

**UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES**

**DÉVELOPPEMENT D'UNE ARCHITECTURE DE LOCALISATION ROBUSTE D'UN AGV DANS UN  
ENVIRONNEMENT D'INTÉRIEUR**

**MÉMOIRE PRÉSENTÉ  
COMME EXIGENCE PARTIELLE DE LA  
MAÎTRISE EN GÉNIE ÉLECTRIQUE**

**PAR  
LOTFI ZEGHMI**

**DÉCEMBRE 2020**

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

MAÎTRISE EN GÉNIE ÉLECTRIQUE (M. Sc. A.)

**Direction de recherche :**

---

Souso Kelouwani

Directeur de recherche

**Jury d'évaluation**

---

Souso Kelouwani

Évaluateur

---

Nadjet Zioui

Évaluatrice

---

Alben Cardenas

Évaluateur externe

## Résumé

Aujourd'hui, le défi principal de la navigation naturelle des véhicules de transport intra-usines, plus communément désigné sous l'acronyme AGV pour « Automated Guided Vehicles », est la capacité de s'auto-localiser dans tous types d'environnements, uniquement à partir des informations disponibles des capteurs embarqués, le plus souvent des encodeurs et des télémètres lasers. L'approche la plus populaire à ce jour qui permet de résoudre la plupart des problèmes de localisations connus, est basée sur l'estimation Bayésienne à travers les filtres particulaires (MCL). Ces derniers représentent la croyance postérieure par un ensemble d'échantillons pondérés. Cependant, l'architecture standard du MCL échoue souvent à générer des échantillons au bon endroit car elle utilise généralement une propagation d'échantillons basée sur un modèle de mouvement qui dépend des mesures bruitées des encodeurs. De ce fait, un modèle de mouvement alternatif -Iterative Closest Point- utilisant les mesures du télémètre laser est présenté et qui démontre une meilleure précision que le modèle d'odométrie mais qui échoue à générer des échantillons consistants dans des environnements symétriques. L'objectif principal de ce projet de maîtrise est de développer une nouvelle architecture de localisation robuste basée sur MCL dans tout type d'environnements. Pour cela, nous proposons un nouveau modèle de mouvement qui s'appuie sur un filtre de Kalman Étendu (EKF). Le design du filtre EKF est conçu de telle sorte à générer des échantillons de manière efficace en intégrant une solution adaptative d'estimation de la covariance d'observation.

Une implémentation de la nouvelle architecture de localisation est effectuée sur la plateforme de développement robotique Robotic Operating System (ROS). Des tests dans un

environnement type, généré par le simulateur Gazebo, ont montrés une meilleure précision ( $\approx 4\%$ ) de la solution proposée par rapport aux architectures de localisation standards. Par ailleurs, une validation expérimentale de la robustesse de notre architecture est menée sur un AGV industriel de la compagnie DIVEL Inc. dans des infrastructures réelles.

Pour conclure, l'utilisation d'un filtre de Kalman Étendu (EKF) pour générer une nouvelle proposition de distribution dans le cadre d'un filtre particulaire (MCL) permet d'améliorer les performances de la localisation des AGV dans tous types d'environnements d'intérieurs.

## Remerciements

C'est avec un grand plaisir que nous réservons ces quelques lignes en signe de gratitude et de reconnaissance à toutes celles et ceux qui ont contribué de près et de loin à la réussite de mes travaux de recherche.

Je voudrais tout d'abord remercier mon directeur de recherche, M. Sousso Kelouwani, professeur au département de génie mécanique à l'UQTR, pour son soutien, sa confiance et ses conseils qui ont été nombreux et indispensables dans la réussite de mes travaux.

Un grand merci à mon collègue et ami, M. Ali Amamou, chercheur postdoctoral à l'Institut de Recherche sur l'Hydrogène, pour avoir contribué de manière significative par sa disponibilité et ses précieux commentaires lors de nombreuses discussions.

Je désire remercier mes collègues du laboratoire IRH pour leurs supports et les multiples discussions, qui étaient toujours pertinents et incitant à la recherche.

Enfin, je tiens à remercier du fond de mon cœur mes parents et mon frère, grâce à eux, j'ai réussi.

# Table des matières

Résumé.....	iii
Remerciements.....	v
Table des matières.....	vi
Liste des tableaux.....	x
Liste des abréviations.....	xiii
Chapitre 1 - Introduction.....	1
1.1 Contexte général.....	1
1.2 Types d'AGV.....	2
1.2.1 Véhicules à charge unitaire :.....	2
1.2.2 Chariots élévateurs autonomes.....	3
1.2.3 Remorqueurs.....	4
1.3 Navigation des AGV.....	4
1.3.1 Navigation à chemin fixe.....	5
1.3.2 Navigation à chemin ouvert.....	6
1.4 Problématique de recherche.....	9
1.5 Objectifs de recherche.....	11
1.6 Méthodologie.....	12
Chapitre 2 - État de l'art.....	13

2.1	Problème de localisation.....	13
2.2	Représentation de l'environnement.....	14
2.2.1	Approche métrique.....	15
2.2.2	Approche topologique.....	15
2.3	Types de capteurs.....	16
2.3.1	Capteurs proprioceptifs.....	16
2.3.2	Capteurs extéroceptifs.....	18
2.4	Approche probabiliste du problème de localisation.....	20
2.4.1	Croyance.....	21
2.4.2	Estimation récursive Bayésienne.....	23
2.5	Méthodes d'implémentation.....	27
2.5.1	Filtre de Kalman.....	27
2.5.2	Filtre Particulaire (MCL).....	31
2.6	Conclusion.....	33
Chapitre 3 - Architecture de localisation proposée.....		34
3.1	Présentation des limites de l'architecture MCL classique.....	34
3.1.1	Modèle de mouvement.....	34
3.1.2	Calcul du modèle de mouvement avec ICP.....	38
3.2	Design d'un filtre de Kalman Étendu (EKF).....	41

3.2.1	Modèle du système EKF.....	42
3.2.2	Modèle de mesure EKF .....	43
3.3	Estimation de la matrice de covariance d'observation.....	44
3.3.1	Extraction des caractéristiques géométriques de l'environnement.....	45
3.3.2	Calcul de la covariance.....	46
3.4	Algorithme d'un filtre hybride Kalman-MCL.....	48
3.5	Conclusion.....	50
Chapitre 4 - Implémentation et validation des résultats.....		51
4.1	Implémentation.....	51
4.1.1	ROS.....	51
4.1.2	Description du robot .....	53
4.1.3	Représentation de l'environnement .....	55
4.2	Résultats de simulation.....	56
4.3	Validation expérimentale.....	62
4.3.1	Validation au centre d'essai de l'IRH.....	63
4.3.1	Validation à l'usine de DIVEL .....	64
4.4	Conclusion.....	65
Chapitre 5 - Conclusion .....		66
5.1	Conclusion et discussions.....	66

5.2 Perspectives .....	67
Références (Bibliographie) .....	69

## Liste des tableaux

Tableau 1- Algorithme général du filtre de Bayes .....	26
Tableau 2- Algorithme général du Filtre de Kalman .....	29
Tableau 3- Différence entre le Filtre de Kalman et le Filtre de Kalman Étendu .....	30
Tableau 4- Algorithme général du filtre particulaire (MCL) .....	32
Tableau 5- Algorithme Kalman-MCL .....	50
Tableau 6- Spécifications techniques de l'AGV .....	54
Tableau 7- Covariances calculées dans les zones (a) et (b) .....	59
Tableau 8- Erreur quadratique moyenne des différentes approches de localisation.....	62
Tableau 9- Paramètres et résultats de l'expérimentation à l'IRH .....	64
Tableau 10- Paramètres et résultats de l'expérimentation à l'usine DIVEL.....	64

# Liste des Figures

Figure 1- Véhicule à charge unitaire [5] .....	3
Figure 2- Chariot élévateur [6].....	3
Figure 3 - AGV remorqueur [7].....	4
Figure 4- Guidage par ligne inductive [11].....	5
Figure 5 - Guidage par bandes magnétique [11].....	6
Figure 6 - Guidage par point magnétique [12].....	7
Figure 7 - Guidage par GPS d'intérieur [13] .....	8
Figure 8 - Guidage par laser avec réflecteurs [12].....	9
Figure 9 - Formalisation du problème de localisation [22].....	13
Figure 10 - Différentes représentation d'un environnement réel [24].....	14
Figure 11 - Grille d'occupation [25].....	15
Figure 12 - Carte topologique avec des nœuds colorés qui représentent des lieux spécifiques [26] .....	16
Figure 13 - Principe de fonctionnement d'un codeur optique [28].....	17
Figure 14 - Principe de fonctionnement du capteur à ultrason [30].....	18
Figure 15 - Principe de fonctionnement d'un télémètre laser [32].....	19
Figure 16- Illustration des trois mouvements de base du modèle d'odométrie .....	35
Figure 17 – Propagation des particules (points rouges) utilisant le modèle d'odométrie. La commande et la position réelle du robot sont respectivement représentés par un trait bleu et une flèche noire.....	37
Figure 18- Propagation des particules (points rouges) utilisant le modèle d'ICP. La commande et la position réelle du robot sont respectivement représentés par un trait bleu et une flèche noire .....	40
Figure 19- Propagation des particules (points rouges) utilisant le modèle d'ICP dans un long couloir. La commande et la position réelle du robot	

<p>sont respectivement représentés par un trait bleu et une flèche noire.....</p>	41
Figure 20- Schéma bloc simplifié du filtre EKF pour la fusion multisensorielle .....	42
Figure 21- Représentation d'une ligne utilisant un point normal .....	45
Figure 22- Changement de position entre deux balayages consécutifs.....	47
Figure 23- Schéma de fonctionnement du master sous ROS.....	52
Figure 24- Plateforme mobile de DIVEL et son modèle 3D .....	53
Figure 25- Représentation de l'environnement modéliser sur Gazebo. (a) représente une vue du dessus. (b) représente une perspective alternative de l'environnement ainsi que les mesures du télémètre simulées .....	55
Figure 26- Carte de l'environnement sous forme d'une grille d'occupation. Les espaces occupés, libres et inconnus sont respectivement noirs, blancs et gris .....	56
Figure 27- Schéma simplifié de l'architecture ROS de la structure de localisation.....	57
Figure 28- Représentation de lignes extraites (en rouge) dans deux zones différentes (a) et (b) à partir des mesures du télémètre (en bleu).....	58
Figure 29- Trajectoire de référence du robot .....	60
Figure 30- Trajectoires estimées par les différentes approches .....	61
Figure 31- Centre d'essai de l'IRH .....	63
Figure 32- Usine de DIVEL Inc.....	63

## Liste des abréviations

AGV	Automated Guided Vehicle
EKF	Extended Kalman Filter
ICP	Iterative Closest Point
IRH	Institut de Recherche sur l'Hydrogène
KF	Kalman Filter
Lidar	Light detection and ranging
LMS	Least Mean Square
MCL	Monte Carlo localization
MSE	Mean Square error
GPS	Global Positioning System
ROS	Robotic Operating System

# Chapitre 1 - Introduction

## 1.1 Contexte général

Depuis la révolution industrielle du XVIIIe siècle, les humains ont commencé à remettre aux machines les travaux considérés comme étant fastidieux et physiquement exigeant. Grâce à ce processus d'automatisation continu, il y a eu une amélioration du bien-être humain et une augmentation du niveau de vie, ainsi, dans l'industrie d'aujourd'hui, l'automatisation a permis de rendre accessible une large gamme de produits et de commodités tout en réduisant la quantité de travail humain nécessaire.

L'automatisation industrielle traditionnelle nécessite une phase d'installation longue et coûteuse pour les machines et n'est donc viable généralement que pour les processus de fabrication rigides à haut débit. Un exemple classique est celui des lignes de production de châssis d'automobiles. Ces lignes de production ne sont économiques que lorsqu'elles produisent un grand nombre de véhicules identiques. Le but de l'automatisation flexible, en revanche, est de devenir économiquement rentable tout en étant également destiné à des lots plus petits et à des processus changeants. Par conséquent, une automatisation flexible nécessite de faibles coûts d'adaptabilité. [1]

L'automatisation industrielle flexible est donc considérée comme une technologie clé dans le cadre des défis mondiaux actuels. C'est dans ce contexte que la robotique va être largement utilisée, en particulier pour des tâches telles que le soudage, l'emballage, et la peinture. Tous ces systèmes robotiques se présentent sous la forme de bras manipulateurs qui

effectuent des mouvements répétitifs, néanmoins, pour le transport à grande échelle, ces systèmes robotiques ne sont pas particulièrement pratiques et, par conséquent, des véhicules auto-guidés (AGV) ont dû être développés [1]. Ces derniers sont définis par [2] comme étant des systèmes de transport de matériels sans conducteur et utilisés en industrie dans des sites de productions ou des entrepôts d'usine. De nos jours, les AGV sont de plus en plus utilisés car ils permettent d'améliorer l'efficacité de la manutention de matériel et ainsi d'accroître les gains de productivité [3].

## **1.2 Types d'AGV**

Comme les AGV sont utilisés dans de nombreux domaines d'application, ils peuvent être très différents et avoir divers attributs. Selon [4], la meilleure façon de classer les AGV est de regarder les charges qu'ils transportent. Nous pouvons dès lors définir trois principales catégories d'AGV : les véhicules à charge unitaire, les chariots élévateurs et les remorqueurs.

### *1.2.1 Véhicules à charge unitaire :*

Il s'agit de véhicules motorisés capables de transporter un seul produit (une bobine, un moteur) ou une palette ou un bac contenant des produits [4]. Ces véhicules passent généralement sous un chariot à roulettes ou un wagon de matériel et le soulèvent légèrement afin de le transporter. Notons que ce type d'AGV possède de nombreux avantages tel que le fait qu'ils nécessitent moins d'espace que les autres AGV car le conteneur lui-même détermine presque entièrement l'espace requis [4]. Il a également une grande maniabilité lors du chargement et du déchargement.



Figure 1- Véhicule à charge unitaire [5]

### 1.2.2 *Chariots élévateurs autonomes*

Les chariots élévateurs autonomes sont inspirés des chariots manuels classiques, leurs principales charges sont des palettes ou des conteneurs compatibles avec les chariots élévateurs. Ils sont idéaux pour les tâches fastidieuses et exigeantes en main-d'œuvre, mais qui ne nécessitent pas beaucoup de compétences [4]. Les chariots élévateurs automatisés sont bien adaptés pour les longs trajets, le transbordement et les opérations de rangement dans des entrepôts [4].



Figure 2- Chariot élévateur [6]

### 1.2.3 Remorqueurs

Ce sont des véhicules motorisés capables de tirer derrière eux un ou plusieurs véhicules non motorisés avec des charges, à la façon d'un train. La principale application de ces AGV remorqueurs est la fourniture de composants aux lignes d'assemblage et de production. Chaque remorqueur peut être composé de plusieurs chariots, ce qui lui confère une plus grande capacité de transport que les véhicules à charge unitaires ou les chariots élévateurs [4].



Figure 3 - AGV remorqueur [7]

## 1.3 Navigation des AGV

Un AGV navigue entre différents emplacements dans sa zone d'opération pour accomplir ses différentes tâches. Pour ce faire, il doit connaître sa propre position dans son système de coordination fixe qui se fait par différentes méthodes de navigation [8]. Les méthodes de navigation pour les AGV peuvent être séparées en deux groupes différents; navigation à chemin fixe et navigation à chemin ouvert [9].

### 1.3.1 Navigation à chemin fixe

Avec une navigation fixe, l'AGV est guidé à l'aide de directives physiques sur ou sous le sol. Cela peut être effectué par des lignes inductives ou par des bandes magnétiques. Pour la navigation fixe, les chemins sont prédéfinis et la navigation est facile car seul un capteur est nécessaire pour détecter la ligne directrice. Les modifications du chemin nécessitent des changements physiques et le système doit être arrêté [9]. Les emplacements spécifiques dans l'entrepôt ou l'usine sont également physiques et nécessitent des marquages spéciaux qui contiennent des informations supplémentaires pour l'AGV. Les marquages peuvent être des bandes métalliques, des aimants, des bandes adhésives ou des marquages colorés où l'AGV est programmé pour savoir quoi faire lorsque le marquage est reconnu [9].

- Guidage par lignes inductives : Cela consiste à encastrer un fil métallique sur le chemin de conduite du véhicule et à charger un courant basse fréquence et basse tension sur le fil métallique pour générer un champ magnétique [10]. Deux bobines sont montées sous l'AGV qui mesurent l'écart par rapport au fil, ce qui permet à l'AGV de suivre le chemin.



Figure 4- Guidage par ligne inductive [11]

- Guidage par bandes magnétique : Cette technique est similaire au guidage par lignes inductives, mais au lieu d'avoir des fils métalliques enfoui dans le sol, on utilise des bandes à même le sol afin de définir le chemin que doit suivre l'AGV. Le procédé de guidage est réalisé par l'identification du signal de champ magnétique par le capteur électromagnétique embarqué [4].

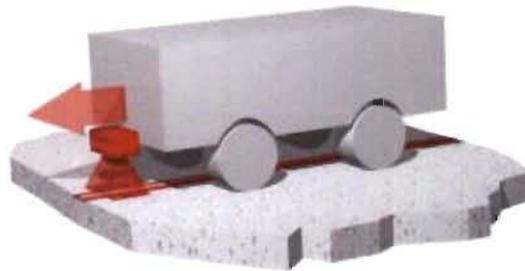


Figure 5 - Guidage par bandes magnétique [11]

### 1.3.2 Navigation à chemin ouvert

La navigation à chemin ouvert, également appelée navigation libre [4], est une méthode de navigation sans chemin de guidage physique. En d'autres termes, l'itinéraire que doit emprunter l'AGV n'est pas physiquement fixe, Et lorsque des changements sont nécessaires, l'AGV doit seulement être reprogrammés. Il en résulte qu'aucun travail manuel extensif n'est requis pour modifier les chemins de guidage [9]. Même si les AGV libres peuvent emprunter n'importe quel chemin, une feuille de route est nécessaire. Cette limitation à suivre certains chemins rend le système moins flexible mais il est nécessaire afin réduire la complexité du système [9].

- Guidage par point magnétique : La navigation par points magnétiques est une méthode qui utilise des points d'ancrage magnétiques dans le sol pour se repérer [4].

