text{VIII.6})$$

$$\mathbf{H}_p = \ln \left[ \sqrt{(2\pi \cdot e)^N |\mathbf{C}_k|} \right] \quad (\text{VIII.7})$$

$$= \frac{1}{2} [N \ln(2\pi \cdot e) + \ln(|\mathbf{C}_k|)] \quad (\text{VIII.8})$$

$$= 1.4189 \cdot N + \frac{\ln(|\mathbf{C}_k|)}{2} \quad (\text{VIII.9})$$

### VIII.1.4 Information mutuelle

On appelle information mutuelle la quantité d'information qu'apporte la donnée d'une des variables sur une autre. Elle donne en effet le degré de corrélation entre les deux variables. Pour deux événements  $a$  et  $b$ , son expression est donnée par les deux équations [VIII.10](#) et [VIII.11](#).

$$I(a; b) = \log_2 \frac{p(a|b)}{p(a)} \quad (\text{VIII.10})$$

D'après la relation de Bayes :

$$I(a; b) = I(b; a) = \log_2 \frac{p(a, b)}{p(a)p(b)} \quad (\text{VIII.11})$$

Nous pouvons également définir l'information propre conditionnelle de  $a$  sachant  $b$  qui représente la quantité d'information apportée par la réalisation de l'événement  $a$  sachant l'événement  $b$  qui est donnée par l'équation [VIII.12](#).

$$I(a|b) = -\log_2 p(a|b) \quad (\text{VIII.12})$$

Étant données les deux équations [VIII.10](#) et [VIII.12](#), l'information mutuelle peut être aussi écrite sous la forme [VIII.13](#).

$$I(a; b) = I(a) - I(a|b) \quad (\text{VIII.13})$$

Cette fonctionnelle peut être aussi écrite en fonction de l'entropie afin de représenter la quantité d'information que l'on peut obtenir en moyenne sur une grandeur  $a$  connaissant la valeur d'une autre variable  $b$ . Pour cela, une étude plus détaillée sur la théorie de l'information peut être consultée dans ([Delobel, 2018](#)) et ([Sendrier, 2007](#)).

## VIII.2 Analyse géométrique sur les segments du plan

Dans le [chapitre IV](#) nous avons décrit certains triplets, dont le triplet lidar-mur qui impliquait le travail avec des segments du plan. Cette partie récapitule quelques résultats.

### VIII.2.1 Modélisation polaire d'un segment

On recherche la représentation  $(\rho, \theta)$  (voir figure [VIII.2](#)) d'un segment défini par ses coordonnées  $AB$  dans le repère  $(xOy)$ .

On sait que (voir annexe §[VIII.2.2.2](#)) que la distance  $\rho$  du segment à l'origine sera donnée par :

$$\rho = \frac{\|\underline{AO} \wedge \underline{AB}\|}{\|\underline{AB}\|}$$

Puisqu'on est dans le plan, on aura :

$$\begin{aligned} \underline{AO} \wedge \underline{AB} &= AO_x \cdot AB_y - AO_y \cdot AB_x \\ &= A_x(B_y - A_y) - A_y(B_x - A_x) \\ &= A_x B_y - A_y B_x \end{aligned}$$

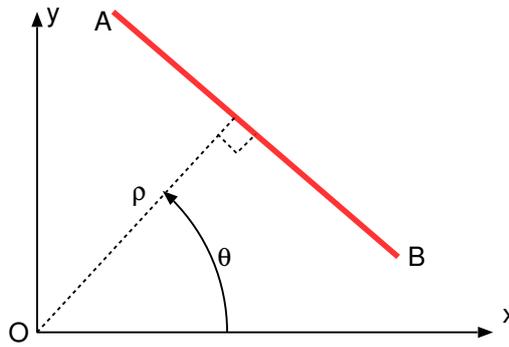


FIGURE VIII.2 – Représentation  $(\rho, \theta)$  d'un segment  $AB$

Et donc :

$$\rho = \frac{\|AO \wedge AB\|}{\|AB\|} = \frac{|A_x B_y - A_y B_x|}{\sqrt{(B_x - A_x)^2 + (B_y - A_y)^2}}$$

L'angle  $\theta$  sera donné par :

$$\theta = \arctan\left(\frac{B_y - A_y}{B_x - A_x}\right) + \frac{\pi}{2}$$

Ainsi :

$$\rho = \frac{A_x B_y - A_y B_x}{\sqrt{(B_x - A_x)^2 + (B_y - A_y)^2}} \quad \text{et} \quad \theta = \arctan\left(\frac{B_y - A_y}{B_x - A_x}\right) + \frac{\pi}{2} \quad (\text{VIII.14})$$

### VIII.2.2 Distance d'un point à un segment

On cherche la distance d'un point  $P$  à un segment donné  $[AB]$  (voir fig. VIII.3).

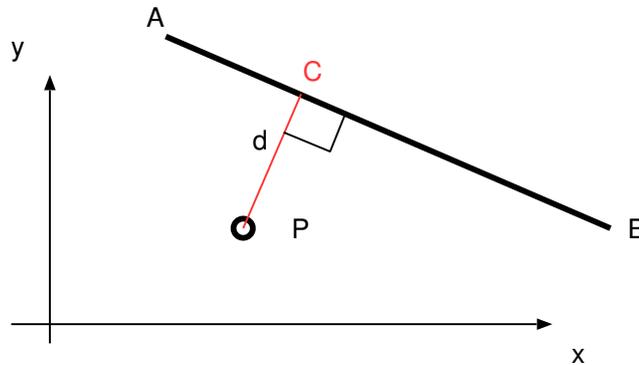


FIGURE VIII.3 – Position du problème : on recherche la distance entre le point  $P$  et le segment

#### VIII.2.2.1 Test de l'appartenance du point $C$ au segment $[AB]$

Il faudra tout d'abord tester si le point  $C$  (voir figure VIII.3) est dans le segment ou non. Pour cela, on testera si la projection du segment  $[AP]$  sur le segment  $[AB]$  a une longueur  $AC$  telle que :

$$l > 0 \quad \text{et} \quad l < AB \quad \text{avec} \quad l = AC = \frac{AP \cdot AB}{\|AB\|}$$

Le point  $C$  appartiendra donc au segment  $[AB]$  si :

$$0 < \frac{AP \cdot AB}{\|AB\|^2} < 1$$

### VIII.2.2.2 Distance de $C$ au segment $[AB]$ dans le cas où le point $C$ est entre $A$ et $B$

Dans ce cas, il est connu que :

$$d = \frac{\|\underline{AP} \wedge \underline{AB}\|}{\|\underline{AB}\|}$$

### VIII.2.2.3 Cas où le point $C$ n'est pas entre $A$ et $B$

Dans ce cas, le point  $C$  sera plus proche soit de  $A$  soit de  $B$ . Il sera plus proche de  $A$  si la distance  $l = AC < 0$  et sinon sera plus proche de  $B$  si  $l > \|\underline{AB}\|$ .

### VIII.2.2.4 Récapitulatif

La distance  $d$  d'un point  $P$  à un segment  $[AB]$  sera telle que (figure VIII.3) :

$$d = \begin{cases} \|\underline{AP}\| & \text{si } \underline{AB} \cdot \underline{AP} < 0 \\ \|\underline{BP}\| & \text{si } \underline{AB} \cdot \underline{AP} > \|\underline{AB}\|^2 \\ \frac{\|\underline{AP} \wedge \underline{AB}\|}{\|\underline{AB}\|} & \text{sinon} \end{cases}$$

### VIII.2.3 Point d'intersection d'un segment avec un autre

Nous cherchons les coordonnées du point  $P$  d'intersection entre les deux segments  $[AB]$  et  $[CD]$ . Considérons la figure VIII.4.