Les points d'ancrage sont des aimants permanents passifs qui sont percés dans le sol puis recouverts d'époxy, d'une couche de vinyle ou peints. L'AGV est capable de prendre des mesures avec un capteur magnétique et peut, de ce fait, localiser sa position absolue.



Figure 6 - Guidage par point magnétique [12]

- Guidage par GPS : Le système GPS est souvent utilisé dans la navigation autonome des véhicules automobiles et permet d'obtenir la position du véhicule à travers des satellites par triangulation. Ce système a été envisagé pour la navigation des AGV mais reste problématique dans des environnements d'intérieur tel que des entrepôts ou usines pour différentes raisons qu'on peut citer : - Une précision de l'ordre du mètre, ce qui est inacceptable pour localiser des AGV à taille humaine, - La nécessité d'avoir un ciel dégagé et clair tandis que le domaine d'opération des AGV est essentiellement en intérieur [4].

Pour toutes les raisons citées précédemment, des systèmes de GPS d'intérieur ont été développés, le principe de fonctionnement de ces derniers réside sur l'utilisation de balises radio comme émetteurs afin que le récepteur placé sur l'AGV puisse se localiser dans la zone d'opération du véhicule [8].

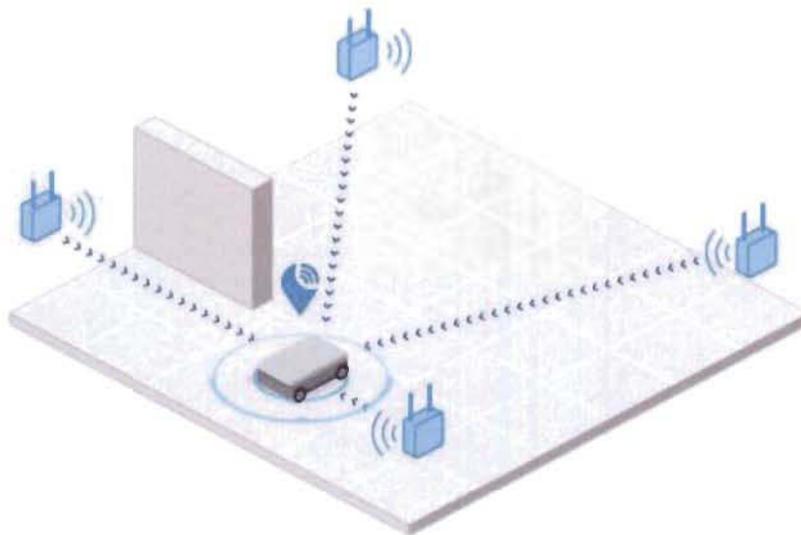


Figure 7 - Guidage par GPS d'intérieur [13]

- Guidage par laser avec réflecteurs : Pour ce type de navigation, l'AGV se positionne grâce à des repères artificiels. En effet, un laser rotatif est monté sur le dessus de l'AGV et il émet un faisceau lumineux qui va être réfléchi sur une plaque rétro réfléchissante (réflecteurs) qui est montée sur les murs et les piliers au-dessus de la tête des opérateurs [8]. Les coordonnées des points de repère sont ajoutées à la carte de la zone d'opération lors de la configuration du système. Lorsque les réflecteurs réfléchissent la lumière du laser, l'AGV est capable de trianguler sa position absolue en se basant sur la connaissance des coordonnées de chaque

réflecteur. Au moins deux ou trois points de repère doivent être visibles pour que l'AGV se positionne [4].

La navigation au laser est la méthode de navigation à chemin ouvert la plus importante, car elle est très précise et n'est pas limitée par les conditions du sol [4]. Les limites de cette méthode sont l'augmentation du prix de l'équipement et le fait que la conception de l'AGV est restreinte car le laser rotatif doit avoir une ligne de vue claire et ne peut pas être au même niveau que le personnel [14].

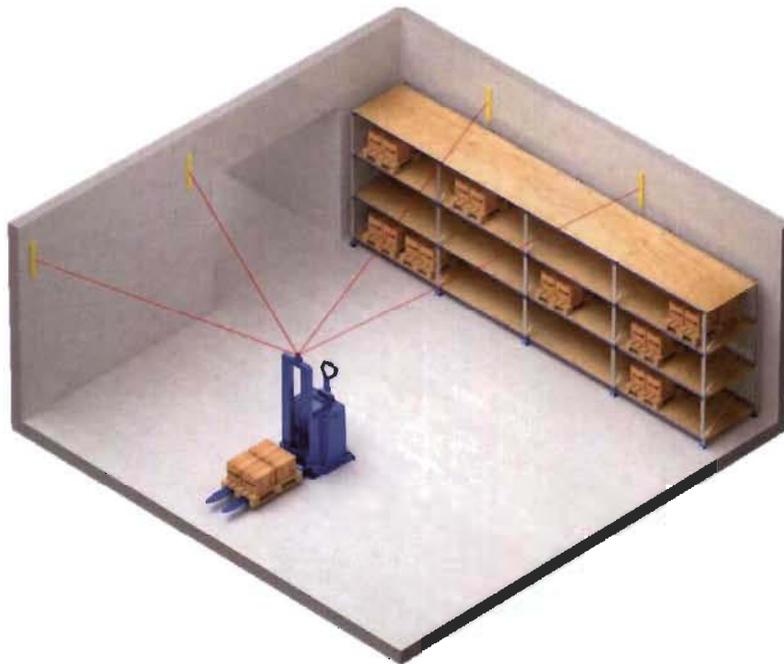


Figure 8 - Guidage par laser avec réflecteurs [12]

#### 1.4 Problématique de recherche

Au-delà des techniques de navigation classiques existantes en industrie, beaucoup de chercheurs se sont penchés sur ce qu'on peut appeler la navigation naturelle [4], [8] [15]. Cette dernière repose sur le fait que l'AGV -considéré comme un robot mobile- puisse se localiser qu'avec des capteurs déjà embarqués sur lui et cela sans aucune aide d'infrastructure

additionnelle dans l'environnement. Dans [15], l'auteur décrit la possibilité d'exploiter des lasers de sécurité dans un but de localisation puisque ces derniers sont déjà présents dans les AGV car obligatoire. Cela a pour avantage d'être plus économique et plus flexible que les autres types de navigation.

L'une des exigences les plus importantes dans la navigation naturelle d'un AGV est sa capacité à s'auto-localiser de manière robuste dans son environnement [16]. Dans ce contexte, le problème de localisation est souvent divisé en trois sous-problèmes [17]:

- Localisation locale : La position courante de l'AGV est localement évaluée en fonction de la position précédente. La difficulté de ce type de problème consiste à éviter la dérive du véhicule [18].
- Localisation globale : Ici la position initiale de l'AGV n'est pas connue, donc il doit se localiser de manière globale dans son environnement [18].
- Kidnapping: Dans ce cas, l'AGV effectue un déplacement imprévu qu'il n'est pas capable d'évaluer. Toute la difficulté de cette situation consiste à détecter ce kidnapping (identifier le fait que l'AGV ne soit plus où il croit être). Une fois le kidnapping identifié, on se retrouve dans un contexte de localisation globale [18].

Tous ces sous-problèmes peuvent être résolus à travers différentes approches qui sont présentées dans [17]. Il subsiste néanmoins un autre sous problème lié à la localisation dans des environnements symétriques que les solutions standards n'ont pas pu résoudre [19]. En effet, le manque de diversité perceptuelle dans les environnements symétriques provoque dans la plupart des solutions standards une convergence prématurées dans un espace sous optimal [20].

Les environnements symétriques doivent être considérés dans le problème de localisation des AGV car ces derniers opèrent dans des environnements d'intérieurs qui ont des caractéristiques géométriques fortement symétriques tel que les long couloirs, rangés de stockages similaires...etc. C'est dans cette optique que le problème général de localisation sera présenté dans les prochains chapitres ainsi que la solution proposée pour le cas des environnements symétriques.

### **1.5 Objectifs de recherche**

Toutes les méthodes standards proposées de localisation sont particulièrement moins fiables dans des environnements symétriques [20],[21], [22]. Le but du projet est d'élaborer une architecture de localisation robuste (répétabilité de la solution à long terme) et précise (se rapprocher au maximum de la position réelle) pour un AGV dans tous types d'environnement d'intérieur et exclusivement avec les capteurs embarqués suivants : encodeurs et télémètres lasers. Afin d'atteindre l'objectif principal mentionné précédemment, les sous-objectifs suivants doivent être traité :

- Compréhension du problème de localisation
- Exploration des solutions proposées de localisation
- Identification des limites des solutions de localisation
- Élaboration d'une architecture de localisation robuste capable de performer dans tous types d'environnements
- Validation de la nouvelle architecture de localisation

## 1.6 Méthodologie

Afin de parvenir à ces objectifs, il est nécessaire d'adopter une chronologie tout au long de ce projet. La méthodologie à suivre dans cette étude se répartie principalement en quatre chapitres :

- Chapitre 2 : État de l'art : Introduction au problème de localisation en robotique mobile terrestre et la description des différentes instances du problème ainsi que les méthodes standard proposées par la littérature.
- Chapitre 3 : Architecture de localisation proposée : Montrer les limites des architectures standards de localisation, plus particulièrement dans des environnements symétriques, et présenter la méthode proposée en détails dans ce chapitre ainsi que l'architecture finale de localisation qui va être utilisée.
- Chapitre 4 : Implémentation et validation des résultats : Ce chapitre fournira des informations détaillées sur l'implémentation de la méthode sur le middleware ROS (Robotic Operating System) ainsi que les résultats obtenus en simulation et en expérimentations dans des environnements d'intérieurs.
- Chapitre 5 : Conclusion et perspectives : On résume le travail effectué accompagné de conclusions et on présente des perspectives pour d'éventuels travaux futurs.

## Chapitre 2 - État de l'art

Comme les AGV sont considérés comme des robots mobiles, il est donc logique de porter notre attention vers les concepts initiés dans ce domaine. Ce chapitre traitera donc essentiellement de la problématique de localisation en robotique mobile terrestre ainsi que des différentes notions et outils qui l'entourent comme les approches classiques de localisation décrites dans la littérature scientifique.

### 2.1 Problème de localisation

Le problème de la localisation consiste à répondre à la question "Où suis-je ?" du point de vue d'un robot. Cela signifie que le robot doit trouver sa position par rapport à l'environnement en déterminant trois paramètres : deux coordonnées cartésiennes  $x$  et  $y$  pour la position et un angle  $\theta$  pour l'orientation. (Voir Figure 9)

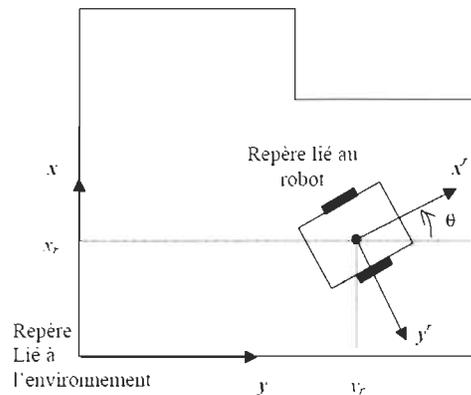


Figure 9 - Formalisation du problème de localisation [23]

Un robot doit avoir accès à deux types d'informations pour déterminer sa position [21]. La première est une information « a priori », qui est collectée par le robot lui-même ou fournie par une source externe dans une phase d'initialisation. En général, ce type

d'informations décrit l'environnement dans lequel le robot se déplace et est souvent représenté par une carte [21]. Le deuxième type d'information est, quant à lui, recueilli à partir des capteurs du robot et fournit des renseignements sur son déplacement lors de sa navigation dans l'environnement [21].

## 2.2 Représentation de l'environnement

Pour effectuer efficacement des missions complexes dans des environnements intérieurs, les robots mobiles autonomes doivent pouvoir acquérir et maintenir des modèles de leurs environnements. Les recherches dans ce domaine [24] ont produit deux approches fondamentales pour la modélisation des environnements de robots intérieurs : l'approche métrique et l'approche topologique.

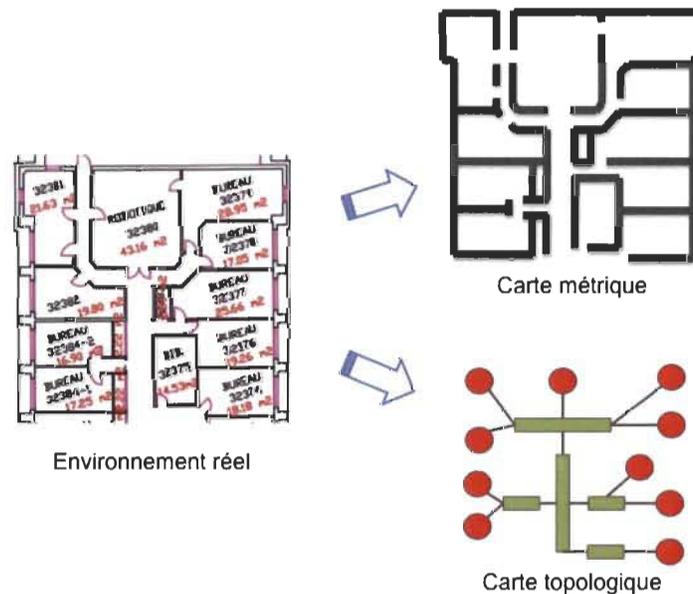


Figure 10 - Différentes représentation d'un environnement réel [25]

### 2.2.1 Approche métrique

Cette approche permet de représenter les environnements par des grilles uniformément espacées [24]. Chaque cellule de la grille peut, par exemple, indiquer la présence d'un obstacle dans la région correspondante de l'environnement [24]. Les cartes métriques considérées ici sont des grilles d'occupations bidimensionnelles discrètes, une valeur est attribuée à chaque cellule qui nous informe sur son état [24]. Plus précisément, la cellule contient la croyance quant à savoir si oui ou non le centre du robot peut être déplacé vers le centre de cette cellule (elle représente ainsi l'espace de configuration du robot projeté dans le plan  $\langle x, y \rangle$ ). Les valeurs d'occupation sont déterminées en fonction des mesures des capteurs [24]. (Voir Figure 11)

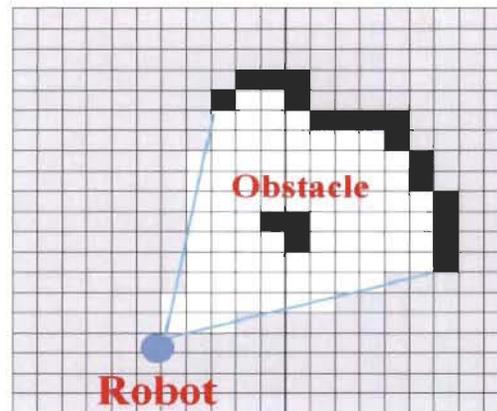


Figure 11 - Grille d'occupation [26]

### 2.2.2 Approche topologique

Dans une approche topologique, l'environnement est représenté par des graphes d'adjacence [24]. Les nœuds des graphes correspondent à des lieux ou des points de repère distincts (tels que des portes) et ils sont reliés par des arcs s'il existe un chemin direct entre eux [24].

L'un des avantages de cette représentation réside dans l'abstraction qui est faite sur la géométrie de l'environnement [18].

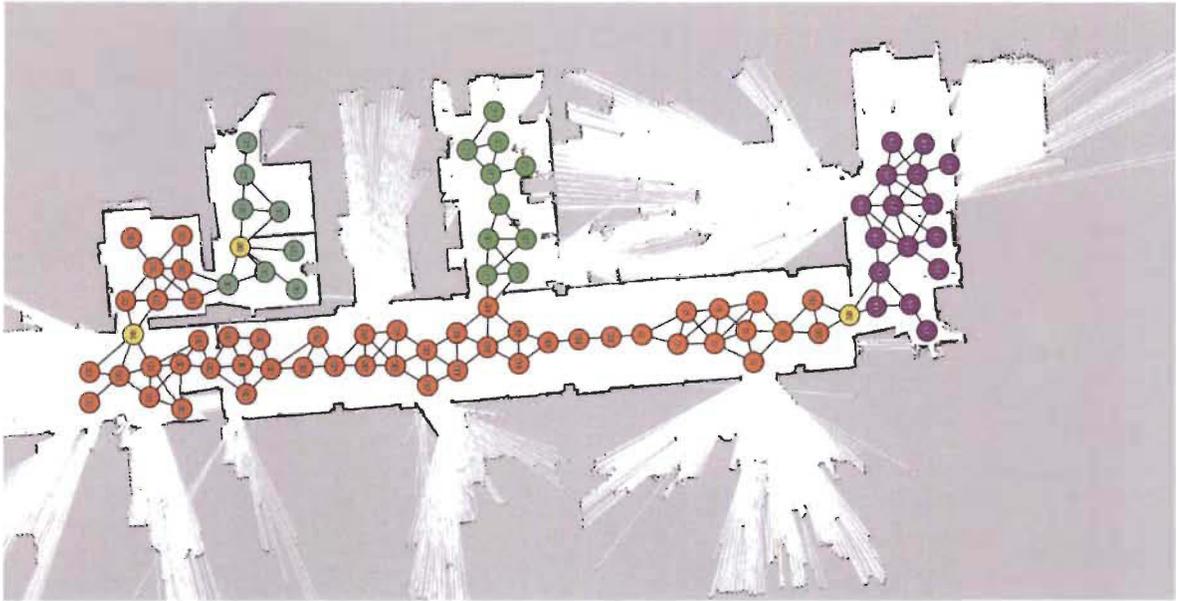


Figure 12 - Carte topologique avec des nœuds colorés qui représentent des lieux spécifiques [27]

### 2.3 Types de capteurs

Pour qu'un robot mobile puisse se localiser, il doit être en mesure de percevoir son environnement et d'évaluer son déplacement. Cela se fait en prenant des mesures à l'aide de divers capteurs qui peuvent être classés en deux grandes familles : les capteurs proprioceptifs et les capteurs extéroceptifs.

#### 2.3.1 Capteurs proprioceptifs

Les capteurs proprioceptifs sont des capteurs qui vont mesurer l'état interne du robot et permettent généralement de mesurer son déplacement [25].

- Encodeurs : C'est un dispositif très populaire utilisant généralement des codeurs optiques pour mesurer la vitesse angulaire et la position d'un moteur ou au niveau de l'arbre d'une roue ou d'un mécanisme de direction [28]. En robotique mobile terrestre, les encodeurs sont énormément utilisés pour contrôler la position ou la vitesse des roues et par conséquent, estimer le déplacement du robot dans son référentiel [25].

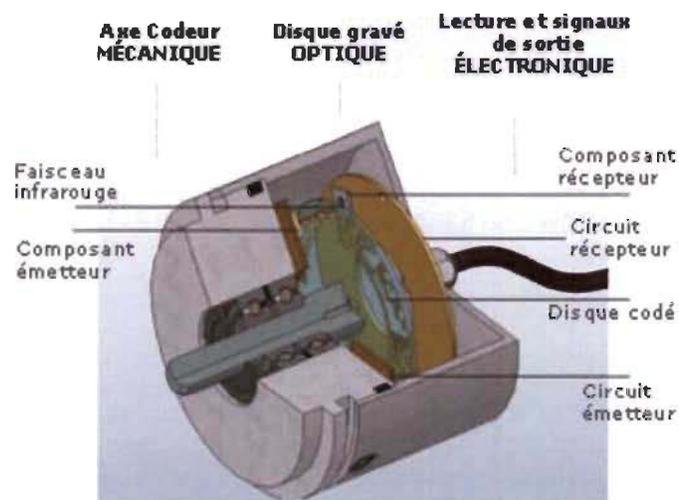


Figure 13 – Les différents éléments constituant un encodeur optique [29]

- Système inertiel : Ils sont utilisés pour déterminer l'orientation et l'inclinaison des robots. Ils nous permettent, avec les informations de vitesse appropriées, d'estimer aussi le déplacement du robot. Un système inertiel est généralement constitué de neuf capteurs : trois accéléromètres qui mesurent l'accélération linéaire, trois gyroscopes qui mesurent les vitesses angulaires et de trois magnétomètres qui permettent, par la mesure du champ magnétique terrestre, de déduire la direction du nord [25].

### 2.3.2 Capteurs extéroceptifs

Les capteurs extéroceptifs sont l'ensemble des capteurs qui perçoivent l'environnement dans lequel se déplace le robot et qui peuvent nous informer sur sa position à travers son environnement [25].

- Capteurs ultrasons : Un capteur à ultrasons est un instrument qui mesure la distance d'un objet à l'aide d'ondes sonores ultrasoniques. Il utilise un transducteur pour envoyer et recevoir des impulsions ultrasoniques qui relaient des informations sur la proximité d'un objet [30].

Le principe de fonctionnement de ce module est simple. Il envoie une impulsion ultrasonique à 40 kHz -inaudible à l'oreille humaine- qui se déplace dans l'air et s'il y a un obstacle ou un objet, il rebondira vers le capteur. En calculant le temps de trajet et la vitesse du son, la distance peut être calculée [31].(Voir Figure 14)

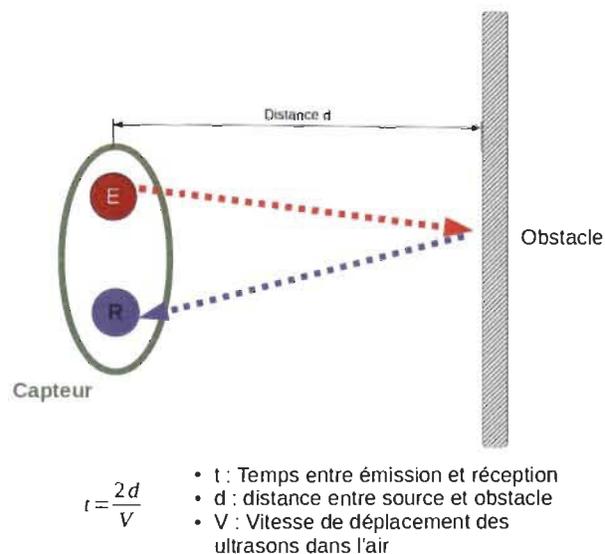


Figure 14 - Principe de fonctionnement du capteur à ultrason [31]

Toutefois, un des principaux inconvénients du capteur ultrason est sa sensibilité à la température et à la pression [30]. La vitesse de propagation des ondes sonores dépend des conditions environnantes [30]. Ils sont également sensibles à d'autres capteurs à proximité, et il peut y avoir une interférence entre ces derniers [31].

- Télémètre laser : Son principe de fonctionnement est sensiblement le même que celui du capteur à ultrason. En effet, Ce type de capteur se compose d'un émetteur qui éclaire une cible avec un faisceau collimaté (par exemple un laser) et d'un récepteur capable de détecter la composante de la lumière qui est essentiellement coaxiale avec le faisceau transmis [32]. Souvent appelés radar optique ou lidar (Light Detection And Ranging), ces appareils produisent une estimation de la distance basée sur le temps nécessaire à la lumière pour atteindre la cible et revenir. Un mécanisme de miroir pivotant est présent dans le télémètre pour balayer le faisceau lumineux afin de couvrir le plan d'une scène de l'environnement [25]. (Voir Figure 15)

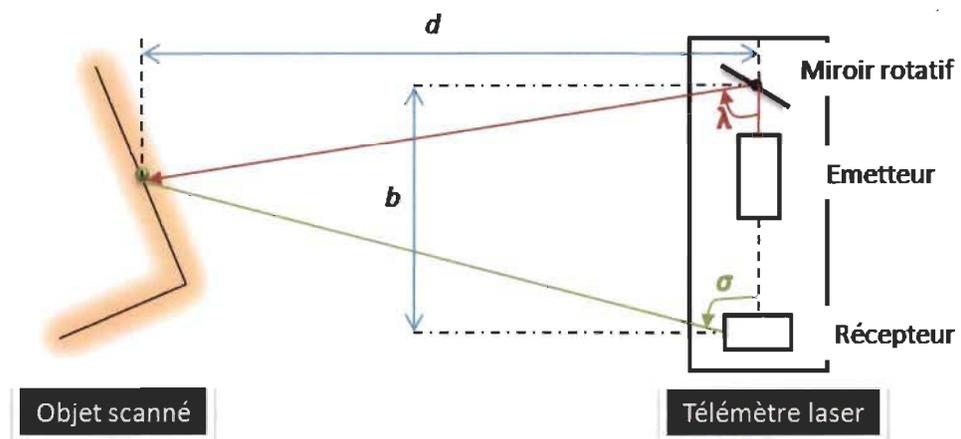


Figure 15 - Principe de fonctionnement d'un télémètre laser [33]

Le principal avantage des techniques de mesure de distance au laser par rapport aux appareils à ultrasons est que la lumière laser à une longueur d'onde beaucoup plus petite, ce qui permet d'envoyer un faisceau de sonde beaucoup plus concentré et donc d'obtenir une précision plus élevée [18]. À l'inverse, ils sont généralement chers et ne peuvent détecter ni les objets transparents (vitres...), ni les objets réfléchissants (miroirs...) [18].

- Caméras : La vision est considérée comme l'une des sources d'information les plus riches en robotique mobile terrestre, la quantité et qualité d'information fournie par les images issues de caméras permet au robot une perception quasi complète de son environnement [25]. Le volume conséquent de données fournies par ces capteurs est cependant difficile à traiter, c'est dans cette voie que les recherches se concentrent car cette dernière est prometteuse pour l'avenir de la robotique mobile terrestre [25].

Il existe une multitude de caméras qui peuvent être utilisées de différentes manières dans la localisation des robots mobiles [18]. L'une d'elles consiste à détecter des amers visuels -des points particuliers qui servent de repère, tels que des portes ou des affiches- à partir desquels il sera possible de calculer la position du robot [25].

Il existe aussi un autre principe qui se repose sur la mise en place de deux caméras observant le même environnement mais de différents points de vue, cela permet d'estimer la position relative des objets par rapport au robot [25].

#### **2.4 Approche probabiliste du problème de localisation**

Rappelons que le problème de localisation est de déterminer la position et l'orientation d'un robot mobile dans une carte de l'environnement et en utilisant les données disponibles des capteurs. L'approche la plus connue pour résoudre les problèmes immanents de la

localisation est une approche probabiliste, l'idée clé de cette approche est de représenter explicitement l'incertitude, en utilisant le calcul de la théorie des probabilités [17]. Autrement dit, au lieu de s'appuyer sur une seule « meilleure estimation », les algorithmes probabilistes représentent les informations par des distributions de probabilités sur tout un espace d'hypothèses possibles [17]. Ce faisant, ils peuvent représenter l'ambiguïté et le degré de croyance d'une manière mathématiquement solide, leur permettant de s'adapter à toutes les sources d'incertitude [17].

#### 2.4.1 Croyance

Si nous regardons le problème de localisation de manière probabiliste, nous pouvons dire que le robot a une croyance [17] sur l'endroit où il se trouve. À tout moment, il ne considère pas un emplacement possible, mais tout l'espace des emplacements. Le problème de localisation consiste donc à estimer la densité de probabilité sur l'espace de tous les emplacements [21].

Le robot a une croyance *Bel* qui est la densité de probabilité sur tous les emplacements  $x \in E$ , où  $E$  est l'ensemble de tous les emplacements. La localisation peut être vue comme le maintien de la croyance, qui est la probabilité que le robot se trouve à l'emplacement  $x_k$  à l'instant  $k$ , étant donné toutes les informations disponibles  $d_{0...k}$  jusqu'à cet instant [17].

$$Bel(x_k) = P(x_k | d_{0...k}) \quad (2.1)$$

Le but de la localisation est de rapprocher le plus possible cette croyance de l'emplacement réel du robot. Si le robot atteint cet objectif, il sait exactement où il se trouve [17].

Afin de mettre à jour la croyance avec les mesures accessibles des capteurs, nous devons exprimer les informations de mesure en termes probabilistes. En d'autres termes, nous devons définir le système stochastique dynamique du robot et son environnement.

### **Probabilité de transition d'état :**

Un robot effectue des actions et change sa position dans l'environnement. Nous définissons l'action  $u_k$  à partir d'un ensemble d'actions possibles et exprimons l'emplacement des changements du robot de manière probabiliste par une densité de transition [17].

$$P(x_k | x_{k-1}, u_k) \quad (2.2)$$

Cette densité de probabilité donne la probabilité que si à l'instant  $k - 1$  le robot était à l'emplacement  $x_{k-1}$  et a effectué l'action  $u_k$ , alors il s'est retrouvé à l'emplacement  $x_k$  à l'instant  $k$ . Autrement dit, la densité de transition décrit comment les actions du robot change son emplacement. Cette densité est donc appelée modèle d'action ou de mouvement [17].

### **Probabilité de perception :**

Nous pouvons également décrire la perception du robot en termes probabilistes. Soit  $Z$  l'espace de toutes les mesures possibles provenant d'un capteur, et que  $z_k$  désigne un élément de  $Z$  observé à l'instant  $k$ . Nous pouvons décrire la probabilité qu'un capteur observe  $z_k$  à partir d'un certain emplacement  $x_k$  à l'instant  $k$  par la densité [17].

$$P(z_k | x_k) \quad (2.3)$$

Contrairement à la densité de transition de l'action du robot, cette densité de probabilité est difficile à calculer [21]. La raison réside dans la dimensionnalité parfois élevée des mesures. Considérez par exemple la complexité de la densité de probabilité si les mesures

proviennent d'une caméra. La densité de probabilité devra donner une probabilité pour chaque image de caméra possible à chaque emplacement possible, ce qui nécessiterait une grande quantité de puissance de calcul.

#### 2.4.2 Estimation récursive Bayésienne

La méthode la plus utilisée pour calculer les croyances est l'algorithme du filtre de Bayes [17]. Cet algorithme calcule la distribution de croyances à partir des données de mesure et de contrôle de manière récursive.

Le filtre de Bayes prend comme entrée la distribution de croyance  $Bel(x_{k-1})$  à l'instant  $k - 1$ , ainsi que le contrôle le plus récent  $u_k$  et les mesures  $z_k$  les plus récentes. La sortie est la croyance  $Bel(x_k)$  à l'instant  $k$ . Les mesures des capteurs et les informations de contrôle peuvent être corrompues par des bruits causés par différentes sources tel que : l'environnement perceptible, la conception mécanique et/ou électrique du capteur...etc. Afin de faire face à ces incertitudes, le filtre de Bayes est réalisé en deux phases [17]: Phase de prévision et phase de correction.

#### **Phase de prévision :**

Supposons que le robot a effectué une action et souhaite inclure le résultat de la mesure de position relative de cette action dans sa croyance. La croyance du robot après avoir intégré l'action  $u_k$  exécuté à l'étape  $k$ , et avant qu'il n'obtienne une nouvelle mesure  $z_k$ , est la croyance a priori et est défini comme suit.

$$Bel^-(x_k) = P(x_k | z_1, u_1, z_2, u_2, \dots, z_{k-1}, u_k) \quad 2.4$$

Nous pouvons réécrire cette définition originale en utilisant le théorème de la probabilité totale et l'hypothèse de Markov [17]. Le théorème de la probabilité totale déclare que la

probabilité d'un résultat est égale à la somme des probabilités de chacun de ses résultats dépendants et partiels. En utilisant ce théorème, nous réécrivons la définition de la croyance apriori (2.4) de la sorte :

$$Bel^-(x_k) = \int_E P(x_k|x_{k-1}, z_1, u_1, \dots, z_{k-1}, u_k) \times P(x_{k-1}|z_1, u_1, \dots, z_{k-1}, u_k) dx \quad 2.5$$

Cette équation décrit que la croyance apriori d'un robot d'être dans l'état  $x_k$  est la somme des probabilités de passer de l'état  $x_{k-1}$  à l'état  $x_k$  compte tenu de toutes les actions et mesures antérieures,  $P(x_k|x_{k-1}, z_1, u_1, \dots, z_{k-1}, u_k)$ , multiplié par la probabilité d'être réellement dans l'état  $x_{k-1}$  étant donné toutes les mesures et actions antérieures [21],  $P(x_{k-1}|z_1, u_1, \dots, z_{k-1}, u_k)$ .

Sachant que l'emplacement physique du robot à l'instant  $k - 1$  ne dépend pas de l'action effectuée à l'instant  $k$  [17], [21], nous pouvons des lors éliminer le terme  $u_k$  de  $P(x_{k-1}|z_1, u_1, \dots, z_{k-1}, u_k)$  et par conséquent réécrire (2.5) en

$$Bel^-(x_k) = \int_E P(x_k|x_{k-1}, z_1, u_1, \dots, z_{k-1}, u_k) \times Bel^+(x_{k-1}) dx \quad 2.6$$

Avec  $Bel^+(x_{k-1}) = P(x_{k-1}|z_1, u_1, \dots, z_{k-1})$  qui représente la croyance apostériori de l'emplacement du robot à l'instant  $k - 1$ .

Pour simplifier l'expression du premier terme de l'intégrale dans (2.5), nous émettons une hypothèse de Markov [17], qui stipule que, étant donné la connaissance de l'état actuel, le passé est indépendant du futur, en d'autres termes, les mesures antérieures n'ont pas d'impact sur le calcul de la probabilité de distribution de la densité de transition actuel,

$$P(x_k|x_{k-1}, z_1, u_1, \dots, z_{k-1}, u_k) = P(x_k|x_{k-1}, u_k) \quad 2.7$$

La partie droite de cette équation est la probabilité conditionnelle d'être dans l'état  $x_k$  étant donné la connaissance de l'état précédent et de l'action effectuée. Nous l'avons défini comme le modèle d'action dans (2.2). En substituant le résultat dans (2.7), nous obtenons une équation qui peut être utilisée pour incorporer efficacement les actions du robot dans sa croyance,

$$Bel^-(x_k) = \int_E P(x_k|x_{k-1}, u_k) \times Bel^+(x_{k-1}) dx \quad 2.8$$

### Phase de correction :

Nous voulons incorporer le vecteur de la dernière mesure  $z_k$  dans la croyance a priori  $Bel^-(x_k)$  pour former la croyance postérieure  $Bel^+(x_k)$ . Avec la règle de Bayes et l'hypothèse de Markov, nous pouvons écrire cette croyance postérieure sous la forme suivante [17].

$$Bel^+(x_k) = \frac{P(z_k|x_k, z_1, u_1, \dots, z_{k-1}, u_k) \times P(x_k|z_1, u_1, \dots, z_{k-1}, u_k)}{P(z_k|z_1, u_1, \dots, z_{k-1}, u_k)} = \frac{P(z_k|x_k, z_1, u_1, \dots, z_{k-1}, u_k) \times Bel^-(x_k)}{P(z_k|z_1, u_1, \dots, z_{k-1}, u_k)} \quad 2.9$$

Pour rendre les calculs de l'équation (2.9) moins complexes, nous émettons à nouveau l'hypothèse de Markov. Dans ce cas, nous l'utilisons pour déclarer qu'une mesure issue d'un capteur n'est pas influencée par les emplacements précédents du robot, peu importe comment le robot est arrivé à l'emplacement actuel. En d'autres termes, la probabilité d'observer une mesure est indépendante des actions et observations effectuées avant l'arrivée du robot dans son état actuel. Nous utilisons cette hypothèse pour réécrire le premier terme dans le numérateur de (2.9),

$$Bel^+(x_k) = \frac{P(z_k|x_k)Bel^-(x_k)}{P(z_k|z_1, u_1, \dots, z_{k-1}, u_k)} \quad 2.10$$

L'équation (2.10) montre comment nous pouvons exprimer la croyance postérieure en termes de croyance antérieure et de modèle de capteurs.

Finalement, nous pouvons combiner les résultats dérivés en une seule équation de localisation en tenant compte des informations de détection et d'action. En substituant l'équation (2.8) à l'équation (2.10), la croyance postérieure devient,

$$\begin{aligned} Bel^+(x_k) &= \frac{P(z_k|x_k) \int_E P(x_k|x_{k-1}, u_k) \times Bel^+(x_{k-1}) dx}{P(z_k|z_1, u_1, \dots, z_{k-1}, u_k)} \\ &= \eta_k P(z_k|x_k) \int_E P(x_k|x_{k-1}, u_k) \times Bel^+(x_{k-1}) dx \end{aligned} \quad 2.11$$

Avec  $\eta_k$  est la densité de probabilité normalisée  $P(z_k|z_1, u_1, \dots, z_{k-1}, u_k)^{-1}$ .