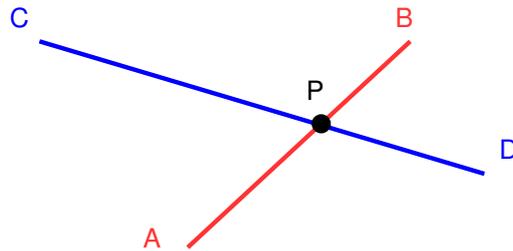


FIGURE VIII.4 – Point d'intersection de deux segments  $[AB]$  et  $[CD]$

Comme  $\underline{AP}$  et  $\underline{AB}$  puis  $\underline{CP}$  et  $\underline{CD}$  sont alignés, on aura :

$$\underline{AP} = \alpha \underline{AB} \quad \text{et} \quad \underline{CP} = \beta \underline{CD}$$

On a aussi :

$$\underline{AP} = \underline{AC} + \underline{CP}$$

et donc :

$$\alpha \underline{AB} = \underline{AC} + \beta \underline{CD}$$

En notant  $\underline{V} = (V_x, V_y)$ , on aura :

$$\begin{cases} AP_x = AC_x + \beta CD_x \\ AP_y = AC_y + \beta CD_y \end{cases}$$

La solution à ce système est :

$$\begin{cases} \alpha = \frac{CD_x \cdot AC_y - AC_x \cdot CD_y}{den} \\ \beta = \frac{AB_x \cdot AC_y - AC_x \cdot AB_y}{den} \end{cases} \quad \text{avec} \quad den = AB_y \cdot CD_x - AB_x \cdot CD_y$$

Ainsi, puisque  $\underline{AP} = \alpha \underline{AB}$ , on aura :

$$P - A = \alpha \underline{AB} \implies P = \alpha \underline{AB} + A$$

Soit :

$$\begin{pmatrix} P_x \\ P_y \end{pmatrix} = \begin{pmatrix} \frac{AB_x(CD_x \cdot AC_y - AC_x \cdot CD_y)}{den} + A_x \\ \frac{AB_y(CD_x \cdot AC_y - AC_x \cdot CD_y)}{den} + A_y \end{pmatrix} \quad \text{avec} \quad den = AB_y \cdot CD_x - AB_x \cdot CD_y$$

### VIII.2.4 Intersection d'une droite avec un segment

On considère ici une droite passant par un point  $O$  et dont on connaît l'angle  $\theta$ , et qui intersecte un segment  $[AB]$ . On cherche la distance d'un point  $P$  à l'origine  $O$  comme montré en figure VIII.5.

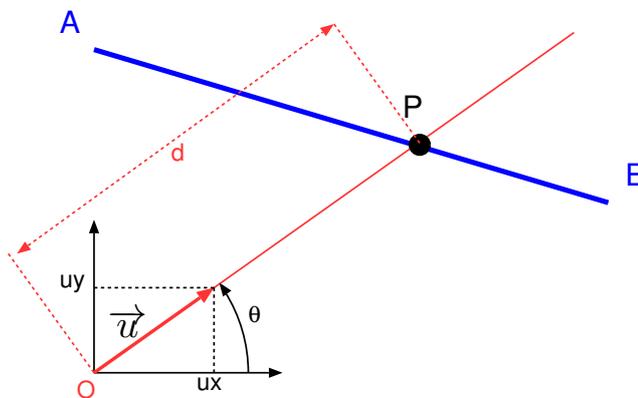


FIGURE VIII.5 – Intersection d'une droite passant par l'origine  $O$  avec un segment  $AB$

On considère le vecteur unitaire  $\underline{u} = (\cos(\theta), \sin(\theta)) = (u_x, u_y)$ . On aura ainsi :

$$\underline{OP} = \alpha \underline{u} \quad \text{et} \quad \underline{AP} = \beta \underline{AB}$$

Comme  $\underline{OP} = \underline{OA} + \underline{AP}$  on aura :

$$\alpha \underline{u} = \underline{OA} + \beta \underline{AB}$$

On aura ainsi le système d'équations suivant :

$$\begin{cases} \alpha u_x = OA_x + \beta AB_x \\ \alpha u_y = OA_y + \beta AB_y \end{cases}$$

Qui a pour solution :

$$\alpha = \frac{OA_x \cdot AB_y - OA_y \cdot AB_x}{u_x AB_y - u_y AB_x} \quad \text{et} \quad \beta = \frac{OA_x \cdot u_y - OA_y \cdot u_x}{u_x AB_y - u_y AB_x}$$

Comme  $\underline{u}$  est unitaire, la distance  $d$  sera donc directement la valeur de  $\alpha$ . Par ailleurs, de manière à vérifier si le point  $P$  est bien sur le segment  $[AB]$  on fera le test  $\beta \in [0, 1]$ . Ainsi :

$$d = \alpha = \frac{OA_x \cdot AB_y - OA_y \cdot AB_x}{u_x AB_y - u_y AB_x} \quad \text{et} \quad \beta = \frac{OA_x \cdot u_y - OA_y \cdot u_x}{u_x AB_y - u_y AB_x} \in [0, 1]$$

### VIII.3 Produit de gaussiennes

Considérons deux gaussiennes  $g_1(x) = \mathcal{N}(\mu_1, \mathbf{C}_1; x)$  et  $g_2(x) = \mathcal{N}(\mu_2, \mathbf{C}_2; x)$  Le produit de ces deux gaussiennes donnera une gaussienne dénormalisée telle que :

$$g(x) = K \cdot \mathcal{N}(\mu, \mathbf{C}; x) = \mathcal{N}(\mu_1, \mathbf{C}_1; x) \cdot \mathcal{N}(\mu_2, \mathbf{C}_2; x)$$

Avec :

$$\mathbf{C} = [\mathbf{C}_1^{-1} + \mathbf{C}_2^{-1}]^{-1}$$
$$\mu = [\mathbf{C}_1^{-1} + \mathbf{C}_2^{-1}]^{-1} [\mathbf{C}_1^{-1} \underline{\mu}_1 + \mathbf{C}_2^{-1} \underline{\mu}_2]$$

Comme il s'agit d'une gaussienne dénormalisée, son expression sera donnée par :

$$g(x) = K \cdot \mathcal{N}(\mu, \mathbf{C}; x) = \frac{K}{\sqrt{(2\pi)^n |\mathbf{C}|}} \exp\left(-\frac{1}{2}(\underline{x} - \underline{\mu})^\top \mathbf{C}^{-1}(\underline{x} - \underline{\mu})\right)$$

Donc pour  $x = \mu$ , nous aurons :

$$g(\mu) = \frac{K}{\sqrt{(2\pi)^n |\mathbf{C}|}} = g_1(\mu) \cdot g_2(\mu)$$

Soit :

$$K = \sqrt{(2\pi)^n |\mathbf{C}|} \cdot g_1(\mu) \cdot g_2(\mu) = \sqrt{(2\pi)^n |\mathbf{C}|} \cdot \mathcal{N}(\mu_1, \mathbf{C}_1; \mu) \cdot \mathcal{N}(\mu_2, \mathbf{C}_2; \mu)$$

# Bibliographie

- Abdelghani, B., Farès, B., Abdelkrim, N., Fethi, D., and Rabah, L. (2018). Stereo visual odometry for mobile robot. In 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), pages 1–7. IEEE.
- Abouzahir, M. (2017). Algorithmes SLAM : Vers une implémentation embarquée. PhD thesis. Thèse de doctorat Robotique Université Ibn Zohr (Agadir) et Robotique Université Paris-Saclay (ComUE) 2017.
- Abuhadrous, I. (2005). Onboard real time system for 3D localization and modelling by multi-sensor data fusion. Theses, École Nationale Supérieure des Mines de Paris.
- Agarwal, H., Tiwari, P., and Tiwari, R. G. (2019). Exploiting sensor fusion for mobile robot localization. In 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pages 463–466. IEEE.
- Agunbiade, O. and Zuva, T. (2018). Simultaneous localization and mapping in application to autonomous robot. In 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC), pages 1–5.
- Ahmed, M., Karagiorgou, S., Pfoser, D., and Wenk, C. (2015). Map construction algorithms. In Map Construction Algorithms, pages 1–14. Springer.
- Ahmed, N. A. and Gokhale, D. (1989). Entropy expressions and their estimators for multivariate distributions. IEEE Transactions on Information Theory, 35(3) :688–692.
- Al Khatib, E. I., Jaradat, M. A., Abdel-Hafez, M., and Roigari, M. (2015). Multiple sensor fusion for mobile robot localization and navigation using the extended kalman filter. In 2015 10th international symposium on mechatronics and its applications (ISMA), pages 1–5. IEEE.
- Alsahfi, T., Almotairi, M., Elmasri, R., and Alshemaimri, B. (2019). Road map generation and feature extraction from gps trajectories data. In Proceedings of the 12th ACM SIGSPATIAL International Workshop on Computational Transportation Science, pages 1–10.
- Ammann, N. and Mayo, L. G. (2018). Undelayed initialization of inverse depth parameterized landmarks in ukf-slam with error state formulation. In 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pages 918–923. IEEE.
- Arras, K. O., Castellanos, J. A., Schilt, M., and Siegwart, R. (2001). Towards feature-based multi-hypothesis localization and tracking. In Fourth European Workshop on Advanced Mobile Robots EUROBOT, pages 57–65. Lund, Sweden.
- Asghar, R., Garzón, M., Lussereau, J., and Laugier, C. (2019). Map Relative Localization Based on Visual and Topological Map Matching. PhD thesis, Inria Chroma.
- Atanassov, V., Borisova, D., Petkov, D., Dimitrov, V., Vasileva, H., and Goranova, M. (2018). Multisensor earth observation systems : data fusion. In Sensors, Systems, and Next-Generation Satellites XXII, volume 10785, page 107851R. International Society for Optics and Photonics.
- Aynaud, C. (2015). Localisation précise et fiable de véhicules par approche multisensorielle. Theses, Université Blaise Pascal - Clermont-Ferrand II.
- Aynaud, C., Bernay-Angeletti, C., Aufrère, R., Lequievre, L., Debain, C., and Chapuis, R. (2017). Real-time multisensor vehicle localization : A geographical information system ? based approach. IEEE Robotics & Automation Magazine, 24(3) :65–74.
- Aynaud, C., Bernay-Angeletti, C., Chapuis, R., Aufrère, R., and Debain, C. (2014a). Approche Top-Down de perception multisensorielle : application à la localisation de véhicules. In Reconnaissance de Formes et Intelligence Artificielle (RFIA) 2014, pages –, France.
- Aynaud, C., Bernay-Angeletti, C., Chapuis, R., Aufrère, R., and Debain, C. (2014b). Vehicle localization by using a multi-modality top down approach. In 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), pages 1415–1420. IEEE.

- Azimirad, E., Haddadnia, J., and Izadipour, A. (2015). A comprehensive review of the multi-sensor data fusion architectures. Journal of Theoretical & Applied Information Technology, 71(1).
- Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam) : Part ii. IEEE robotics & automation magazine, 13(3) :108–117.
- Bastani, F., He, S., Jagwani, S., Park, E., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Madden, S., and Sadeghi, M. A. (2019). Inferring and improving street maps with data-driven automation. arXiv preprint arXiv :1910.04869.
- Bayle, B. (2008). Robotique mobile. Université de Strasbourg, Tech. Rep.
- Becker, C., Ribas, D., and Ridao, P. (2012). Simultaneous sonar beacon localization & auv navigation. IFAC Proceedings Volumes, 45(27) :200–205.
- Ben-Afia, A., Deambrogio, L., Salós, D., Escher, A.-C., Macabiau, C., Soulier, L., and Gay-Bellile, V. (2014). Review and classification of vision-based localisation techniques in unknown environments. IET Radar, Sonar & Navigation, 8(9) :1059–1072.
- Benjamin, B., Erinc, G., and Carpin, S. (2015). Real-time wifi localization of heterogeneous robot teams using an online random forest. Autonomous robots, 39(2) :155–167.
- Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In Sensor fusion IV : control paradigms and data structures, volume 1611, pages 586–606. International Society for Optics and Photonics.
- Betke, M. and Gurdits, L. (1997). Mobile robot localization using landmarks. IEEE transactions on robotics and automation, 13(2) :251–263.
- Bhinge, S., Levin-Schwartz, Y., and Adalı, T. (2017). Data-driven fusion of multi-camera video sequences : Application to abandoned object detection. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1697–1701. IEEE.
- Bian, X. and Li, X. R. (2017). Estimation fusion with data-driven communication. In 2017 20th International Conference on Information Fusion (Fusion), pages 1–8. IEEE.
- Birem, M. (2015). Localisation et détection de fermeture de boucle basées saillance visuelle : algorithmes et architectures matérielles. Theses, Université Blaise Pascal - Clermont-Ferrand II.
- Biswas, J. and Veloso, M. (2010). Wifi localization and navigation for autonomous indoor mobile robots. In 2010 IEEE international conference on robotics and automation, pages 4379–4384. IEEE.
- Bloch, I. (2003). Fusion d'informations en traitement du signal et des images. Hermes Science Publication, 2.
- Bonnifait, P. (1997). Localisation précise en position et attitude des robots mobiles d'extérieur à évolutions lentes. PhD thesis, Nantes.
- Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W., and Teller, S. (2003). An atlas framework for scalable mapping. In 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), volume 2, pages 1899–1906. IEEE.
- Bukhori, I., Ismail, Z., and Namerikawa, T. (2015). Detection strategy for kidnapped robot problem in landmark-based map monte carlo localization. In 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), pages 75–80. IEEE.
- Cabi, S., Colmenarejo, S. G., Novikov, A., Konyushkova, K., Reed, S., Jeong, R., Zolna, K., Aytar, Y., Budden, D., Vecerik, M., et al. (2019). A framework for data-driven robotics. arXiv preprint arXiv :1909.12200.
- Cai, G.-S., Lin, H.-Y., and Kao, S.-F. (2019). Mobile robot localization using gps, imu and visual odometry. In 2019 International Automatic Control Conference (CACS), pages 1–6. IEEE.
- Cai, J. and Zhong, X. (2015). An adaptive square root cubature kalman filter based slam algorithm for mobile robots. In 2015 IEEE International Conference on Mechatronics and Automation (ICMA), pages 2215–2219. IEEE.
- Cappelle, C. (2008). Localisation de véhicules et détection d'obstacles : apport d'un modèle virtuel 3D urbain. PhD thesis, Lille 1.
- Caramazana, L., Arroyo, R., and Bergasa, L. (2016). Visual odometry correction based on loop closure detection. In Open Conference on Future Trends in Robotics (RoboCity16), pages 97–104.
- Castanedo, F. (2013). A Review of Data Fusion Techniques. ISSN : 2356-6140 Library Catalog : www.hindawi.com Pages : e704504 Publisher : Hindawi Volume : 2013.
- Chahine, C. (2016). Fusion d'informations par la théorie de l'évidence pour la segmentation d'images. Theses, Université Paris-Est; École Doctorale des Sciences et de Technologie (Beyrouth).

- Chan, S.-H., Wu, P.-T., and Fu, L.-C. (2018). Robust 2d indoor localization through laser slam and visual slam fusion. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 1263–1268. IEEE.
- Chausse, F., Laneurit, J., and Chapuis, R. (2006). Localisation d'un véhicule sur une carte routière précise. Traitement du Signal, 23, NO 2 :159–178.
- Chen, S. and Chen, C. (2012). Probabilistic fuzzy system for uncertain localization and map building of mobile robots. IEEE Transactions on Instrumentation and Measurement, 61(6) :1546–1560.
- Chen, X., Läbe, T., Milioto, A., Röbling, T., Vysotska, O., Haag, A., Behley, J., Stachniss, C., and Fraunhofer, F. (2020). Overlapnet : Loop closing for lidar-based slam. In Proc. of Robotics : Science and Systems (RSS).
- Cheng, Q. and Zhang, S. (2019). Efficiently handling scale variation for pedestrian detection. In International Conference on Intelligent Science and Big Data Engineering, pages 178–190. Springer.
- Cho, B.-S., Moon, W.-s., Seo, W.-J., and Baek, K.-R. (2011). A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding. Journal of mechanical science and technology, 25(11) :2907–2917.
- Cho, Y., Kim, G., and Kim, A. (2019). Deeplo : Geometry-aware deep lidar odometry. arXiv preprint arXiv :1902.10562.
- Choi, K., Suhr, J. K., and Jung, H. G. (2020). In-lane localization and ego-lane identification method based on highway lane endpoints. Journal of Advanced Transportation, 2020.
- Chong, K. S. and Kleeman, L. (1997). Accurate odometry and error modelling for a mobile robot. In Proceedings of International Conference on Robotics and Automation, volume 4, pages 2783–2788. IEEE.
- Choudhary, S., Indelman, V., Christensen, H. I., and Dellaert, F. (2015). Information-based reduced landmark slam. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 4620–4627. IEEE.
- Cuadra-Troncoso, J., Rivas-Casado, A., Álvarez-Sánchez, J., de la Paz-López, F., and Obregón-Castellanos, D. (2015). Towards robot localization using bluetooth low energy beacons rssi measures. In International Work-Conference on the Interplay Between Natural and Artificial Computation, pages 222–231. Springer.
- CUI, S., GAO, J., ZHANG, J., and FAN, Z. (2016). Autonomous exploration approach of mobile robot based on visual fastslam. Pattern Recognition and Artificial Intelligence, 29(12) :4.
- Dao, N. X., You, B.-J., Oh, S.-R., and Hwangbo, M. (2003). Visual self-localization for indoor mobile robots using natural lines. In Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453), volume 2, pages 1252–1257. IEEE.
- Dasarathy, B. V. (1997). Sensor fusion potential exploitation-innovative architectures and illustrative applications. Proceedings of the IEEE, 85(1) :24–38.
- Datondji, R. E., Ragot, N., Nasri, Y., Khemmar, R., and Boutteau, R. (2015). Odométrie visuelle par vision omnidirectionnelle pour la navigation autonome d'une chaise roulante motorisée. In Journées francophones des jeunes chercheurs en vision par ordinateur, Amiens, France.
- de Almeida, J. P. L. S., Nakashima, R. T., Neves-Jr, F., and de Arruda, L. V. R. (2020). A global/local path planner for multi-robot systems with uncertain robot localization. Journal of Intelligent & Robotic Systems, pages 1–23.
- Delobel, L. (2018). Agrégation d'information pour la localisation d'un robot mobile sur une carte imparfaite. Theses, Université Clermont Auvergne.
- Delobel, L., Aufrère, R., Chapuis, R., Debain, C., and Chateau, T. (2017). Towards automated map updating for mobile robot localization. In 2017 IEEE Intelligent Vehicles Symposium (IV), pages 1342–1347. IEEE.
- Delobel, L., Aufrère, R., Debain, C., Chapuis, R., and Chateau, T. (2018). A real-time map refinement method using a multi-sensor localization framework. IEEE Transactions on Intelligent Transportation Systems, 20(5) :1644–1658.
- Delobel, L., Aynaud, C., Aufrère, R., Debain, C., Chapuis, R., Chateau, T., and Bernay-Angeletti, C. (2015). Robust localization using a top-down approach with several lidar sensors. In 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), pages 2371–2376. IEEE.
- Dempster, A. P. (2008). Upper and lower probabilities induced by a multivalued mapping. In Classic works of the Dempster-Shafer theory of belief functions, pages 57–72. Springer.
- DENCEUX, T. (1998). Application du modèle des croyances transférables en reconnaissance de formes. rn (A), 1 :2.