---

### Algorithme 1 : Filtre de Bayes

---

Entrée	:	$Bel^+(x_{k-1}), u_k, z_k$
Étape 1	:	<i>Phase de prévision :</i> $Bel^-(x_k) = \int_E P(x_k x_{k-1}, u_k) \times Bel^+(x_{k-1}) dx$
Étape 2	:	<i>Phase de correction :</i> $Bel^+(x_k) = \eta_k P(z_k x_k) Bel^-(x_k)$
Sortie	:	$Bel^+(x_k)$

---

Tableau 1- Algorithme général du filtre de Bayes

## 2.5 Méthodes d'implémentation

Le filtre de Bayes est un concept abstrait dans la mesure où il ne fournit qu'un cadre probabiliste pour l'estimation d'un état récursif [17], [18], [34]. L'implémentation du filtre de Bayes nécessite de spécifier le modèle perceptuel  $P(z_k|x_k)$ , la dynamique  $P(x_k|x_{k-1}, u_k)$ , et la représentation de la croyance  $Bel^+(x_k)$  [34]. Les propriétés des différentes implémentations des filtres de Bayes diffèrent fortement dans la façon dont elles représentent les densités de probabilité sur l'état  $x_k$  [17], [34]. Nous allons discuter maintenant des différentes implémentations du filtre de Bayes dans le cadre de la localisation d'un robot mobile.

### 2.5.1 Filtre de Kalman

La technique la plus connue pour mettre en œuvre des filtres Bayésien est probablement le filtre de Kalman (KF) [17], [34]. Le filtre de Kalman a été inventé dans les années 1950 par Rudolph Emil Kalman, comme technique de filtrage et de prévision dans les systèmes linéaires.

Le filtre de Kalman représente les croyances par la représentation des moments [34] : À l'instant  $k$ , la croyance est représentée par une gaussienne avec une moyenne  $\mu_k$  et une covariance  $\Sigma_k$ ,

$$Bel^+(x_k) \approx N(x_k, \mu_k, \Sigma_k) = \det(2\pi\Sigma_k)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_k - \mu_k)^T \Sigma_k^{-1} (x_k - \mu_k)\right) \quad 2.13$$

L'estimation des états du systèmes par le filtre de Kalman est basée sur le modèle linéaire du processus [35] qui décrit la probabilité  $P(x_k|x_{k-1}, u_k)$

$$x_k = A_k x_{k-1} + B_k u_k + \varepsilon_k \quad 2.14$$

Ici,  $x_k$  et  $x_{k-1}$ , sont les vecteurs d'états, et  $u_k$  est le vecteur de control.  $A_k$  est une matrice carrée de dimension  $n \times n$ , ou  $n$  est la dimension du vecteur d'état  $x_k$ .  $B_k$  est une matrice de dimension  $n \times m$ , ou  $m$  est la dimension du vecteur de control  $u_k$ .

La variable aléatoire  $\varepsilon_k$  est un vecteur aléatoire gaussien qui modélise le caractère aléatoire de la transition d'état. Sa moyenne est nulle et sa covariance est notée  $R_k$ .

Concernant le modèle perceptuel  $P(z_k|x_k)$ , l'équation suivante décrit la relation linéaire entre les états du système et les mesures effectuées à l'instant  $k$  [35]

$$z_k = C_k x_k + \delta_k \quad 2.15$$

$C_k$  est défini en tant que matrice de dimension  $k \times n$ , ou  $k$  est la dimension du vecteur de mesure  $z_k$ . Le vecteur  $\delta_k$  quant à lui décrit le bruit de mesure et sa distribution est gaussienne avec une moyenne nulle et une covariance notée  $Q_k$ .

Le rôle du filtre de Kalman est de fournir une estimation de  $x_k$  à l'instant  $k$  sur deux étapes : une prévision et une mise à jour, étant donné l'estimation initiale, la série de mesures, et les informations du système décrit par  $A, B, C, Q$  et  $R$ . Bien que les matrices de covariance soient censées refléter les statistiques des bruits, la vraie statistique des bruits n'est pas réellement connue dans de nombreuses applications pratiques [35]. Par conséquent,  $Q$  et  $R$  sont généralement utilisés comme paramètres de réglage que l'utilisateur peut ajuster pour obtenir les performances souhaitées [35]. (L'algorithme général du filtre de Kalman est présenté dans le Tableau 2).

---

**Algorithme 2 : Filtre de Kalman**


---

Entrée	:	$\mu_{k-1}, \Sigma_{k-1}, u_k, z_k$
Étape 1	:	<i>Phase de prévision :</i> $\bar{\mu}_k = A_k \mu_{k-1} + B_k u_k$ $\bar{\Sigma}_k = A_k \Sigma_{k-1} A_k^T + Q$
Étape 2	:	<i>Phase de correction :</i> $K_k = \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + R)^{-1}$ $\mu_k = \bar{\mu}_k + K_k (z_k - C_k \bar{\mu}_k)$ $\Sigma_k = (I - K_k C_k) \bar{\Sigma}_k$
Sortie	:	$\mu_k, \Sigma_k$

---

Tableau 2- Algorithme général du Filtre de Kalman

Les hypothèses concernant la linéarité du modèle de processus et du modèle perceptuel sont rarement satisfaites en pratique [34], [35], [17]. Cette observation rend le filtre de Kalman inapplicable à presque tous les problèmes pratiques. C'est dans ce contexte qu'un filtre de Kalman étendu (EKF) a été proposé dans la littérature afin de surmonter l'hypothèse de linéarité.

Dans le cadre du EKF, le modèle du processus et le modèle perceptuel sont gouvernés par des fonctions non linéaires  $g$  et  $h$ , respectivement :

$$x_k = g(u_k, x_{k-1}) + \varepsilon_k \quad 2.16$$

$$z_k = h(x_k) + \delta_k \quad 2.17$$

Ces modèles généralisent ceux postulés dans l'Équations (2.14) and (2.15), ainsi la fonction  $g$  remplace les matrices  $A_k$  et  $B_k$ , et  $h$  remplace  $C_k$ .

Afin d'utiliser des modèles non linéaires, le filtre de Kalman étendu (EKF) propose une linéarisation via un développement de Taylor en calculant la Jacobienne de chaque modèle [17]

$$G_k = \left. \frac{\partial g}{\partial x} \right|_{x=\mu_{k-1}} \quad 2.18$$

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{x=\bar{\mu}_k} \quad 2.19$$

En faisant cela, nous linéarisons les modèles autour de l'estimation actuelle et par conséquent les mécanismes de propagation de croyances seront équivalents à ceux du filtre de Kalman classique. Les différences entre EKF et KF sont présentées dans le Tableau 3.

	Filtre de Kalman	Filtre de Kalman Étendu
Prévision	$\bar{\mu}_k = A_k \mu_{k-1} + B_k u_k$ $\bar{\Sigma}_k = A_k \Sigma_{k-1} A_k^T + Q$	$\bar{\mu}_k = g(u_k, \mu_{k-1})$ $\bar{\Sigma}_k = G_k \Sigma_{k-1} G_k^T + Q$
Correction	$K_k = \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + R)^{-1}$ $\mu_k = \bar{\mu}_k + K_k (z_k - C_k \bar{\mu}_k)$ $\Sigma_k = (I - K_k C_k) \bar{\Sigma}_k$	$K_k = \bar{\Sigma}_k H_k^T (H_k \bar{\Sigma}_k H_k^T + R)^{-1}$ $\mu_k = \bar{\mu}_k + K_k (z_k - h(\bar{\mu}_k))$ $\Sigma_k = (I - K_k H_k) \bar{\Sigma}_k$

Tableau 3- Différence entre le Filtre de Kalman et le Filtre de Kalman Étendu

Pour conclure, Thrun, S. pointe dans [17], que l'EKF n'est applicable que pour une localisation locale car les filtres de Kalman sont Gaussiens et ne représentent que des distributions unimodales de l'incertitude alors que cette dernière est plus complexe dans un contexte de localisation globale.

### 2.5.2 Filtre Particulaire (MCL)

Les filtres particuliers constituent une alternative populaire aux techniques gaussiennes, ils ne reposent pas sur une forme fonctionnelle fixe de la croyance postérieure, comme les filtres de Kalman [34]. Au lieu de cela, ils approchent les postérieurs par un nombre fini d'échantillons, appelés particules, ce qui les rend intéressant lorsqu'un robot doit faire face à des incertitudes globales.

Les filtres particuliers utilisé dans le cadre spécifique de l'estimation de la position d'un robot dans une carte donnée sont généralement désignés par le terme MCL (*Monte Carlo Localization*) [34]. (Par soucis de lisibilité et de compréhension, nous allons assimiler le terme MCL au filtre particulaire pour la suite de ce document)

L'algorithme classique du MCL représente la croyance par un ensemble de particules [34]

$$Bel^+(x_k) \approx S_k = \left\{ \langle x_k^{(i)}, w_k^{(i)} \rangle \mid i = 1, \dots, N_{particules} \right\} \quad 2.20$$

Où chaque  $x_k^{(i)}$  est une position possible du robot dans son environnement qui est associé à un facteur d'importance  $w_k^{(i)}$ . Le calcul de la distribution de probabilité a posteriori  $Bel^+(x_k)$  se fait généralement en trois étapes [17], [34] : la prévision, la mise à jour et le rééchantillonnage.

La prévision est effectuée en déplaçant les particules suivant la proposition de distribution  $P(x_k | x_{k-1}, u_k)$ . Ensuite, les poids  $w_k^{(i)}$  sont calculés à l'aide du modèle perceptuel  $P(z_k | x_k)$ . Enfin, dans l'étape de rééchantillonnage, les particules de faible poids sont remplacées par celles de poids significatifs. Cette dernière étape peut être effectuée à

travers plusieurs méthodes (Plusieurs d'entre elles sont traitées dans [17]). (L'algorithme général du filtre particulaire (MCL) est décrit ci-dessous dans le Tableau 4.)

---

**Algorithme 3 : Filtre particulaire (MCL)**

---

Entrée :  $u_k, z_k, \text{carte de l'environnement} : m, N_{\text{particules}}$

Étape 1 : *Initialisation :*

*Nous générons aléatoirement  $N_{\text{particules}}$  particules afin d'obtenir la croyance postérieure initiale  $Bel^+(x_0)$*

Étape 2 : *Phase de prévision :*

**POUR**  $k = 1, \dots, N_{\text{particules}}$

$$x_k^{(i)} \sim P(x_k^{(i)} | x_{k-1}^{(i)}, u_k)$$

**FIN POUR**

Étape 3 : *Phase de correction :*

**POUR**  $k = 1, \dots, N_{\text{particules}}$

$$w_k^{(i)} \sim P(z_k | x_k^{(i)}, m)$$

**FIN POUR**

Étape 4 : *Rééchantillonnage :*

**POUR**  $k = 1, \dots, N_{\text{particules}}$

*Redistribuer les particules  $x_k^{(i)}$  suivant leurs poids  $w_k^{(i)}$*

**FIN POUR**

Sortie :  $Bel^+(x_k) \sim \{x_k^{(i)}, w_k^{(i)} \mid i = 1, \dots, N_{\text{particules}}\}$

---

Tableau 4- Algorithme général du filtre particulaire (MCL)

Comme le suggère [17], MCL peut représenter des distributions de probabilités multimodales complexes contrairement à d'autres méthodes d'estimation comme le filtre de Kalman. Cependant cette technique n'est généralement pas capable de suivre plusieurs hypothèses

dans des environnements indiscernables, cela peut être nuisible, car nous perdons la trace de positions éventuellement correctes. Toujours selon [17], cette lacune peut être liée directement au choix de la proposition de distribution  $P(x_k|x_{k-1}, u_k)$ . Nous discuterons plus en détails de cette dernière lors du prochain chapitre.

## 2.6 Conclusion

Cet état de l'art permet, en premier lieu, de définir le problème général de localisation en robotique mobile terrestre ainsi qu'une approche probabiliste proposant un cadre d'estimation récursive basée sur la théorie Bayésienne. De plus, différentes méthodes d'implémentations du filtre bayésien sont traitées (filtre de Kalman et le filtre particulaire) ainsi que leurs limites concernant l'estimation d'état dans la localisation en robotique mobile terrestre. Finalement, cet état de l'art justifie le choix du filtre particulaire (MCL) qui permet de représenter des distributions de probabilités multimodales complexes.

## Chapitre 3 - Architecture de localisation proposée

Comme nous l'avons mentionné dans le chapitre précédent, parmi de nombreuses techniques de localisation, MCL est sans doute devenue l'approche la plus populaire à ce jour. Cependant, la technique MCL standard est parfois incapable de maintenir des distributions de croyances multimodales qui sont présentes dans des situations complexes telles que les environnements symétriques [21]. Nous proposons donc une nouvelle approche de l'architecture du MCL qui permet d'estimer la position d'un AGV dans tous types d'environnements d'intérieurs

### 3.1 Présentation des limites de l'architecture MCL classique

La phase de prévision du MCL est primordiale dans le processus d'estimation de la position du robot car c'est durant cette phase que les particules sont propagées et dans le cas d'une prévision sensiblement incertaine, la correction peut s'avérer inefficace, et de ce fait avoir comme conséquence une divergence du filtre [36]. C'est dans cette optique que nous allons nous concentrer sur cette phase à travers ce chapitre et plus particulièrement sur le modèle de mouvement. En effet, si nous connaissons la distribution initiale de l'état du robot, la phase de prévision correspondra au « suivi » de la distribution de probabilité de transition d'état  $P(x_k | x_{k-1}, u_k)$  qui décrit le mouvement du robot.

#### 3.1.1 *Modèle de mouvement*

Il existe différents modèles de mouvement, le plus populaire et le plus utilisé dans la littérature concernant la localisation en robotique mobile terrestre est présenté dans [17]. Il s'agit d'un modèle décrivant la probabilité  $P(x_k | x_{k-1}, u_k)$  qui dépend principalement d'une

position initiale du robot  $x_{k-1}$  ainsi que de la commande  $u_k$ , cette dernière représente le changement incrémental de la position relative du robot mesurée par des capteurs, plus communément appelée « odométrie ».

Pour dériver un modèle de mouvement probabiliste en utilisant l'odométrie, la différence relative entre deux positions quelconques a dû être représentée par une concaténation de trois mouvements de base [17]: une rotation, un mouvement en ligne droite (translation) et une autre rotation.

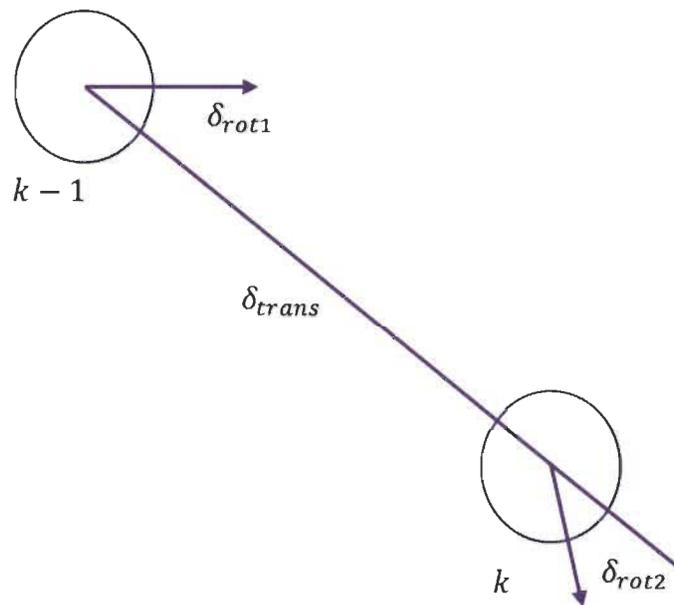


Figure 16- Illustration des trois mouvements de base du modèle d'odométrie

Les équations suivantes montrent comment calculer les valeurs des deux rotations et la translation à partir de la lecture de l'odométrie  $u_k = (\bar{x}_{k-1} \bar{x}_k)$ , avec  $\bar{x}_{k-1} = (\bar{x} \bar{y} \bar{\theta})$  et  $\bar{x}_k = (\bar{x}' \bar{y}' \bar{\theta}')$ .

$$\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta} \quad 3.1$$

$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2} \quad 3.2$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1} \quad 3.3$$

Pour modéliser l'erreur de mouvement, nous supposons que les valeurs « vraies » de la rotation et de la translation sont obtenues à partir des valeurs mesurées en soustrayant le bruit gaussien indépendant  $\varepsilon_b$  avec une moyenne nulle et une variance  $b$

$$\hat{\delta}_{rot1} = \delta_{rot1} - \varepsilon_{\alpha_1|\delta_{rot1}|+\alpha_2|\delta_{trans}|} \quad 3.4$$

$$\hat{\delta}_{trans} = \delta_{trans} - \varepsilon_{\alpha_3|\delta_{trans}|+\alpha_4|\delta_{rot1}+\delta_{rot2}|} \quad 3.5$$

$$\hat{\delta}_{rot2} = \delta_{rot2} - \varepsilon_{\alpha_1|\delta_{rot2}|+\alpha_2|\delta_{trans}|} \quad 3.6$$

Les paramètres  $\alpha_1$  à  $\alpha_4$  sont des paramètres d'erreur spécifiques au robot, qui définissent l'erreur accumulée durant le mouvement [17].

Par conséquent, la vraie position,  $x_k = (x' y' \theta')$ , est obtenue à partir de  $x_{k-1} = (x y \theta)$  par une rotation initiale d'angle  $\hat{\delta}_{rot1}$ , suivie d'une translation de distance  $\hat{\delta}_{trans}$ , suivie d'une autre rotation d'angle  $\hat{\delta}_{rot2}$ . Donc,

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1}) \\ \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1}) \\ \hat{\delta}_{rot1} + \hat{\delta}_{rot2} \end{pmatrix} \quad 3.7$$

La plupart des modèles de mouvement pour des robots mobiles se basent essentiellement sur des commandes  $u_k$  qui dépendent des mesures effectuées par les encodeurs du robot [17], [25], [23]. Néanmoins, ces derniers sont soumis à des incertitudes dues à des bruits pouvant provenir de différentes sources telles que : glissement des roues sur le sol, la conception

mécanique du robot, la résolution des encodeurs...etc. Ces erreurs sont cumulatives et augmentent au fil du temps et ne sont pas bornées [17], [18], [25], [34], [36].

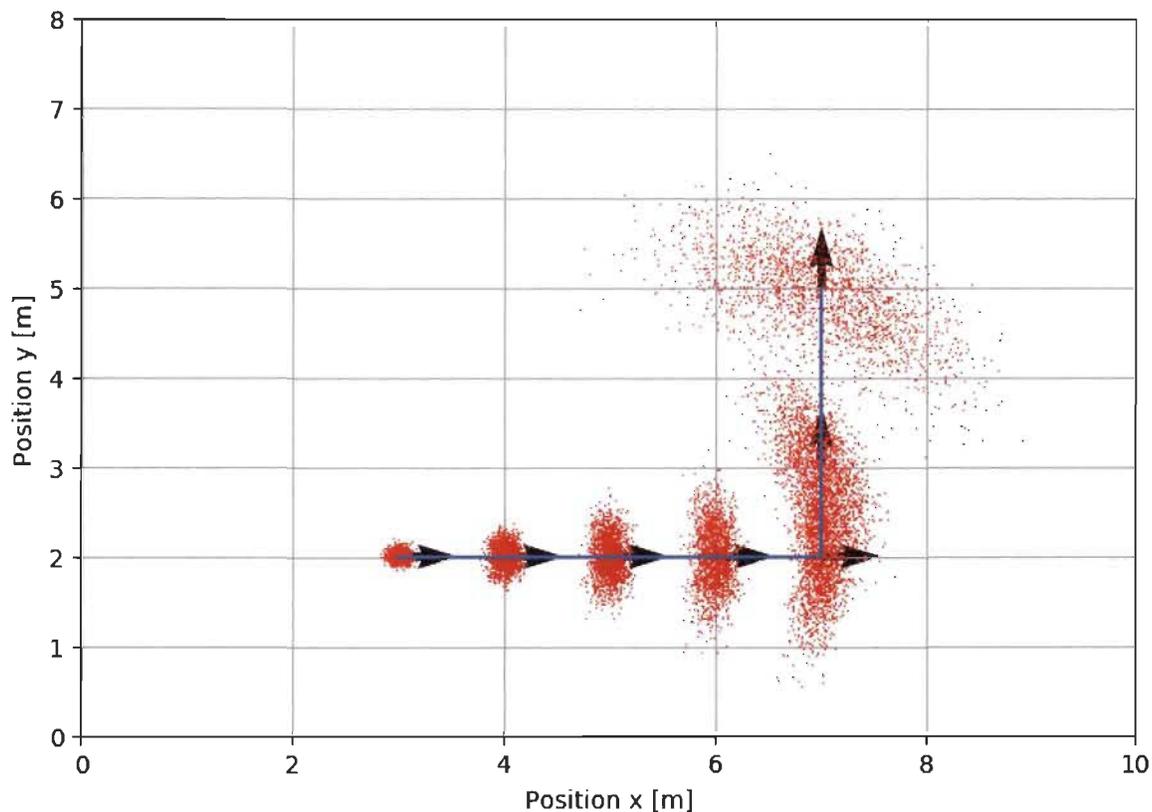


Figure 17 – Propagation des particules (points rouges) utilisant le modèle d'odométrie. La commande et la position « parfaite » du robot sont respectivement représentés par un trait bleu et une flèche noire

La position « parfaite » du robot est calculée analytiquement à partir du modèle d'odométrie sans l'ajout des bruits gaussiens définis plus haut. Comme le montre la figure ci-dessus, la dispersion des particules se fait de plus en plus grande au fil des commandes, ce qui rend la croyance de la position du robot incertaine.

### 3.1.2 Calcul du modèle de mouvement avec ICP

Dans cette section, nous présenterons comment utiliser l'algorithme Iterative Closest Point (ICP) [37] afin d'améliorer le modèle de mouvement. L'algorithme ICP est un algorithme populaire en robotique, qui peut attribuer des correspondances entre deux ensembles de points et récupérer la transformation qui relie un ensemble de points à l'autre.

Supposons que les points du laser à l'instant  $(k - 1)$  sont  $Q = \{q_i\}_{i=1}^{N_q}$ , et à l'instant  $k$  sont  $P = \{p_i\}_{i=1}^{N_p}$ . Les transformations relatives de la position du robot entre l'instant  $(k - 1)$  et  $k$  peuvent être calculer comme suit :

$$\min_{R, T, c(i) \in \{1, \dots, N_q\}} \left( \sum_{i=1}^N \|Rp_i + T - q_{c(i)}\|^2 \right) \quad 3.8$$

$$R'R = I \text{ et } \det(R) = 1$$

Où  $q_{c(i)}$  est le point le plus proche de  $p_i$ ,  $R$  est une matrice de rotation et  $T$  est un vecteur de translation. L'algorithme ICP résout le problème des moindres carrés présenté dans l'Eq. (3.8) en utilisant deux étapes itératives :

- 1- Avec les résultats de transformation obtenus dans l'itération précédente  $n - 1$ , on construit la nouvelle correspondance pour chaque point dans les ensembles de données  $P$  à partir de l'ensemble de données  $Q$ .

$$c(i) = \operatorname{argmin}_{j \in \{1, \dots, N_q\}} \left( \|R_{n-1}p_i + T_{n-1} - q_j\|^2 \right) \quad 3.9$$

- 2- En fonction du nouvel ensemble de correspondances, on calcul la nouvelle transformation comme suit :

$$(R_n, T_n) \underset{R, T, c(i) \in \{1, \dots, N_q\}}{\operatorname{argmin}} \left( \sum_{i=1}^{N_p} \|R p_i + T - q_{c(i)}\|^2 \right) \quad 3.10$$

L'équation (3.9) peut être résolue par de nombreuses méthodes efficaces, telles que la recherche du point le plus proche basée sur l'algorithme « *Arbre k-d* » ou ses variantes [38] et l'équation (3.10) peut être résolu par de nombreuses méthodes [39]. Pour le problème de correspondance de scan 2D, une solution efficace est présentée dans [40]. Lorsque l'algorithme ICP est convergent, nous pouvons récupérer la transformation relative  $\hat{s}_k$  de la position du robot entre l'instant  $(k - 1)$  et  $k$  [40]. Comme l'ICP effectue une association de données point à point explicite pendant l'itération, ceci peut introduire une erreur puisque les points de chaque balayage représentent une surface et non un ensemble d'emplacements discrets, donc pour simplifier les calculs, nous supposons que la distribution d'erreur des résultats ICP est gaussienne, avec des moyennes nulles et une variance constante  $\Sigma$ .

Enfin, l'estimation de mouvement du robot  $P(x_k | x_{k-1}, u_k)$  peut être calculé en utilisant les résultats ICP [36],

$$p(x_k | x_{k-1}, u_k) = x_{k-1} + M s_k \quad 3.11$$

Où  $s_k \sim N(\hat{s}_k, \Sigma_k)$ , et  $M$  étant une matrice définie comme suit :

$$M = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) & 0 \\ \sin(\theta_k) & \cos(\theta_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad 3.12$$

Le modèle de mouvement utilisant l'ICP est très précis car il utilise des informations extéroceptives, ce qui permet d'obtenir directement l'estimation de mouvement du robot sans

être soumis à une erreur cumulative comme celle présente dans le cas d'un modèle de mouvement utilisant des informations proprioceptives (l'odométrie par exemple).

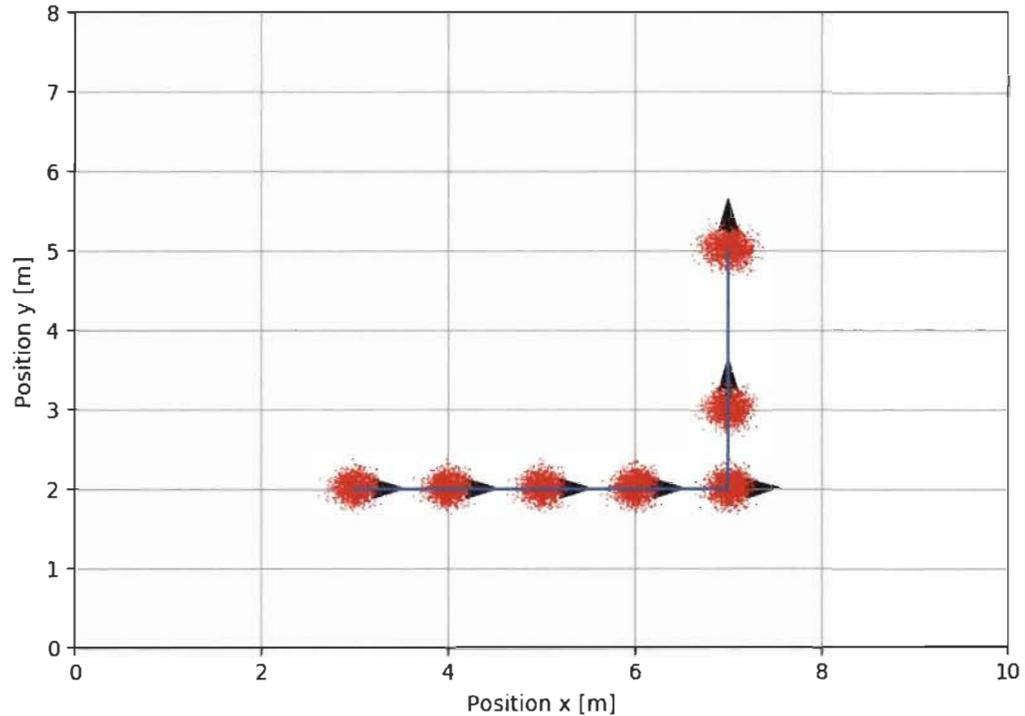


Figure 18- Propagation des particules (points rouges) utilisant le modèle d'ICP. La commande et la position « parfaite » du robot sont respectivement représentés par un trait bleu et une flèche noire

Cependant, cette méthode est soumise à une erreur incertaine dû à des ambiguïtés perceptuelles [25]. En réalité, dans des environnement dits symétriques et sans relief tel que les longs corridors, il n'y a pas suffisamment d'informations pertinentes afin d'estimer complètement la position du robot [41] car du point de vue du télémètre les perceptions sont identiques. (Voir Figure 19.)

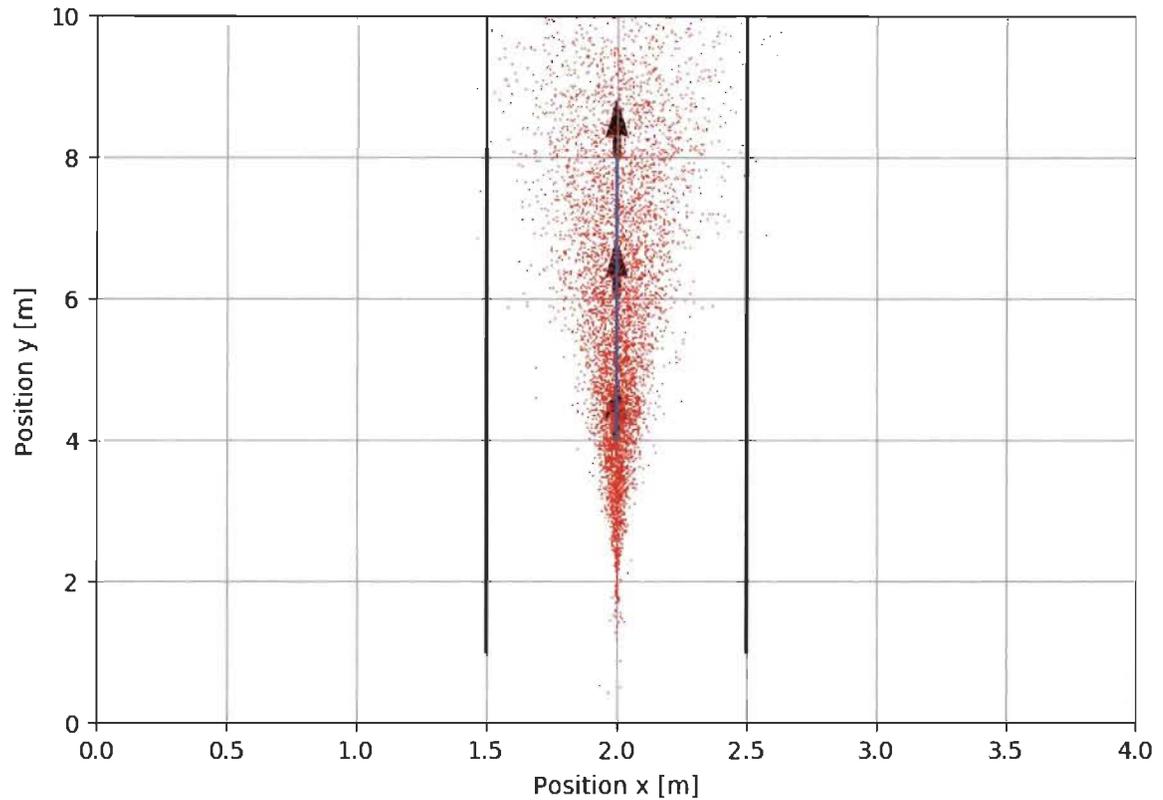


Figure 19- Propagation des particules (points rouges) utilisant le modèle d'ICP dans un long couloir. La commande et la position « parfaite » du robot sont respectivement représentés par un trait bleu et une flèche noire

Pour résumé, les limites de l'architecture standard de localisation utilisant MCL réside dans le choix de la proposition de distribution utilisée

### 3.2 Design d'un filtre de Kalman Étendu (EKF)

La solution pour résoudre les problèmes inhérents à chaque type d'informations, c'est de les fusionner. En effet, il est possible d'utiliser des capteurs proprioceptifs (les encodeurs) afin de différencier deux positions physiques distinctes mais similaires pour le système extéroceptif (télémètre laser) [25]. En d'autres termes, l'approche proposée permet de calculer la proposition de distribution  $P(x_k | x_{k-1}, u_k)$  à partir d'un filtre de Kalman Étendu (EKF) afin d'utiliser les informations pertinentes de deux systèmes de capteurs différents en

limitant les défauts spécifiques à chacun. Notons que l'utilisation d'autres capteurs pour la fusion de données est possible, tel que les capteurs inertiels, néanmoins le contexte de nos travaux se focalisent principalement sur les capteurs déjà présents sur les AGV industriels.

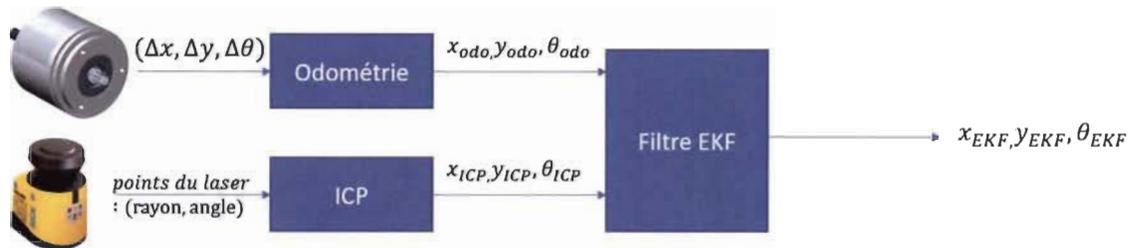


Figure 20- Schéma bloc simplifié du filtre EKF pour la fusion multisensorielle

Pour la suite de cette section, nous allons décrire en détails la solution proposée à base de filtre EKF ainsi que les concepts intrinsèques de l'algorithme.

### 3.2.1 Modèle du système EKF

Comme décrit auparavant, le filtre de Kalman Étendu (EKF) maintient son estimation locale postérieure de l'état, représentée par la moyenne  $\mu_k$  et la covariance  $S_k$ . Nous rappelons également que « l'astuce » de l'EKF réside dans la linéarisation du modèle de mouvement et de mesure. Pour cela, nous décomposons le modèle de mouvement en une partie sans bruit et une composante de bruit aléatoire avec une moyenne (approximativement) nulle tel que :

$$\underbrace{\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix}}_{x_k} = \underbrace{\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \delta_{trans} \cos(\theta + \delta_{rot1}) \\ \delta_{trans} \sin(\theta + \delta_{rot1}) \\ \delta_{rot1} + \delta_{rot2} \end{pmatrix}}_{g(u_k, x_{k-1})} + N(0, Q) \quad 3.13$$

Avec  $x_k$  le vecteur d'état du système à l'instant  $k$ ,  $g(u_k, x_{k-1})$  est le modèle de mouvement « sans bruit » avec  $u_k = (\delta_{trans}, \delta_{rot1}, \delta_{rot2})$  calculé à partir de (3.1),(3.2),(3.3),

et  $N(0, Q)$  est la composante d'un bruit aléatoire gaussien à moyenne nulle et d'une variance  $Q$ .

Rappelons aussi que la linéarisation de l'EKF approxime  $g$  par un développement de Taylor :

$$g(u_k, x_{k-1}) \approx g(u_k, \mu_{k-1}) + G_k(x_{k-1} - \mu_{k-1}) \quad 3.14$$

La fonction  $g(u_k, \mu_{k-1})$  s'obtient simplement en remplaçant l'état exact  $x_{k-1}$  -que nous ne connaissons pas- par notre estimation  $\mu_{k-1}$  -que nous connaissons. La Jacobienne  $G_k$  est la dérivée de la fonction  $g$  par rapport à  $\mu_{k-1}$ , et évalué en  $u_k$  et  $\mu_{k-1}$  :

$$G_k = \begin{bmatrix} 1 & 0 & -\delta_{trans} \sin(\mu_{\theta, k-1} + \delta_{rot1}) \\ 0 & 1 & \delta_{trans} \cos(\mu_{\theta, k-1} + \delta_{rot1}) \\ 0 & 0 & 1 \end{bmatrix} \quad 3.15$$

Avec  $\mu_{\theta, k-1}$  qui représente la troisième composante du vecteur  $\mu_{k-1}$ , qui est l'estimation de l'angle de rotation  $\theta$ .