- Dinh, V. Q., Munir, F., Azam, S., Yow, K.-C., and Jeon, M. (2020). Transfer learning for vehicle detection using two cameras with different focal lengths. *Information Sciences*, 514 :71–87.
- Dos Santos, M. M., De Giacomo, G. G., Drews, P. L., and Botelho, S. S. (2019). Underwater sonar and aerial images data fusion for robot localization. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 578–583. IEEE.
- Dufourd, D. (2005). *Des cartes combinatoires pour la construction automatique de modèles d’environnement par un robot mobile*. PhD thesis, Toulouse, INPT.
- Durikovic, R. and Kunovsky, P. (2019). Processing of rgb and multi-spectral data for geographic information system needs. In *2019 Communication and Information Technologies (KIT)*, pages 1–8. IEEE.
- Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping : part i. *IEEE robotics & automation magazine*, 13(2) :99–110.
- El Badaoui El Najjar, M. (2003). *Localisation dynamique d’un véhicule sur une carte routière numérique pour l’assistance à la conduite*. PhD thesis, Compiègne.
- El Mokhtari, K. (2015). *Estimation circulaire multi-modèles appliquée au Map matching en environnement contraint*. Theses, Université du Littoral Côte d’Opale.
- El Mokhtari, K., Reboul, S., Choquel, J.-B., Amami, B., and Benjelloun, M. (2016). Indoor localization by particle map matching. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 812–817. IEEE.
- Encarnacion, I. V. and Tiglaio, N. M. C. (2016). Rtklib-based gps localization for multipath mitigation in its applications. In *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 1077–1082. IEEE.
- Faisal, M., Alsulaiman, M., Hedjar, R., Mathkour, H., Zuair, M., Altaheri, H., Zakariah, M., Bencherif, M. A., and Mekhtiche, M. A. (2016). Enhancement of mobile robot localization using extended kalman filter. *Advances in Mechanical Engineering*, 8(11) :1687814016680142.
- Foo, P. H. and Ng, G. W. (2013). High-level information fusion : An overview. *J. Adv. Inf. Fusion*, 8(1) :33–72.
- Fraundorfer, F. and Scaramuzza, D. (2012). Visual odometry : Part ii : Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2) :78–90.
- Fremont, V. (2009). *Odométrie 3D vision/lidar pour les véhicules intelligents*. In *Journées Nationales de la Recherche en Robotique, JNRR’09*, Neuvy-sur-Barangeon, France.
- Frittella, S., Manoorkar, K., Palmigiano, A., Tzimoulis, A., and Wijnberg, N. (2020). Toward a dempster-shafer theory of concepts. *International Journal of Approximate Reasoning*, 125 :14–25.
- Frost, D., Prisacariu, V., and Murray, D. (2018). Recovering stable scale in monocular slam using object-supplemented bundle adjustment. *IEEE Transactions on Robotics*, 34(3) :736–747.
- Gao, L., Xiong, L., Xia, X., Lu, Y., and Yu, Z. (2019). Vehicle localization with vehicle dynamics during gnss outages. In *The IAVSD International Symposium on Dynamics of Vehicles on Roads and Tracks*, pages 1071–1079. Springer.
- Gao, R., Ye, F., Luo, G., and Cong, J. (2018). Indoor map construction via mobile crowdsensing. In *Smartphone-Based Indoor Map Construction*, pages 3–30. Springer.
- Gharib, S. S. M. and Esmaili, P. (2019). Anfis based ukf-slam path planning method. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–5. IEEE.
- Gong, Y., Meng, D., and Seibel, E. J. (2015). Bound constrained bundle adjustment for reliable 3d reconstruction. *Optics express*, 23(8) :10771–10785.
- Gong, Z., Ying, R., Wen, F., Qian, J., and Liu, P. (2019). Tightly coupled integration of gnss and vision slam using 10-dof optimization on manifold. *IEEE Sensors Journal*, 19(24) :12105–12117.
- Gonzalez, A., Devy, M., and Solà, J. (2011). *SLAM visuel monoculaire par caméra infrarouge*. In *ORASIS - Congrès des jeunes chercheurs en vision par ordinateur*, Praz-sur-Arly, France. INRIA Grenoble Rhône-Alpes.
- Grant, W. S., Voorhies, R. C., and Itti, L. (2019). Efficient velodyne slam with point and plane features. *Autonomous Robots*, 43(5) :1207–1224.
- Greenfeld, J. S. (2002). Matching gps observations to locations on a digital map. In *81th annual meeting of the transportation research board*, volume 1, pages 164–173. Washington, DC.