### 3.2.2 Modèle de mesure EKF

Concernant le modèle de mesure, les résultats de la méthode ICP seront utilisés directement lors de l'étape de correction :

$$z_k = \begin{pmatrix} x_{ICP} \\ y_{ICP} \\ \theta_{ICP} \end{pmatrix} = x_k + N(0, R) \quad 3.16$$

Avec  $z_k$  étant le vecteur d'observation qui est obtenus à partir des mesures du télémètre laser et  $N(0, R)$  est la composante d'un bruit aléatoire gaussien à moyenne nulle et d'une variance  $R$ .

Comme le modèle de mesure est linéaire, nous n'avons pas besoin de calculer la Jacobienne, ainsi la matrice de mesure  $H$  est constante et est défini comme suit :

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad 3.17$$

### 3.3 Estimation de la matrice de covariance d'observation

Pour une implémentation efficace du EKF, il est primordial de configurer proprement les matrices de covariances et plus précisément celle induite par les erreurs d'observations [42]. Ces dernières sont liées à l'estimation de la position calculée par l'ICP qui dépend non seulement de la qualité des données acquises par le télémètre laser mais également de l'environnement. En effet, comme nous avons pu le montrer dans la section (3.1.2), la méthode de l'ICP est sujette à des ambiguïté perceptuelle due à la symétrie de certains éléments de l'environnement. Malheureusement, la méthode ICP décrite auparavant ne prend pas en considération ces ambiguïtés dans l'estimation de la position, et la covariance correspondante n'est généralement pas disponible. Par conséquent, pour améliorer la qualité de l'estimation de la position dans des environnements symétriques, il faudrait pondérer la sortie de l'ICP. Cet objectif peut être atteint en ajustant de manière adaptative la matrice de covariance d'observation  $R$ .

L'approche que nous proposons consiste à extraire, à partir des données du télémètre, les caractéristiques géométriques d'un environnement symétrique et par la suite incorporer l'influence de ces dernières dans l'estimation de la covariance.

### 3.3.1 Extraction des caractéristiques géométriques de l'environnement

La plupart des environnements d'intérieurs qui comportent des éléments symétriques peuvent être représentés par des segments de lignes [43], [44]. La représentation de la ligne est définie par un 'point normal' qui est l'intersection perpendiculaire de la ligne extraite et d'une ligne originaire du télémètre laser [44]. Ce point est caractérisé par ses coordonnées polaires : la distance  $\rho$  et l'angle  $\alpha$ .

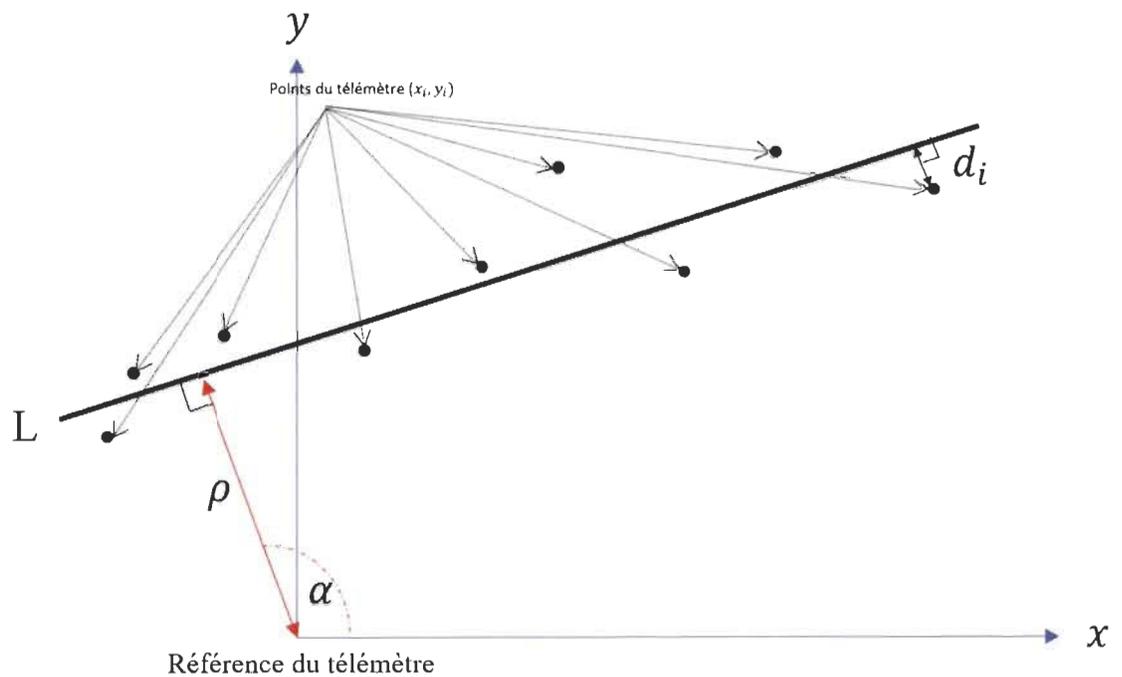


Figure 21- Représentation d'une ligne utilisant un point normal

Afin d'extraire une ligne à partir des mesures bruitées du télémètre laser, il suffit de trouver une ligne candidate  $L(\rho, \alpha)$  qui permet de minimiser la distance perpendiculaire  $d_i$  de chaque point du télémètre  $(x_i, y_i)$  à cette ligne (Voir Figure 19). Pour cela nous proposons la fonction de cout définie dans [44]

$$J = \min \sum_{i=0}^n (d_i)^2 = \min \sum_{i=0}^n (\rho - x_i \cos \alpha - y_i \sin \alpha)^2 \quad 3.18$$

La solution basée sur les moindres carrés pour la fonction de coût ci-dessus est donnée comme suit [44]

$$\rho = \bar{x} \cos \alpha + \bar{y} \sin \alpha \quad 3.19$$

$$\alpha = \frac{1}{2} \arctan \left( \frac{-2 \sum_{i=1}^n (\bar{x} - x_i)(\bar{y} - y_i)}{\sum_{i=1}^n (\bar{y} - y_i)^2 - \sum_{i=1}^n (\bar{x} - x_i)^2} \right) \quad 3.20$$

Avec  $\bar{x} = \frac{1}{n} \sum_{i=0}^n x_i$  et  $\bar{y} = \frac{1}{n} \sum_{i=0}^n y_i$

### 3.3.2 Calcul de la covariance

Une fois les segments de lignes extraits, les mêmes segments de lignes apparaissant dans deux balayages consécutifs du télémètre sont mis en correspondance pour dériver le changement de position relative et la covariance associée. La Figure 22 illustre ce changement de position entre deux balayages consécutifs.

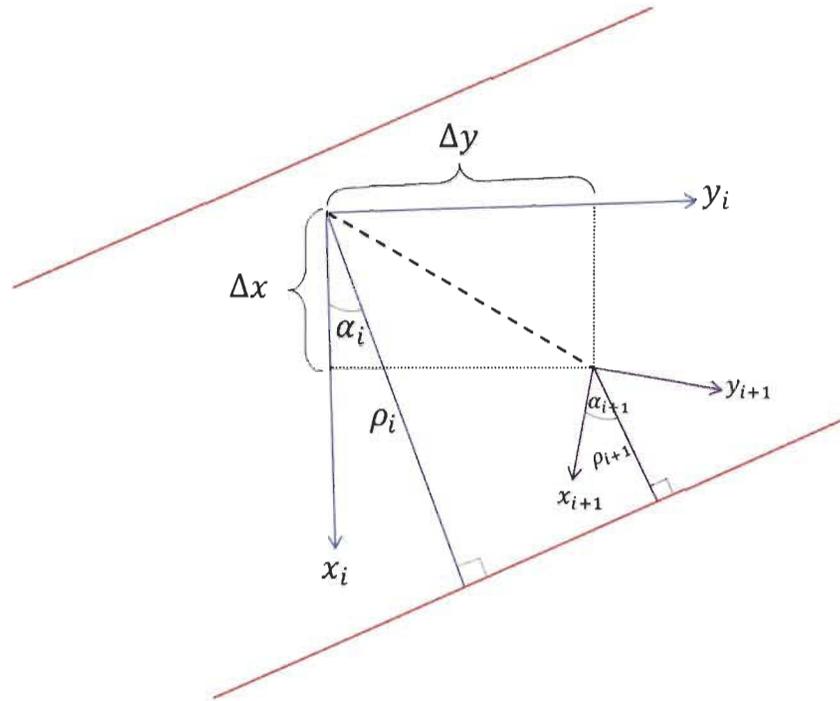


Figure 22- Changement de position entre deux balayages consécutifs

À partir de la Figure 22 et comme défini dans [43] et [44], La relation entre le changement de distance perpendiculaire  $\Delta\rho_{i,i+1}$  et le changement de position relative  $(\Delta x, \Delta y)$  est donnée ci-dessous,

$$\Delta\rho_{i,i+1} = \rho_i - \rho_{i+1} = \Delta x \cdot \cos\alpha_i + \Delta y \cdot \sin\alpha_i \quad 3.21$$

Supposons maintenant que  $k$ -lignes sont extraites et mises en correspondances pour deux balayages consécutifs ; l'équation (3.21) peut être écrites de la sorte,

$$\begin{bmatrix} \Delta\rho_{i,i+1}^1 \\ \dots \\ \Delta\rho_{i,i+1}^k \end{bmatrix} = \begin{bmatrix} \cos\alpha_i^1 & \sin\alpha_i^1 \\ \dots & \dots \\ \cos\alpha_i^k & \sin\alpha_i^k \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = H_\rho \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad 3.22$$

En appliquant la méthode des moindres carrés (LMS) sur l'équation (3.22), nous pouvons obtenir le vecteur de déplacement  $(\Delta x, \Delta y)$  ainsi que sa covariance correspondante  $R$  qui est calculée ainsi [43],

$$R = (H_\rho^T H_\rho)^{-1} \sigma_\rho$$

Où  $\sigma_\rho$  est la variance du changement de la distance perpendiculaire.

Lorsque des lignes parallèles sont extraites et mise en correspondance, des singularités peuvent survenir dues à la nature de la matrice  $H_\rho$ , si au contraire, au moins deux lignes perpendiculaires sont extraites, cette singularité va disparaître [43]. Effectivement, si on prend l'exemple d'un long couloir, la valeur inverse de  $H_\rho^T H_\rho$  va être très grande ce qui va induire une large covariance. En associant la covariance  $R$  à la position relative estimée par ICP, le filtre EKF va avoir plus de confiance dans le modèle d'odométrie plutôt que dans le modèle de mesure durant la fusion dans les environnements symétriques.

### 3.4 Algorithme d'un filtre hybride Kalman-MCL

Il est maintenant possible de générer une nouvelle proposition de distribution pour le filtre particulaire MCL à partir de l'EKF, ce cadre permet d'utiliser efficacement les dernières mesures d'observations lors de la phase de prévision du filtre MCL.

À l'instant  $k$ , le filtre EKF est d'abord utilisé pour propager chaque particule suivant la distribution suivante,

$$P(x_k^{(i)} | x_{k-1}^{(i)}, u_k) = N(\mu_k^{(i)}, \Sigma_k^{(i)}), \quad i = 1, \dots, N_{particules} \quad 3.23$$

Où  $\mu_k$  et  $\Sigma_k$  sont respectivement la moyenne et la covariance estimées par le filtre EKF du vecteur de position  $x_k$ .

À partir de la nouvelle proposition de distribution, le filtre à particule MCL va prendre le relais et estimer la position du robot dans une carte de l'environnement donnée. Finalement, l'algorithme du filtre hybride Kalman-MCL est présenté ci-dessous en détails,

---

### Algorithme 5 : Filtre hybride Kalman-MCL

---

Étape 1 : Initialisation :  $k = 0$

POUR  $i = 1, \dots, N_{particules}$  :

Générer des particules en supposant qu'on connaisse la distribution initiale de la position du robot :  $x_0^i \sim P(x_0^i) = N(\mu_0^{(i)}, \Sigma_0^{(i)})$

FIN POUR.

Étape 2 : POUR  $k = 1, 2, \dots$

(1) Utiliser le filtre EKF pour mettre à jour(propager) les particules :

POUR  $i = 1, \dots, N_{particules}$  :

(a) Prédire l'estimation de l'état et sa covariance avec EKF :

$$\begin{aligned}\bar{\mu}_k^{(i)} &= g(u_k, \mu_{k-1}^{(i)}) \\ \bar{\Sigma}_k^{(i)} &= G_k^{(i)} \Sigma_{k-1}^{(i)} G_k^{(i)T} + Q\end{aligned}$$

(b) Mettre à jour la prévision avec les nouvelles observations :

$$\begin{aligned}K_k^{(i)} &= \bar{\Sigma}_k^{(i)} H_k^{(i)T} \left( H_k^{(i)} \bar{\Sigma}_k^{(i)} H_k^{(i)T} + R_k \right)^{-1} \\ \mu_k^{(i)} &= \bar{\mu}_k^{(i)} + K_k^{(i)} (z_k - H \bar{\mu}_k^{(i)}) \\ \Sigma_k^{(i)} &= (I - K_k^{(i)} H) \bar{\Sigma}_k^{(i)}\end{aligned}$$

(c) Générer  $x_k^i \sim P(x_k^i | x_{k-1}^{(i)}, u_k) = N(\mu_k^{(i)}, \Sigma_k^{(i)})$

(d) Attribuer un poids à chaque particule suivant les données d'observations et la carte d'environnement.

FIN POUR.

(2) Normaliser les poids :

POUR  $i = 1, \dots, N_{particules}$  :

$$w_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1:N_{particules}} w_k^j}$$

FIN POUR.

---

---

**Algorithme 5 : Filtre hybride Kalman-MCL**


---

- (3) Rééchantillonner les particules en éliminant celles qui ont une faible pondération et en augmentant celles qui ont une plus grande pondération. La méthode utilisée est celle du KLD-Sampling [17]
- (4) Après le rééchantillonnage, nous disposerons du groupe de particules finales décrivant la croyance postérieure de la position du robot :

$$S = \left\{ x_k^{(i)}, w_k^{(i)} \right\}_{i=1}^{N_{particules}}$$

FIN POUR.

Étape 3 :  $k = k + 1$ , refaire étape 2 ou fin de l'algorithme

---

Tableau 5-Algorithme Kalman-MCL

### 3.5 Conclusion

L'une des limites du filtre particulaire (MCL) est d'abord présentée dans ce chapitre. Cette dernière fait référence au choix de la proposition de distribution lors de la phase de prévision du filtre MCL. Ensuite, une nouvelle proposition de distribution basée sur un filtre de Kalman Étendu (EKF) est proposée afin de fusionner deux sources de données émanant respectivement d'un modèle d'odométrie et d'un modèle utilisant la méthode de l'ICP. Finalement, dans le but d'utiliser efficacement les données des capteurs dans des environnements symétriques, une méthode de calcul adaptative de la covariance d'observation de l'EKF est alors appliquée.

## Chapitre 4 - Implémentation et validation des résultats

Dans le but d'évaluer la nouvelle architecture de localisation proposée, nous allons en premier lieu présenter les détails de l'implémentation de la méthode dans l'infrastructure de développement logicielle pour robot (ROS) et par la suite estimer les résultats obtenus par les tests en simulation et en expérimentation.

### 4.1 Implémentation

#### 4.1.1 ROS

Afin de tester et d'évaluer notre méthode dans des conditions proches de la réalité, nous devrions avoir la possibilité de modéliser un AGV et ses différents attributs ainsi que des capteurs dans des conditions normales d'opérations pour incorporer les incertitudes de mesures dans nos calculs. Pour concrétiser cela, nous optons pour la célèbre plateforme ROS (Robotic Operating System) [45]. Il s'agit d'un méta-système d'exploitation qui fournit une collection de bibliothèques, d'outils et de conventions qui facilitent la mise en œuvre de systèmes robotiques complexes et de fonctionnalités telles que : la gestion de paquets, abstraction matérielle, construction de fonctions fréquemment utilisées, gestion de dispositifs de bas niveau et transfert de messages entre processus [46]. La modularité de ROS augmente considérablement la stabilité du système pendant le fonctionnement, car l'exécution des différentes tâches est gérée de manière indépendante. La plateforme peut prendre en charge l'intégration de scripts écrits avec des langages de programmation tels que C++, C, Java et

Python dans un seul projet. De plus, ROS dispose de nombreux outils utiles pour la visualisation, le débogage et la simulation de systèmes robotiques.

Le calcul dans ROS est effectué à l'aide d'un réseau de processus appelés *nœuds* ROS. Un *nœud* est un code indépendant et exécutable qui est responsable de l'exécution de certaines tâches tel que : initialisation, traitement des données entrantes/sortantes, simulation de capteurs bruités, cartographie, planification, localisation...etc. Chaque *nœud* peut communiquer avec d'autres à travers un bus de communication appelé *topics*. La connexion entre les *nœuds* est gérée par un *master* et suit le processus suivant [46]:

1. Un premier *nœud* avertit le *master* qu'il a une donnée à partager
2. Un deuxième *nœud* avertit le *master* qu'il souhaite avoir accès à une donnée
3. Une connexion entre les deux *nœuds* est créée
4. Le premier *nœud* peut envoyer des données au second

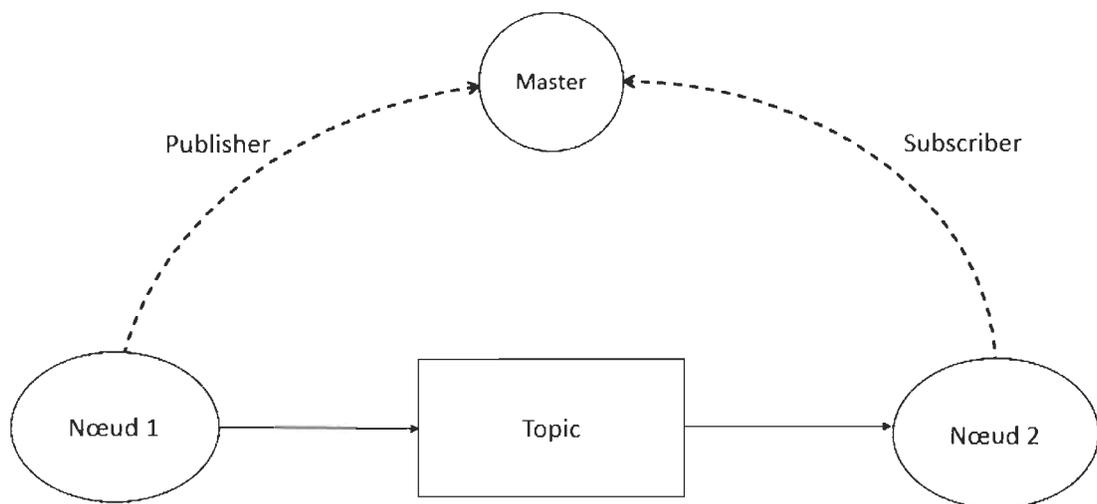


Figure 23- Schéma de fonctionnement du master sous ROS

Un *nœud* qui publie des données est appelé un *publisher* et un *noeud* qui souscrit à des données est appelé un *subscriber*. Un *noeud* peut être à la fois *publisher* et *subscriber* [46]. Les messages envoyés sur les topics sont pour la plupart standardisés ce qui rend le système extrêmement flexible.

#### 4.1.2 Description du robot

Dans l'optique de représenter un AGV et ses capteurs dans ROS, un modèle 3D de ce robot doit être défini. Par conséquent l'une des méthodes utilisées est celle du format URDF (Unified Robotic Description Format) [47], ce dernier est un fichier XML qui permet de décrire et de normaliser dans ROS tous les éléments spécifiques au robot (capteurs, actionneurs, joints, liens...etc). Nous avons donc pris comme modèle l'AGV conçu par la compagnie DIVEL Inc. (Québec, Canada) [48] qui développe et commercialise des plateformes mobiles pour le transport intra-usine. Cet AGV comprend deux moteurs différentiels avec des encodeurs incrémentaux ainsi que deux télémètres laser.

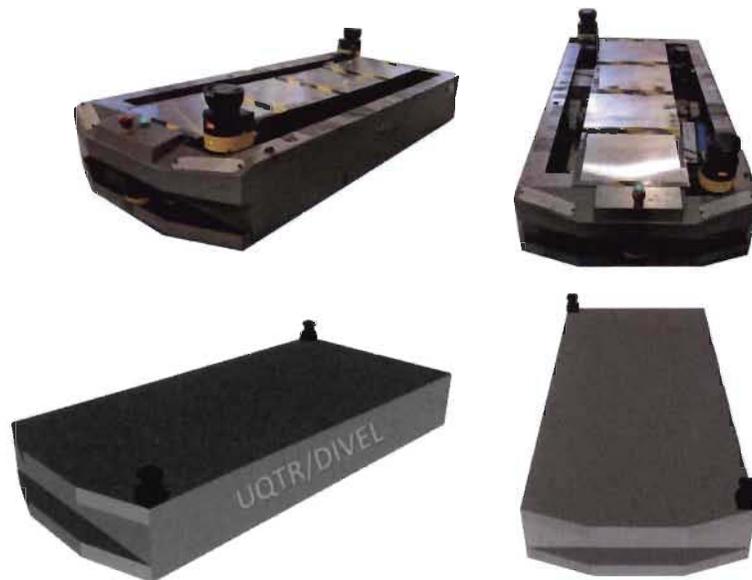


Figure 24-Plateforme mobile de DIVEL et son modèle 3D

Enfin, pour simuler les mesures des télémètres laser, des plugins génériques [49] sont proposés par ROS et qui permettent de spécifier les caractéristiques qu'on désire. Dans un souci de réalisme, nous avons donc défini les mêmes spécifications des télémètres (SICK MicroScan 3) [50] utilisés par l'AGV de DIVEL Inc. Un tableau récapitulatif des spécifications du robot utilisé en simulation est présenté ci-dessous.

<b>Spécifications</b>	<b>Valeurs</b>
<b>Châssis</b>	
Masse	90 Kg
Dimensions (longueur × largeur × hauteur)	168 × 76 × 23 cm
<b>Télémètre laser</b>	
Erreur statistique	± 10 mm
Résolution angulaire	0.25 degrés
Portée maximale	20 m
Ouverture angulaire	270 degrés
Fréquence de scan	20 Hz
<b>Moteurs</b>	
Type de contrôleur	PID
Vitesse linéaire max.	1.5 m/s
Vitesse angulaire max.	0.5 m/s

Tableau 6-Spécifications techniques de l'AGV

### 4.1.3 Représentation de l'environnement

Une fois le modèle de notre AGV défini, il est maintenant primordial d'établir l'environnement d'intérieur où le robot va opérer. Par conséquent, un modèle 3D d'un environnement d'intérieur complexe a été construit à partir du simulateur Gazebo [51] qui est un outil à part entière permettant la modélisation de « mondes » et leurs utilisations conjointes avec ROS. Dans le but de tester notre méthode de manière efficace, la construction de l'environnement a été faite de telle sorte à contenir des éléments complexes (virage, long couloir, zones structurées) comme peut le montrer la figure suivante.

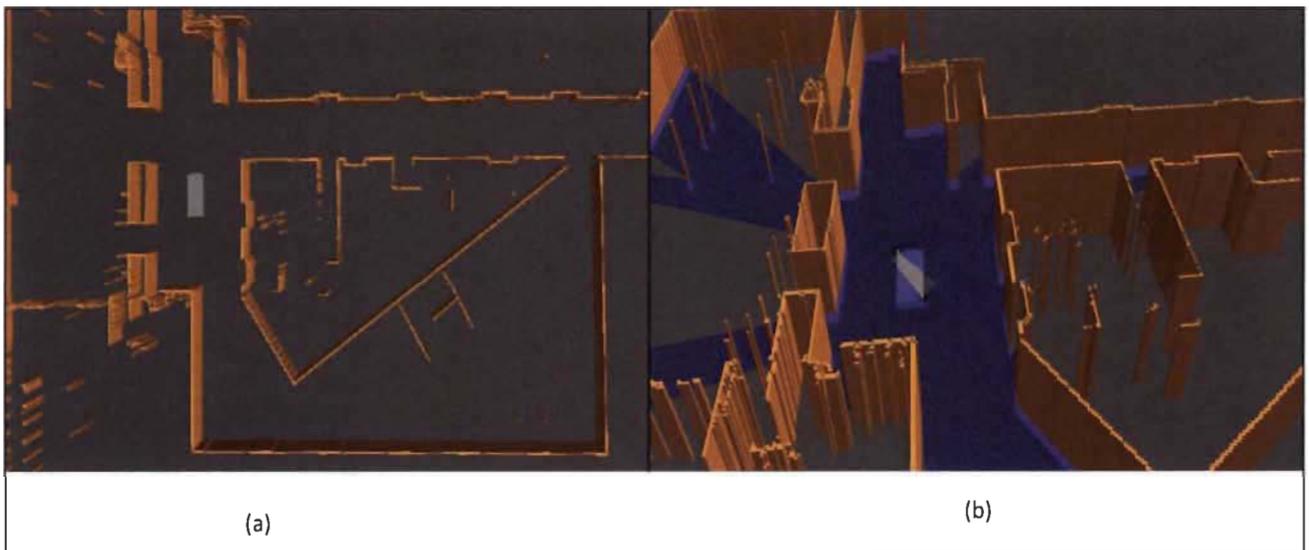


Figure 25- Représentation de l'environnement modéliser sur Gazebo. (a) représente une vue du dessus. (b) représente une perspective alternative de l'environnement ainsi que les mesures du télémètre simulées

Comme nous avons pu le mentionner dans l'état de l'art, afin de se localiser un robot doit avoir en mémoire une carte de son environnement. Pour ce faire, Il existe une multitude de méthodes et de recherches qui se font dans le domaine de la cartographie en robotique mobile terrestre et qui peuvent être référencées dans [52], néanmoins, le but de nos travaux ne s'inscrit pas dans cette lignée, donc pour générer une carte de l'environnement modélisée ci-dessus, nous avons utilisé le plugin du simulateur Gazebo « *gazebo\_ros\_2Dmap\_plugin* »

[53] qui permet la génération de carte sous forme de grille d'occupation 2D à partir du modèle de l'environnement.(Voir Figure 26 )

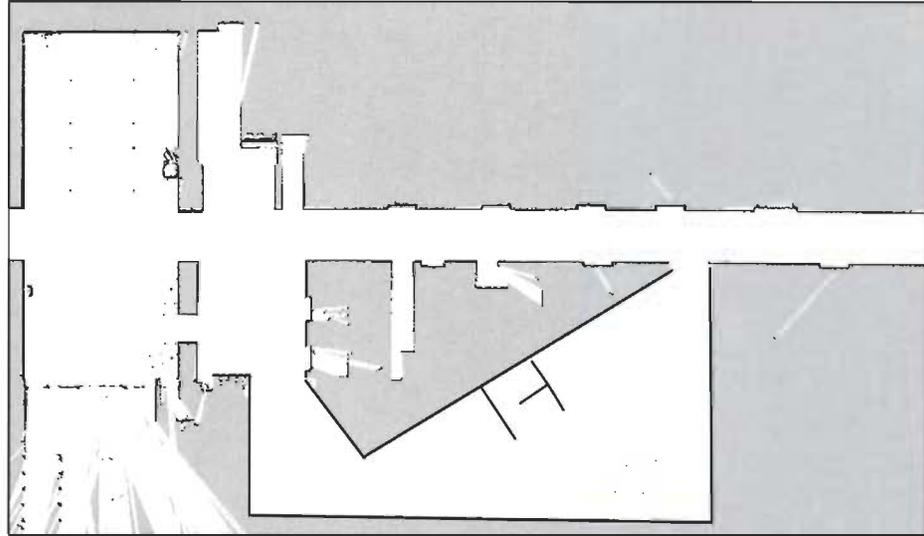


Figure 26-Carte de l'environnement sous forme d'une grille d'occupation. Les espaces occupés, libres et inconnus sont respectivement noirs, blancs et gris

## 4.2 Résultats de simulation

Avant de présenter les résultats de l'implémentation, il est nécessaire de montrer une vue d'ensemble des différents nœuds développés sur ROS qui sont impliqués dans la localisation ainsi que le rôle de chacun.

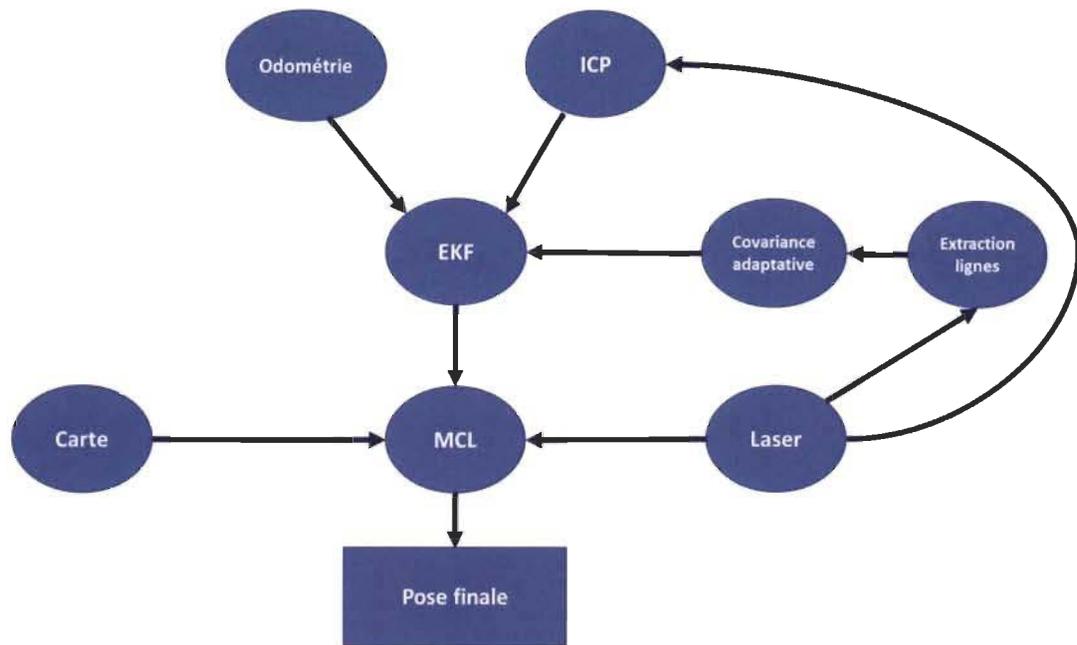


Figure 27- Schéma simplifié de l'architecture ROS de la structure de localisation.

Chaque nœud possède une fonctionnalité bien précise qui est résumée comme suit :

- Odométrie : Ce nœud permet de calculer la position relative dérivée du modèle d'odométrie décrit dans 3.1.1
- ICP : Ce nœud dérive une position relative à partir des mesures du télémètre laser simulées à partir du plugin proposé par ROS
- Extraction lignes : Ce nœud utilise les mesures brutes du télémètre laser pour la segmentation de ligne(s).
- Covariance adaptative : La covariance décrite dans 3.2.2 est calculée à chaque extraction de ligne(s)
- EKF : Ce nœud permet de fusionner les mesures émanant de l'Odométrie et l'ICP.

- MCL : Ce nœud est une implémentation d'un filtre particulaire qui utilise lors de sa phase de prévision la résultante de l'EKF. Les données de la carte ainsi que celles du télémètre laser sont quant à elles utilisées lors de la phase de mise à jour du filtre.
- Laser et Carte : Nous obtenons à partir de ces nœuds les données nécessaires du télémètre laser (Laser) et une grille d'occupation 2D (Carte)

Pour adresser l'influence de l'environnement sur le calcul de la covariance d'observation, les mesures brutes du télémètre laser sont récoltées dans deux zones différentes de l'environnement. (Voir Figure 28.)

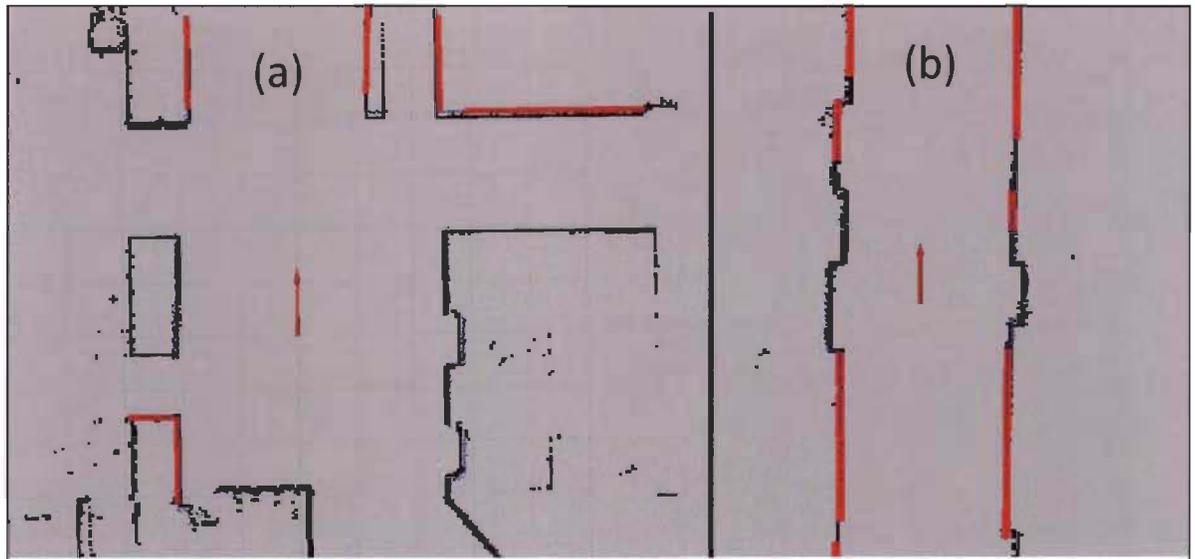


Figure 28- Représentation de lignes extraites (en rouge) dans deux zones différentes (a) et (b) à partir des mesures du télémètre (en bleu)

La figure 28.a illustre un exemple de lignes extraites à partir des mesures brutes du télémètres dans une zone impliquant des coins et bordures. Ici, six lignes ont été extraites. La figure 28.b montre quant à elle un exemple de lignes extraites dans un long couloir. Notons qu'elles sont toutes parallèles.

Les résultats de l'estimation de la covariance pour les deux exemples cités précédemment sont montrés dans le Tableau 7.

<b>Exemples</b>	<b>Covariance(x)</b>	<b>Covariance(y)</b>
Exemple (a)	$2.7 \times 10^{-3}$	$3.1 \times 10^{-3}$
Exemple (b)	$3.6 \times 10^{-3}$	$2.5 \times 10^{-1}$

Tableau 7- Covariances calculées dans les zones (a) et (b)

La covariance estimée dans le deuxième cas (b) est plus large que celle estimée dans le premier cas (a) suivant l'axe Y. Cela conduit à la conclusion que la disposition géométrique des lignes extraites crée une singularité qui a un impact sur l'estimation de la covariance d'observation et plus largement sur l'estimation de la position.

Une fois l'influence de l'environnement sur le calcul de la covariance démontrée, nous allons maintenant évaluer l'estimation de la position du robot générer par notre architecture de localisation. La Figure 29 montre la « vraie » trajectoire du robot qui effectue une boucle fermée. Cette trajectoire est fournie par le simulateur (à partir de son moteur physique) et va servir de référence dans un but de comparaison entre les différentes architectures de localisation.