- Gruyer, D., Belaroussi, R., Vigneron, V., and Cord, A. (2014). How to manage conflict and ambiguities in localization and map matching. *Journal of Intelligent Systems*, 23(2) :171–182.
- Guechi, E.-H. (2010). Trajectory tracking of a non-holonomic mobile robot : an approach with Takagi-Sugeno fuzzy model and taking into account of output delays. Theses, Université de Valenciennes et du Hainaut-Cambresis.
- Gupta, A. K. and Shanker, U. (2020). Study of fuzzy logic and particle swarm methods in map matching algorithm. *SN Applied Sciences*, 2(4) :1–17.
- Guyonneau, D. (2013). Contribution à la détermination de surfaces conjuguées pour la transmission de puissance. PhD thesis, Aix-Marseille.
- Han-hai, S., Fa-jiang, H., Shu-wen, D., and Kang-le, W. (2017). Research on slam algorithm of iterated extended kalman filtering for multi-sensor fusion. In *Proceedings of the 3rd International Conference on Communication and Information Processing*, pages 242–246.
- Hao, G. and Sun, S. (2020). Distributed fusion filter for nonlinear multi-sensor systems with correlated noises. *IEEE Access*, 8 :39548–39560.
- He, M., Zhu, C., Huang, Q., Ren, B., and Liu, J. (2020). A review of monocular visual odometry. *The Visual Computer*, 36(5) :1053–1065.
- Hemant, K., Subhash, S., KRISHNA, K. M., and Pandey, A. K. (2008). Localizing from multi-hypotheses states minimizing expected path lengths for mobile robots. In *Advances In Mobile Robotics*, pages 79–86. World Scientific.
- Herrero-Pérez, D., Martínez-Barberá, H., LeBlanc, K., and Saffiotti, A. (2010). Fuzzy uncertainty modeling for grid based localization of mobile robots. *International Journal of Approximate Reasoning*, 51(8) :912–932.
- Holder, M., Hellwig, S., and Winner, H. (2019). Real-time pose graph slam based on radar. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1145–1151. IEEE.
- Hong, D., Heo, G., Myung, C.-W., and Ra, W.-S. (2017). Data association approach to mobile robot localization using non-unique landmarks. In *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, pages 1049–1053. IEEE.
- Hong, X., Chang, N., Wang, W., and Yin, Q. (2018). Weak interference detection improvement in gps multipath environment. In *2018 14th IEEE International Conference on Signal Processing (ICSP)*, pages 64–69. IEEE.
- Hou, Y., Feng, Z., and Xu, T. (2020). Decision making of mobile robot based on multimodal fusion. In *Proceedings of 2020 the 6th International Conference on Computing and Data Engineering*, pages 243–246.
- Ibraheem, M. (2010). Gyroscope-enhanced dead reckoning localization system for an intelligent walker. In *2010 International Conference on Information, Networking and Automation (ICINA)*, volume 1, pages V1–67. IEEE.
- Isobe, Y., Masuyama, G., and Umeda, K. (2018). Occlusion handling for a target-tracking robot with a stereo camera. *Robomech Journal*, 5(1) :1–13.
- Ito, S., Endres, F., Kuderer, M., Tipaldi, G. D., Stachniss, C., and Burgard, W. (2014). W-rgb-d : floor-plan-based indoor global localization using a depth camera and wifi. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 417–422. IEEE.
- Jensfelt, P. and Kristensen, S. (2001). Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5) :748–760.
- Jha, H., Lodhi, V., and Chakravarty, D. (2019). Object detection and identification using vision and radar data fusion system for ground-based navigation. In *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 590–593. IEEE.
- Jia, Z., Guangchang, D., Feng, C., Xiaodan, X., Chengming, Q., and Lin, L. (2019). A deep learning fusion recognition method based on sar image data. *Procedia computer science*, 147 :533–541.
- Jiang, R., Yang, S., Ge, S. S., Liu, X., Wang, H., and Lee, T. H. (2018). Gps/odometry/map fusion for vehicle positioning using potential function. *Autonomous Robots*, 42(1) :99–110.
- Jin, Z., Shao, Y., So, M., Sable, C., Shlayan, N., and Luchtenburg, D. M. (2019). A multisensor data fusion approach for simultaneous localization and mapping. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1317–1322. IEEE.
- Jo, K., Kim, C., and Sunwoo, M. (2018). Simultaneous localization and map change update for the high definition map-based autonomous driving car. *Sensors*, 18(9) :3145.

- Jones, E. S. and Soatto, S. (2011). Visual-inertial navigation, mapping and localization : A scalable real-time causal approach. The International Journal of Robotics Research, 30(4) :407–430.
- Joukhadar, A., Hanna, D. K., Müller, A., and Stöger, C. (2019). Ukf-assisted slam for 4wddmr localization and mapping. In Mechanism, Machine, Robotics and Mechatronics Sciences, pages 259–270. Springer.
- Julier, S. J. and Uhlmann, J. K. (1997). New extension of the kalman filter to nonlinear systems. In Signal processing, sensor fusion, and target recognition VI, volume 3068, pages 182–193. International Society for Optics and Photonics.
- Karabegović, I. and Dolecek, V. (2017). Mobile robotics. In Detecting and Mitigating Robotic Cyber Security Risks, pages 232–260. IGI Global.
- Kasmi, A., Denis, D., Aufrere, R., and Chapuis, R. (2019). Probabilistic framework for ego-lane determination. In 2019 IEEE Intelligent Vehicles Symposium (IV), pages 1746–1752. IEEE.
- Kassas, Z. Z. M., Maaref, M., Morales, J. J., Khalife, J. J., and Shamei, K. (2020). Robust vehicular localization and map matching in urban environments through imu, gnss, and cellular signals. IEEE Intelligent Transportation Systems Magazine, 12(3) :36–52.
- Kaviani, S., O'Brien, M., Van Brummelen, J., Najjaran, H., and Michelson, D. (2016). Ins/gps localization for reliable cooperative driving. In 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pages 1–4. IEEE.
- Keatmanee, C., Jakborvornphan, S., Potiwanna, C., San-Uml, W., and Dailey, M. N. (2018). Vision-based lane keeping-a survey. In 2018 International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES), pages 1–6. IEEE.
- Kenmogne, I.-F. (2019). Localisation ensembliste de drones à l'aide de méthodes par intervalles. PhD thesis, Université de Rennes 1.
- Khairuddin, A. R., Talib, M. S., and Haron, H. (2015). Review on simultaneous localization and mapping (slam). In 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pages 85–90. IEEE.
- Khorshidi, A., Shahri, A. M., and Oskoei, M. A. (2016). Evolutionary particle filter applied to leader-labor multi-robot localization for communication failure and kidnapped situations. In 2016 4th International Conference on Robotics and Mechatronics (ICROM), pages 619–624. IEEE.
- Khosyi'in, M., Prasetyowati, S. A. D., Nawawi, Z., and Suprpto, B. Y. (2019). Review and design of gps-rfid localization for autonomous vehicle navigation. In Proceedings of the 2019 2nd International Conference on Electronics and Electrical Engineering Technology, pages 42–46.
- Kim, T. and Choi, Y. (2017). Direct lidar odometry for a rotating multi-beam lidar. In ISOFIG 2017At : Gyeongju, Korea.
- Kirtay, M., Albanese, U., Vannucci, L., Schillaci, G., Laschi, C., and Falotico, E. (2020). The icub multisensor datasets for robot and computer vision applications. arXiv preprint arXiv :2003.01994.
- Knyaz, V. (2019). Multimodal data fusion for object recognition. In Multimodal Sensing : Technologies and Applications, volume 11059, page 110590P. International Society for Optics and Photonics.
- Kristensen, S. and Jensfelt, P. (2003). An experimental comparison of localisation methods, the mhl sessions. In Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453), volume 1, pages 992–997. IEEE.
- Kubelka, V., Oswald, L., Pomerleau, F., Colas, F., Svoboda, T., and Reinstein, M. (2015). Robust data fusion of multimodal sensory information for mobile robots. Journal of Field Robotics, 32(4) :447–473.
- Kulkarni, K., Turaga, P., Srivastava, A., and Chellappa, R. (2019). Pattern Recognition, pages 1–15. American Cancer Society.
- Kyriakides, I. (2019). Cognitive fusion for target tracking. Synthesis Lectures on Algorithms and Software in Engineering, 10(1) :1–65.
- Ladhari, M., Chapuis, R., Aufrère, R., Debain, C., and Malaterre, L. (2019). Parsimonious vehicle localization architecture using a generic top-down fusion process. In 2019 22th International Conference on Information Fusion (FUSION), pages 1–8. IEEE.
- Lamalle, A. (2012). Évaluation de performance d'un système de fusion d'informations : application au traitement d'images 3D. PhD thesis, Grenoble.
- Laneurit, J. (2006). Perception multisensorielle pour la localisation d'un robot mobile en environnement extérieur, application aux véhicules routiers. Theses, Université Blaise Pascal - Clermont-Ferrand II.
- Larnaout, D. (2014). Localisation d'un véhicule à l'aide d'un SLAM visuel contraint. Theses, Université Blaise Pascal - Clermont-Ferrand II.

- Laudy, C. and Museux, N. (2019). How to evaluate high level fusion algorithms ? In 2019 22th International Conference on Information Fusion (FUSION), pages 1–8. IEEE.
- Le Barz, C. (2015). Navigation visuelle pour les missions autonomes des petits drones. Theses, Université Pierre et Marie Curie - Paris VI.
- Lee, S., Tewolde, G. S., Lim, J., and Kwon, J. (2015). Vision based localization for multiple mobile robots using low-cost vision sensor. In 2015 IEEE International Conference on Electro/Information Technology (EIT), pages 280–285. IEEE.
- Lei, X., Feng, B., Wang, G., Liu, W., and Yang, Y. (2020). A novel fastslam framework based on 2d lidar for autonomous mobile robot. Electronics, 9(4) :695.
- Leonard, J. J., Durrant-Whyte, H. F., and Cox, I. J. (1992). Dynamic map building for an autonomous mobile robot. The International Journal of Robotics Research, 11(4) :286–298.
- Lerasle, F. (2008). Perception pour la robotique mobile en environnement humain. Habilitation à diriger des recherches, Université Paul Sabatier - Toulouse III.
- Lerma, C. D. C. (2011). Efficient simultaneous localisation and mapping in large and complex environments. PhD thesis, Universidad de Zaragoza.
- Li, Q., Chen, S., Wang, C., Li, X., Wen, C., Cheng, M., and Li, J. (2019). Lo-net : Deep real-time lidar odometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8473–8482.
- Li, Z. and Wang, N. (2020). Dmlo : Deep matching lidar odometry. arXiv, pages arXiv–2004.
- Liang, S., Cao, Z., Guan, P., Wang, C., Yu, J., and Wang, S. (2020). A novel sparse geometric 3-d lidar odometry approach. IEEE Systems Journal.
- Lin, M., Yang, C., Li, D., and Zhou, G. (2019). Intelligent filter-based slam for mobile robots with improved localization performance. IEEE Access, 7 :113284–113297.
- Liu, H., Chen, M., Zhang, G., Bao, H., and Bao, Y. (2018). Ice-ba : Incremental, consistent and efficient bundle adjustment for visual-inertial slam. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1974–1982.
- Liu, J., Danait, N., Hu, S., and Sengupta, S. (2013). A leave-one-feature-out wrapper method for feature selection in data classification. In 2013 6th International Conference on Biomedical Engineering and Informatics, pages 656–660. IEEE.
- Liu, L., Liu, D., Zhang, Y., and Peng, Y. (2016). Effective sensor selection and data anomaly detection for condition monitoring of aircraft engines. Sensors, 16(5) :623.
- Liu, Z., Shi, Z., and Xu, W. (2010). Multi-component information-equalized extended strong tracking filter for global localization : A scheme robust to kidnapping and symmetrical environments. Robotics and Autonomous Systems, 58(5) :465–487.
- Loghmani, M. R., Planamente, M., Caputo, B., and Vincze, M. (2019). Recurrent convolutional fusion for rgb-d object recognition. IEEE Robotics and Automation Letters, 4(3) :2878–2885.
- Lu, X., Wang, H., Tang, S., Huang, H., and Li, C. (2020). Dm-slam : Monocular slam in dynamic environments. Applied Sciences, 10 :4252.
- Luo, R. C. and Hsiao, T. J. (2018). Kidnapping and re-localizing solutions for autonomous service robotics. In IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, pages 2552–2557. IEEE.
- Ma, L. (2015). A map matching algorithm based on fuzzy logic. In Applied Mechanics and Materials, volume 722, pages 316–320. Trans Tech Publ.
- Mandel, C. and Birbach, O. (2013). Localization in urban environments by matching sensor data to map information. In 2013 European Conference on Mobile Robots, pages 50–55. IEEE.
- Marasini, D., Quatto, P., and Ripamonti, E. (2016). Fuzzy theory : Applications and criticism. Sistemi Intelligenti, 28 :319–342.
- Martin, A. (2005). La fusion d’informations. Polycopié de cours ENSIETA-Réf, 1484 :117.
- Martins, R. (2017). Odométrie visuelle directe et cartographie dense de grands environnements à base d’images panoramiques RGB-D. PhD thesis, Paris Sciences et Lettres.
- Matsumoto, T., Takahashi, T., Iwahashi, M., Kimura, T., Salbiah, S., and Mokhtar, N. (2011). Visual compensation in localization of a robot on a ceiling map. Scientific Research and Essays, 6(1) :131–135.

- Mehranian, P., Moghadam, A. A., and Basir, O. A. (2018). A dempster-shafer sensor fusion approach for traffic incident detection and localization. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pages 3911–3916. IEEE.
- Melbouci, K. (2017). Contributions au RGBD-SLAM. PhD thesis, Clermont Auvergne.
- Melbouci, k., Naudet Collette, S., Gay-Bellile, V., Ait Aider, O., Carrier, M., and Dhome, M. (2015). Ajustement de faisceaux du SLAM revisité en utilisant un capteur RGB-D. In Journées francophones des jeunes chercheurs en vision par ordinateur, Amiens, France.
- Mennillo, L. (2019). Reconstruction 3D de l'environnement dynamique d'un véhicule à l'aide d'un système multi-caméras hétérogène en stéréo. Theses, Université Clermont Auvergne.
- Mennillo, L., Royer, E., MONDOT, F., MOUSAIN, J., and Dhome, M. (2017). Multibody reconstruction of the dynamic scene surrounding a vehicle using a wide baseline and multifocal stereo system. In Workshop on Planning, Perception and Navigation for Intelligent Vehicles (satellite event of IROS'17), Vancouver, Canada.
- Milner, C. and Pervan, B. (2020). Bounding fault probabilities for advanced raim. IEEE Transactions on Aerospace and Electronic Systems.
- Milstein, A., Sánchez, J. N., and Williamson, E. T. (2002). Robust global localization using clustered particle filtering. In AAAI/IAAI, pages 581–586.
- Miyagusuku, R., Yamashita, A., and Asama, H. (2016). Improving gaussian processes based mapping of wireless signals using path loss models. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4610–4615. IEEE.
- Mohan, C. (2010). Fuzzy set theory and fuzzy logic. United Kingdom : Anshan Ltd.
- Moreau, J., Ambellouis, S., and Ruichek, Y. (2017). Fisheye-based method for gps localization improvement in unknown semi-obstructed areas. Sensors, 17(1) :119.
- Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., and Sayd, P. (2006). 3d reconstruction of complex structures with bundle adjustment : an incremental approach. In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., pages 3055–3061. IEEE.
- Murabito, F., Spampinato, C., Palazzo, S., Giordano, D., Pogorelov, K., and Riegler, M. (2018). Top-down saliency detection driven by visual classification. Computer Vision and Image Understanding, 172 :67–76.
- Nam, D. V. and Gon-Woo, K. (2020). Robust stereo visual inertial navigation system based on multi-stage outlier removal in dynamic environments. Sensors, 20(10) :2922.
- Nassreddine, G., Abdallah, F., and Dencœur, T. (2008). Map matching algorithm using belief function theory. In 2008 11th International Conference on Information Fusion, pages 1–8. IEEE.
- Nattharith, P. (2010). Mobile robot navigation using a behavioural control strategy. PhD thesis, University of Newcastle upon Tyne.
- Nemoto, T., Onodera, K., Mohan, R. E., Iwase, M., and Wood, K. (2018). An application of the simultaneous localization and mapping (slam) method based on the unscented kalman filter (ukf) to a reconfigurable quadruped robot with crawling locomotion. In 2018 International Conference on Reconfigurable Mechanisms and Robots (ReMAR), pages 1–8. IEEE.
- Nguyen, L. V., Kodagoda, S., Ranasinghe, R., and Dissanayake, G. (2015). Information-driven adaptive sampling strategy for mobile robotic wireless sensor network. IEEE Transactions on Control Systems Technology, 24(1) :372–379.
- Niewola, A. and Podseǳkowski, L. (2020). Psd–probabilistic algorithm for mobile robot 6d localization without natural and artificial landmarks based on 2.5 d map and a new type of laser scanner in gps-denied scenarios. Mechatronics, 65 :102308.
- Osman, M., Hussein, A., and Al-Kaff, A. (2019). Intelligent vehicles localization approaches between estimation and information : A review. In 2019 IEEE International Conference of Vehicular Electronics and Safety (ICVES), pages 1–8. IEEE.
- Özcan, A. H. and Ünsalan, C. (2020). Probabilistic object detection and shape extraction in remote sensing data. Computer Vision and Image Understanding, page 102953.
- Padhy, R. P., Xia, F., Choudhury, S. K., Sa, P. K., and Bakshi, S. (2018). Monocular vision aided autonomous uav navigation in indoor corridor environments. IEEE Transactions on Sustainable Computing, 4(1) :96–108.
- Paz, L. M., Piniés, P., Tardós, J. D., and Neira, J. (2008). Large-scale 6-dof slam with stereo-in-hand. IEEE transactions on robotics, 24(5) :946–957.
- Peng, Z., Li, Y., and Hao, G. (2020). The research on distributed fusion estimation based on machine learning. IEEE Access, 8 :38174–38184.

- Pierre, C., Chapuis, R., Aufrère, R., Laneurit, J., and Debain, C. (2018a). Range-only based cooperative localization for mobile robots. In 2018 21st International Conference on Information Fusion (FUSION), pages 1933–1939. IEEE.
- Pierre, C., Chapuis, R., Aufrère, R., Laneurit, J., and Debain, C. (2018b). Range-only based cooperative localization for mobile robots. In 2018 21st International Conference on Information Fusion (FUSION), pages 1933–1939. IEEE.
- Pollard, E. and Gingras, D. (2012). Improved low cost gps localization by using communicative vehicles. In 2012 12th International Conference on Control Automation Robotics & Vision (ICARCV), pages 317–322. IEEE.
- Qayum, A. and Razzaq, A. (2019). The impact of features on feature location. In 2019 International Conference on Frontiers of Information Technology (FIT), pages 1–15. IEEE.
- Qiu, C., Zhu, X., and Zhao, X. (2012). Vision-based unscented fastslam for mobile robot. In Proceedings of the 10th World Congress on Intelligent Control and Automation, pages 3758–3763.
- Qu, X., Soheilian, B., and Paparoditis, N. (2015). Vehicle localization using mono-camera and geo-referenced traffic signs. In 2015 IEEE Intelligent Vehicles Symposium (IV), pages 605–610.
- Quddus, M. A. (2006). High integrity map matching algorithms for advanced transport telematics applications. PhD thesis, Imperial College London London.
- Quddus, M. A., Ochieng, W. Y., and Noland, R. B. (2007). Current map-matching algorithms for transport applications : State-of-the-art and future research directions. Transportation research part c : Emerging technologies, 15(5) :312–328.
- Quddus, M. A., Ochieng, W. Y., Zhao, L., and Noland, R. B. (2003). A general map matching algorithm for transport telematics applications. GPS solutions, 7(3) :157–167.
- Rekleitis, I. M. (2004). A particle filter tutorial for mobile robot localization. Centre for Intelligent Machines, McGill University, 3480.
- Reynaud, R. and Bouaziz, S. (2005). Multisensors systems architecture. TRAITEMENT DU SIGNAL, 22(4) :393.
- Rives, P. and Devy, M. (2001). Perception pour la localisation. In Laumond, J.-P., editor, La robotique mobile. Hermes.
- Roheda, S., Krim, H., Luo, Z.-Q., and Wu, T. (2019). Event driven fusion. arXiv preprint arXiv :1904.11520.
- Ruan, W., Sheng, Q. Z., Yao, L., Li, X., Falkner, N. J., and Yang, L. (2018). Device-free human localization and tracking with uhf passive rfid tags : A data-driven approach. Journal of Network and Computer Applications, 104 :78–96.
- Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In Proceedings third international conference on 3-D digital imaging and modeling, pages 145–152. IEEE.
- Sabatini, R. (2016). Multisensor systems and data fusion for unmanned aircraft navigation and tracking. Emerging Sensing Technologies Summit 2016 (ESTS 2016)At : Melbourne (Australia).
- Sabry, M., Al-Kaff, A., Hussein, A., and Abdennadher, S. (2019). Ground vehicle monocular visual odometry. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pages 3587–3592. IEEE.
- Saman, A. B. S. H. and Lotfy, A. H. (2016). An implementation of slam with extended kalman filter. In 2016 6th International Conference on Intelligent and Advanced Systems (ICIAS), pages 1–4. IEEE.
- Scaramuzza, D. and Fraundorfer, F. (2011). Visual odometry [tutorial]. IEEE robotics & automation magazine, 18(4) :80–92.
- Scharwächter, T. and Franke, U. (2015). Low-level fusion of color, texture and depth for robust road scene understanding. In 2015 IEEE Intelligent Vehicles Symposium (IV), pages 599–604. IEEE.
- Schmitt, M. and Zhu, X. X. (2016). Data fusion and remote sensing : An ever-growing relationship. IEEE Geoscience and Remote Sensing Magazine, 4(4) :6–23.
- Schult, N., Reineking, T., Kluss, T., and Zetsche, C. (2015). Information-driven active audio-visual source localization. PloS one, 10(9) :e0137057.
- Schwartzing, W., Alonso-Mora, J., and Rus, D. (2018). Planning and decision-making for autonomous vehicles. Annual Review of Control, Robotics, and Autonomous Systems.
- Semenas, R. and Bausys, R. (2018). Autonomous navigation in the robots" local space by multi criteria decision making. In 2018 Open Conference of Electrical, Electronic and Information Sciences (eStream), pages 1–6. IEEE.
- Sendrier, N. (2007). Introduction à la théorie de l'information. Cours théorie de l'information École polytechnique.

- Seow, Y., Miyagusuku, R., Yamashita, A., and Asama, H. (2017). Detecting and solving the kidnapped robot problem using laser range finder and wifi signal. In 2017 IEEE international conference on real-time computing and robotics (RCAR), pages 303–308. IEEE.
- Shafer, G. (1976). A mathematical theory of evidence, volume 42. Princeton university press.
- Shamsudin, A. U., Ohno, K., Hamada, R., Kojima, S., Westfechtel, T., Suzuki, T., Okada, Y., Tadokoro, S., Fujita, J., and Amano, H. (2018). Consistent map building in petrochemical complexes for firefighter robots using slam based on gps and lidar. Robomech Journal, 5(1) :1–13.
- Shannon, C. E. (1948). A mathematical theory of communication. The Bell system technical journal, 27(3) :379–423.
- Shariatmadar, Z. S. and Faez, K. (2019). Visual saliency detection via integrating bottom-up and top-down information. Optik, 178 :1195–1207.
- Slowak, P. and Kaniewski, P. (2020). Lidar-based slam implementation using kalman filter. In Radioelectronic Systems Conference 2019, volume 11442, page 114420N. International Society for Optics and Photonics.
- Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In Autonomous robot vehicles, pages 167–193. Springer.
- Snidaro, L. and Visentini, I. (2016). Context for dynamic and multi-level fusion. In Context-Enhanced Information Fusion, pages 431–451. Springer.
- Somawirata, I. K., Widodo, K. A., and Achmadi, S. (2019). Road detection for supporting autonomous guided electric vehicle robot. In Proceedings of the 2019 4th International Conference on Robotics, Control and Automation, pages 111–115.
- Sun, L., Zhao, J., He, X., and Ye, C. (2018). Dlo : Direct lidar odometry for 2.5 d outdoor environment. In 2018 IEEE Intelligent Vehicles Symposium (IV), pages 1–5. IEEE.
- Swart, I., Irwin, B. V., and Grobler, M. M. (2019). Adaptation of the jdl model for multi-sensor national cyber security data fusion. In National Security : Breakthroughs in Research and Practice, pages 92–107. IGI Global.
- Thomas, Q. M., Wasenmüller, O., and Stricker, D. (2019). Delio : Decoupled lidar odometry. In 2019 IEEE Intelligent Vehicles Symposium (IV), pages 1549–1556. IEEE.
- Thrun, S. (2002a). Particle filters in robotics. In Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, pages 511–518. Morgan Kaufmann Publishers Inc.
- Thrun, S. (2002b). Probabilistic robotics. Communications of the ACM, 45(3) :52–57.
- Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust monte carlo localization for mobile robots. Artificial intelligence, 128(1-2) :99–141.
- Tsukamoto, K. and Uehara, K. (2012). A map matching method with fuzzy inference. ITEIS, 132(2) :334–335.
- Vallicrosa, G., Ridao, P., Ribas, D., and Palomer, A. (2014). Active range-only beacon localization for auv homing. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2286–2291. IEEE.
- Vasquez, B. P. E. A., Gonzalez, R., Matia, F., and De La Puente, P. (2018). Sensor fusion for tour-guide robot localization. IEEE Access, 6 :78947–78964.
- Verentsov, S., Magerramov, E., Vinogradov, V., Gizatullin, R., Alekseenko, A., Kholodov, Y., and Nikolskiy, E. (2017). Bayesian localization for autonomous vehicle using sensor fusion and traffic signs. In Proceedings of the 2017 International Conference on Robotics and Artificial Intelligence, pages 71–74.
- Wang, H., Zhou, Y., Yu, F., Zhao, L., Wang, C., and Ren, Y. (2019a). Fusional recognition for depressive tendency with multi-modal feature. IEEE Access, 7 :38702–38713.
- Wang, J. (2020). Design and research of intelligent mobile robot based on iot information fusion. In IOP Conference Series : Earth and Environmental Science, volume 470, page 012003. IOP Publishing.
- Wang, J., Song, J., Zhao, L., and Huang, S. (2018). A submap joining based rgb-d slam algorithm using planes as features. In Field and Service Robotics, pages 367–382. Springer.
- Wang, V. and Hayes, M. (2017). Sift localisation accuracy on interpolated speckle images. In 2017 International Conference on Image and Vision Computing New Zealand (IVCNZ), pages 1–6. IEEE.
- Wang, Y., Ding, L., Zhang, Y., and Luo, K. (2019b). Brain-computer fusion artificial intelligence system based on transfer learning. Journal of Computational Methods in Sciences and Engineering, 19(S1) :247–252.

- Wang, Z., Zhu, S., Li, Y., and Zhang, X. (2017). Mobile robot localization algorithm by combining visual odometry with natural landmarks. Transactions of the Chinese Society of Agricultural Engineering, 33(10) :70–77.
- Wassila Leila, R., Noureddine, B., and Belbachir, A. (2015). Les méthodes de localisation gps. In Workshop LAAR 2015 Physique du Rayonnement et ses Applications At université des Sciences et de la Technologie d’Oran.
- Wei, S., Na, L., Lijuan, S., Shulin, S., and Xiangpeng, L. (2012). Two improved methods of sift algorithm combined with harris. In 2012 24Th chinese control and decision conference (CCDC), pages 3251–3254. IEEE.
- White, C. E., Bernstein, D., and Kornhauser, A. L. (2000). Some map matching algorithms for personal navigation assistants. Transportation research part c : emerging technologies, 8(1-6) :91–108.
- White, F. (1991). Data fusion lexicon : data fusion subpanel of the joint directors of laboratories technical panel for c3. IEEE Trans. : San Diego, CA, USA.
- Wu, T. and Ranganathan, A. (2013). Vehicle localization using road markings. In 2013 IEEE Intelligent Vehicles Symposium (IV), pages 1185–1190. IEEE.
- Xu, B. and Chen, Z. (2018). Multi-level fusion based 3d object detection from monocular images. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2345–2353.
- Yan, R. J., Choi, Y. S., Wu, J., and Han, C. S. (2016). Data association of robot localization and mapping using partial compatibility test. Journal of the Korean Society for Precision Engineering, 33(2) :129–138.
- Yang, P., Chen, A., Gao, H., Liu, H., and Wen, X. (2020). A new information code for robot localization. In Journal of Physics : Conference Series, volume 1518, page 012004. IOP Publishing.
- Yashar, B. S. (2016). Multi-Sensor Map Matching Techniques for Autonomous Land Vehicle Navigation. PhD thesis, University of Calgary, Calgary, AB.
- Yin, J. and Yin, F. (2018). Indoor localization based on bluetooth. Artif. Intell. Research, 7(2) :87.
- Yokoya, N., Chanussot, J., and Iwasaki, A. (2012). Hyperspectral and multispectral data fusion based on nonlinear unmixing. In 2012 4th Workshop on Hyperspectral Image and Signal Processing : Evolution in Remote Sensing (WHISPERS), pages 1–4. IEEE.
- Youssef, A. A. and El-Sheimy, N. (2019). Particle imaging velocimetry gyroscope. Sensors, 19(21) :4734.
- Yousuf, S. and Kadri, M. B. (2016). Sensor fusion of ins, odometer and gps for robot localization. In 2016 IEEE Conference on Systems, Process and Control (ICSPC), pages 118–123. IEEE.
- Yozevitch, R. and Ben-Moshe, B. (2019). Gns and robot localization. Autonomous Mobile Robots and Multi-Robot Systems : Motion-Planning, Communication, and Swarming, page 109.
- Yun, J. and Miura, J. (2007). Multi-hypothesis outdoor localization using multiple visual features with a rough map. In Proceedings 2007 IEEE International Conference on Robotics and Automation, pages 3526–3532. IEEE.
- Zair, S. (2016). Détection de données aberrantes appliquée à la localisation GPS. Theses, Université Paris Saclay (COMUE).
- Zhang, E. and Masoud, N. (2020). Increasing gps localization accuracy with reinforcement learning. IEEE Transactions on Intelligent Transportation Systems.
- Zhang, J., Kaess, M., and Singh, S. (2014). Real-time depth enhanced monocular odometry. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4973–4980. IEEE.
- Zhang, J., Ma, Z., and Nuermaiti, N. (2019a). A pedestrian detection model based on binocular information fusion. In 2019 28th Wireless and Optical Communications Conference (WOCC), pages 1–5. IEEE.
- Zhang, J. and Singh, S. (2015). Visual-lidar odometry and mapping : Low-drift, robust, and fast. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 2174–2181. IEEE.
- Zhang, J. and Wang, X. (2020). Light field salient object detection via hybrid priors. In International Conference on Multimedia Modeling, pages 361–372. Springer.
- Zhang, K., Liu, S., Dong, Y., Wang, D., Zhang, Y., and Miao, L. (2017). Vehicle positioning system with multi-hypothesis map matching and robust feedback. IET Intelligent Transport Systems, 11(10) :649–658.
- Zhang, S., Dong, P., and Jing, Z. (2019b). Adaptive cubature kalman filtering slam algorithm based on variational bayes. Journal of Harbin Institute of Technology, 51(4) :12–18.

- Zhang, Y., Chen, B., and Yu, L. (2020a). Fusion estimation under binary sensors. Automatica, 115 :108861.
- Zhang, Z., Dong, P., Tuo, H., Liu, G., and Jia, H. (2020b). Robust variational bayesian adaptive cubature kalman filtering algorithm for simultaneous localization and mapping with heavy-tailed noise. Journal of Shanghai Jiaotong University (Science), 25(1) :76–87.
- Zhao, R., Pang, M., and Zhang, Y. (2018). Robust shape extraction for automatically segmenting raw lidar data of outdoor scenes. International Journal of Remote Sensing, 39(23) :9181–9205.
- Zhou, Q., Jin, D., Zeng, L., and Zhou, Y. (2008). Area concentric beacons localization for wireless sensor networks. In 2008 IEEE Wireless Communications and Networking Conference, pages 2129–2134. IEEE.
- Zhou, X., Su, Z., Huang, D., Zhang, H., Cheng, T., and Wu, J. (2018). Robust global localization by using global visual features and range finders data. In 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), pages 218–223. IEEE.
- Zong, C., Ji, Z., Yu, Y., and Shi, H. (2020). Research on obstacle avoidance method for mobile robot based on multisensor information fusion. Sensors and Materials, 32(4) :1159–1170.

# Webographie

- cyberneticzoo (2016). 1960 – stanford cart – (american). <http://cyberneticzoo.com/cyberneticanimals/1960-stanford-cart-american/>. page consultée le 23/03/2020.
- Goulu, D. (2008). Le chariot robot de léonard. <https://www.drgoulu.com/2008/08/02/598/#.X4WocJrgqCg>. page consultée le 16/04/2020.
- Innowtech (2019). Les algorithmes slam (simultaneous localization and mapping). <http://www.innowtech.com/2019/05/16/les-algorithmes-slam/>. page consultée le 18/04/2020.
- kidioui (2015). La tesla model s en pilote automatique. <https://voiture.kidioui.fr/blog/actualite/la-tesla-model-s-en-pilote-automatique.html>. page consultée le 07/02/2020.
- wikipedia (2020a). General atomics mq-1 predator. [https://fr.wikipedia.org/wiki/General\\_Atomics\\_MQ-1\\_Predator](https://fr.wikipedia.org/wiki/General_Atomics_MQ-1_Predator). page consultée le 23/07/2020.
- wikipedia (2020b). Whitehead torpedo. [https://en.wikipedia.org/wiki/Whitehead\\_torpedo](https://en.wikipedia.org/wiki/Whitehead_torpedo). page consultée le 10/05/2020.
- Wikipédia (2020). Amer (navigation maritime) — wikipédia, l'encyclopédie libre. [http://fr.wikipedia.org/w/index.php?title=Amer\\_\(navigation\\_maritime\)&oldid=168660175](http://fr.wikipedia.org/w/index.php?title=Amer_(navigation_maritime)&oldid=168660175). [page consultée le 22/05/2020].
- Zebra (2014). Que sont les solutions de localisation ? <https://www.zebra.com/fr/fr/solutions/intelligent-edge-solutions/rtls/what-are-location-solutions.html>. page consultée le 10/04/2020.