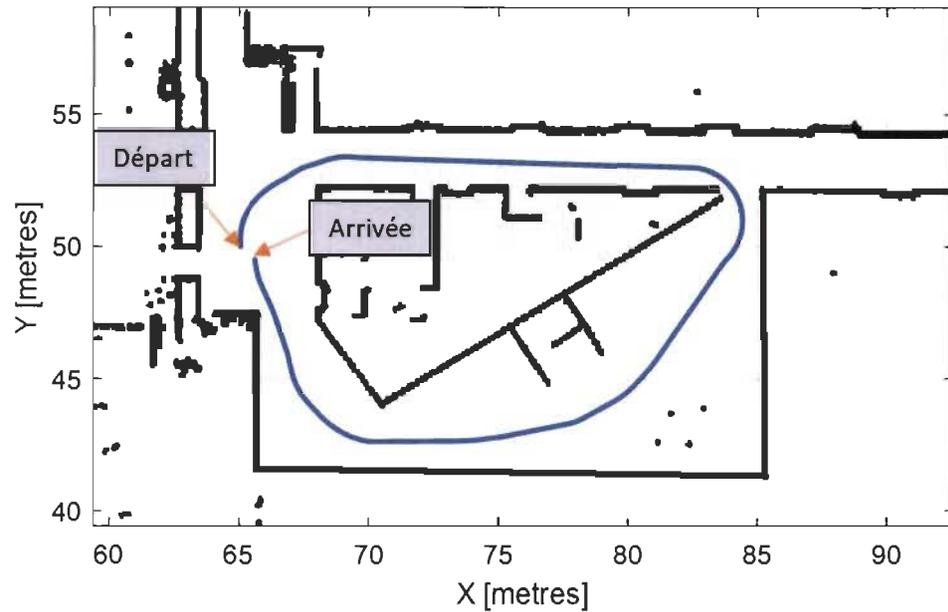


Figure 29- Trajectoire de référence du robot

Pour des raisons de clarté, les abréviations suivantes sont utilisées pour désigner les approches alternatives de localisation :

- MCL-O : Le filtre particulaire (MCL) standard basé sur la proposition de distribution générée par le modèle d'odométrie.
- MCL-I : Le filtre particulaire (MCL) standard basé sur la proposition de distribution générée par le modèle d'ICP.
- MCL-EKF : Notre solution basée sur le filtre particulaire (MCL) avec la nouvelle proposition de distribution générée par l'EKF.

La Figure 30 montre la trajectoire produite par chaque approche de localisation (MCL-O, MCL-I, MCL-EKF) ainsi que la vraie trajectoire effectuée par le robot. Nous observons qu'entre le point de départ (A) et le début du couloir (B), les trajectoires sont sensiblement les mêmes avec une légère dérive de MCL-O, cela s'explique par les incertitudes induites par

les mesures bruitées des encodeurs et du modèle d'odométrie. A partir du milieu du couloir, en plus de la dérive accumulée du MCL-O, la trajectoire du MCL-I commence à décrocher et à suivre une trajectoire sous-optimale, ceci est dû principalement à un minimum local induit par des ambiguïtés perceptuelles que subit le télémètre laser dans le couloir. Finalement, la seule trajectoire qui se rapproche de la vraie est celle produite par MCL-EKF. Cette solution permet d'utiliser une proposition de distribution hybrides entre le modèle d'ICP et celui d'odométrie grâce à la fonctionnalité de fusion de données de l'EKF et au calcul de la covariance d'observation. En effet, dans le couloir, nous pouvons observer que MCL-EKF n'est pas soumis aux erreurs perceptuelles car celles-ci sont compensées par les mesures d'odométrie. A noter que dès que le robot sort du couloir (C) une initialisation de la position de l'ICP est nécessaire afin que l'EKF puisse utiliser adéquatement ces mesures.

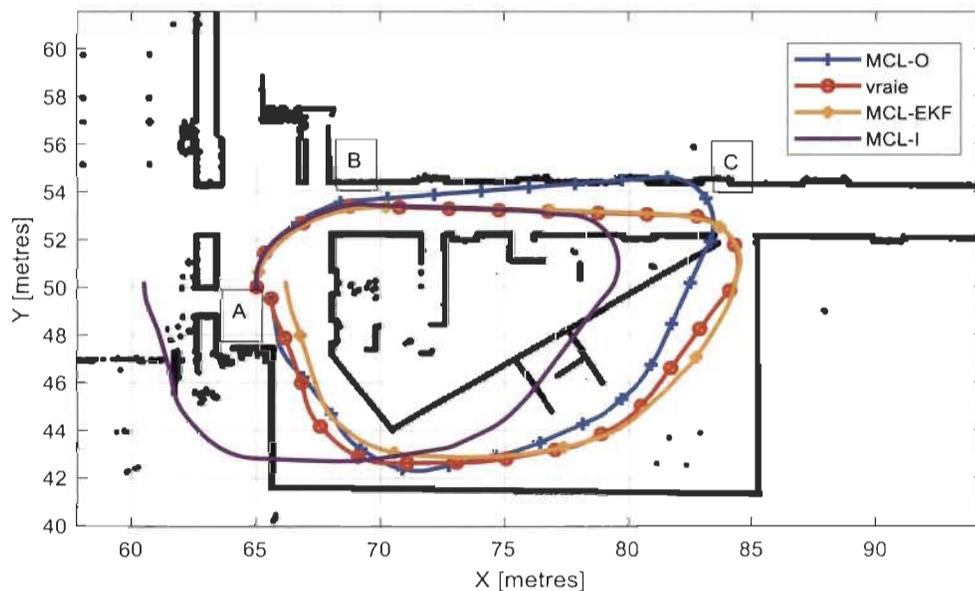


Figure 30- Trajectoires estimées par les différentes approches

Pour une représentation quantitative des résultats présentés précédemment, nous avons calculé l'erreur quadratique moyenne (MSE) [54] de chaque approche de localisation et nous obtenons les résultats dans le tableau suivant.

Approches localisation	MSE
MCL-O	0.5122
MCL-I	2.5838
MCL-EKF	0.0371

Tableau 8- Erreur quadratique moyenne des différentes approches de localisation

### 4.3 Validation expérimentale

Compte tenu de la situation sanitaire mondiale actuelle due au Covid-19, nous n'avons malheureusement pas pu compléter entièrement nos tests de validations (tests de précision et tests de robustesse). Cependant, nous avons pu tester partiellement la robustesse de nos algorithmes sur la plateforme (AGV) de notre partenaire industriel DIVEL Inc. en effectuant une navigation naturelle de 20 cycles consécutifs dans deux environnements de tests différents : Le centre d'essai de l'IRH (Institut de Recherche sur l'Hydrogène) et l'une des usines de DIVEL Inc. (Voir Figure 31 et 32). Un cycle correspond à faire passer l'AGV par un circuit défini par des stations (les stations font références à des points de passage de l'AGV dans son environnement).



Figure 31- Centre d'essai de l'IRH



Figure 32- Usine de DIVEL Inc.

Ces tests permettent d'évaluer la robustesse de notre solution en calculant le temps nécessaire accompli par la plateforme mobile sur plusieurs tours de circuits. Cette métrique permet de juger si l'AGV a pu naviguer sans échecs pouvant être causés par la localisation.

#### *4.3.1 Validation au centre d'essai de l'IRH*

Les résultats expérimentaux de la navigation de l'AGV au centre d'essai de l'IRH sont présentés dans le tableau suivant.

Longueur du circuit	79m				
Nombre de cycle	20				
Planificateur local utilisé	DWA [55]				
Vitesse linéaire	1m/s				
Vitesse angulaire	0.5 rad/s				
Accélération linéaire / angulaire	0.05 m/s <sup>2</sup> / 0.05 rad/s <sup>2</sup>				
Temps total de test	47.1 min				
Temps nécessaire pour un tour de circuit (les 5 premiers cycles)	2.33 min	2.32 min	2.35 min	2.31 min	2.32 min
Distance minimale par rapport aux obstacles	Entre 6'' et 12'' Entre 15cm et 30 cm				

Tableau 9- Paramètres et résultats de l'expérimentation à l'IRH

Un cycle de navigation au centre d'essai de l'IRH est montré dans la Vidéo 1.

Vidéo 1 : <https://www.dropbox.com/s/gs74j284wchr6as/Finale 20DWA 201m 20per 20s.mp4?dl=0>

#### 4.3.1 Validation à l'usine de DIVEL

Les résultats expérimentaux de la navigation de l'AGV à l'usine de DIVEL sont présentés dans le tableau suivant.

Longueur du circuit	90m				
Nombre de cycle	20				
Planificateur local utilisé	DWA [55]				
Vitesse linéaire	1 m/s				
Vitesse angulaire	0.5 rad/s				
Accélération\ décélération	0,05 m/s <sup>2</sup>				
Temps total de test	52.28 min				
Temps nécessaire pour un tour de circuit	2.44 min	2.44 min	2.44 min	2.33 min	2.44 min
Distance minimale par rapport aux obstacles	Entre 6'' et 12'' Entre 15cm et 30 cm				

Tableau 10- Paramètres et résultats de l'expérimentation à l'usine DIVEL

Un cycle de navigation dans l'usine de DIVEL est montré dans la Vidéo 2.

Vidéo 2 : [https://www.dropbox.com/s/q7boalck9axnctq/Finale 20usine 202 20 20DWA 201m 20par 20s.mp4?dl=0](https://www.dropbox.com/s/q7boalck9axnctq/Finale%20usine%202020%20DWA%201m%20par%20s.mp4?dl=0)

#### **4.4 Conclusion**

Une implémentation de notre méthode a été faite sur l'infrastructure de développement logicielle robotique (ROS). Le simulateur Gazebo a été utilisé afin de modéliser un AGV de la compagnie Divel Inc. ainsi qu'un environnement d'intérieur type. Par la suite, nous avons appliqué le filtre particulaire MCL avec trois propositions de distributions différentes à titre comparatif pour juger de la précision de chacune des méthodes. Cette comparaison a été faite en calculant l'erreur quadratique moyenne de chaque méthode avec la trajectoire « vraie » de l'AGV en boucle fermée. Finalement, pour juger de la robustesse, qui est définie comme étant la capacité de répétabilité de la solution à long terme, nous avons effectué des tests de validation en expérimentale, avec un AGV industriel réel de la compagnie Divel Inc., dans le centre d'essai de l'IRH ainsi que l'usine de Divel Inc.

La solution proposée permet d'estimer globalement la position d'un robot mobile terrestre dans des environnements d'intérieurs et de palier à la problématique sous-jacentes de localisation dans des environnements à connotations symétriques, et cela qu'avec des capteurs déjà embarqués sur le robot.

## Chapitre 5 - Conclusion

### 5.1 Conclusion et discussions

Pour conclure ce mémoire, il est important de revoir les objectifs originaux et la problématique introduits dans ce projet afin d'élaborer les bonnes conclusions. L'objectif principal est de développer une approche robuste de localisation d'un AGV dans des environnements d'intérieurs complexes.

Tout d'abord, nous avons étudié le concept de l'approche probabiliste bayésienne appliquée à l'estimation en robotique mobile terrestre. Deux implémentations courantes (le filtre de Kalman et le filtre particulaire) ont été exposées. Il s'est avéré que le filtre particulaire (MCL) soit l'approche la plus adéquate pour la résolution des problématiques classiques de localisation des robots mobiles dans des environnements connus. Néanmoins, l'une des limites du MCL a démontré la sensibilité de ce dernier face à différentes incertitudes introduites par le choix de la proposition de distribution (erreurs aléatoires non bornées pour le modèle d'odométrie, ambiguïtés perceptuelles pour le modèle d'ICP). De ce fait, une nouvelle proposition de distribution a été développée en se basant sur un filtre de Kalman Étendu qui va permettre une fusion des données pondérées de deux sources de capteurs (encodeurs et télémètre laser). Afin de prendre en considération l'influence de la géométrie de l'environnement sur les mesures d'observation du télémètre laser, nous avons conçu un EKF à covariance d'observation adaptative. Cette covariance est estimée en mettant en correspondance des segments de lignes extraites des mesures brutes du télémètre laser, cela

a permis de révéler une singularité lorsque au moins deux lignes parallèles sont extraites, ce qui correspond à la composition des environnements symétriques.

Ensuite, la nouvelle architecture hybride MCL-EKF est présentée. Au départ, l'algorithme produit grâce à l'EKF une estimation a priori de la position du robot, l'innovation ici réside sur l'incorporation des mesures d'observations lors de la phase de prévision du MCL, ce que l'architecture classique n'offrait pas. Puis, basée sur la position a priori et la grille d'occupation, le MCL fournira une estimation finale de la position du robot.

Pour juger de la pertinence de l'architecture proposée, nous avons évalué en simulation la précision de notre approche par rapport à d'autres approches classiques et nous avons obtenus une erreur quadratique moyenne inférieure à 4%. Enfin, l'implémentation de l'architecture dans un AGV différentiel du partenaire industriel DIVEL Inc. a permis d'effectuer des tests de robustesse dans différents environnements réels.

## 5.2 Perspectives

Pour les prochaines études, plusieurs points peuvent être proposés afin d'améliorer l'architecture de localisation présentée dans le présent document,

- Remplacer le filtre EKF par le filtre de Kalman « sans parfum » UKF. Le filtre UKF permet d'utiliser directement les modèles non linéaires à contrario de l'approximation et la linéarisation faites par l'EKF qui est susceptible d'introduire des erreurs.
- Il serait préférable d'estimer l'état de l'erreur au lieu de l'état au complet concernant le filtre EKF. Cette technique désignée sous le nom de : Error-State Extended Kalman Filter (ES-EKF) permet de séparer l'état réelle du robot de son erreur, ce qui permet d'estimer plus efficacement les incertitudes entre les différents capteurs.

- Extraction de caractéristique géométrique additionnelle pour discerner plus précisément l'environnement.

## Références (Bibliographie)

- [1] C. Sprunk, «Highly Accurate Mobile Robot Navigation,» Albert-Ludwigs-University of Freiburg, Department of Computer Science, 2015.
- [2] T. Muller, Automated guided vehicles, Berlin: Springer, 1983.
- [3] «Bien choisir un AGV,» [En ligne]. Available: <http://guide.directindustry.com/fr/bien-choisir-un-agv/>.
- [4] G. Ullrich, Automated Guided Vehicle : A Primer with Practical Applications, Berlin: Springer-Verlag Berlin Heidelberg, 2015.
- [5] «The Swiss-log Transcar Automated Guided Vehicle (AGV) system,» [En ligne]. Available: <https://www.nbt.nhs.uk/about-us/building-brunel/automated-guided-vehicle-system>.
- [6] «Toyota Introduces Automated Driverless Forklift Range,» 2016. [En ligne]. Available: <https://www.toyotamaterialhandling.com.au/press-room/2016/toyota-introduces-automated-driverless-forklift-range/>.
- [7] «Towing AGVs,» [En ligne]. Available: <https://www.jbtc.com/en/north-america/automated-systems/products-and-applications/products/towing-agvs>.
- [8] C. Feledy, «A State of the Art Map of the AGVS Technology,» Lung, 2017.

- [9] H. Martínez-Barberá et D. Herrero-Pérez, «Autonomous navigation of an automated guided vehicle in industrial environments,» *Robotics and Computer-Integrated Manufacturing*, vol. 26, n° 14, pp. 296-311, 2010.
- [10] «What is AGV and how it work?,» 2020. [En ligne]. Available: <http://www.agvblog.com/tag/magnetic-guided-agv/>.
- [11] «Introduction Inductive Track Guidance,» 2015. [En ligne]. Available: <https://www.goetting-agv.com/components/inductive/introduction>.
- [12] «Building AGVs,» [En ligne]. Available: <https://www.kollmorgen.com/en-us/solutions/automated-guided-vehicles/building-agvs/>.
- [13] «AGV Navigation,» [En ligne]. Available: <https://kinexon.com/solutions/agv-navigation>.
- [14] S.-Y. Lee et H.-W. Yang, «Navigation of automated guided vehicles using magnet spot guidance method,» *Robotics and Computer-Integrated Manufacturing*, vol. 28, n° 13, pp. 425-436, 2012.
- [15] H. Sobreira, P. Moreira et P. Costa, «Robust mobile robot localization based on a security laser: an industry case study,» *Industrial Robot: An International Journal*, vol. 6, n° 143, pp. 596-606, 2016.
- [16] G. Vasiljević, D. Miklic, I. Draganjac et Z. Kovacic, «High-accuracy vehicle localization for autonomous warehousing,» *Robotics and Computer-integrated Manufacturing*, vol. 42, pp. 1-16, 2016.
- [17] S. Thrun, W. Burgard et D. Fox, Probabilistic robotics, MIT Press, 2005.
- [18] R. Guyonneau, «Méthodes ensemblistes pour la localisation en robotique mobile,» Université d'Angers, 2013.

- [19] L. Ronghua et H. Bingrong, «Coevolution based adaptive monte carlo localization (CEAMCL),» *International Journal of Advanced Robotic Systems*, vol. 1, n° 13, p. 19, 2004.
- [20] G. Kootstra et B. De Boer, «Tackling the premature convergence problem in Monte-Carlo localization,» *Robotics and Autonomous Systems*, vol. 57, n° 111, pp. 1107-1118, 2009.
- [21] A. Malayeri et Ali, «Robot localization in symmetric environment,» 2010.
- [22] E. Sehestedt et F. Schneider, «MONTE CARLO LOCALIZATION IN HIGHLY SYMMETRIC ENVIRONMENTS,» *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics*, pp. 249-254, 2006.
- [23] O. Ait-Aider, «Localisation référencée modèle d'un robot mobile d'intérieur,» 2002.
- [24] S. Thrun, «Learning Metric-Topological Maps Maps for Indoor Mobile Robot Navigation,» *Artificial Intelligence*, vol. 99, n° 11, pp. 21-71, 1998.
- [25] D. Filliat, «ROB312 - Navigation pour les systèmes autonomes,» 2019.
- [26] T. Nam, J. Shim et Y. Cho, «A 2.5 D map-based mobile robot localization via cooperation of aerial and ground robots,» *Sensors*, vol. 17, n° 112, p. 2730, 2017.
- [27] «Topological Map Dataset,» 2018. [En ligne]. Available: [http://kaiyuzheng.me/documents/sites/cold/topo\\_map\\_dataset/#topological-map-dataset](http://kaiyuzheng.me/documents/sites/cold/topo_map_dataset/#topological-map-dataset).
- [28] «Rotary encoder,» 2020. [En ligne]. Available: [https://en.wikipedia.org/wiki/Rotary\\_encoder#Absolute](https://en.wikipedia.org/wiki/Rotary_encoder#Absolute).
- [29] «Codeur optique,» [En ligne]. Available: <http://www.codechamp.fr/codeur-optique.html>.

- [30] «Capteur ultrason : une technologie de pointe,» [En ligne]. Available: <https://www.eeca.eu/capteur-ultrason-une-technologie-de-pointe/>.
- [31] «Capteurs de distance : lequel choisir ?,» [En ligne]. Available: <https://gastondanslespace.wordpress.com/2015/10/14/capteurs-de-distance-lequel-choisir/>.
- [32] «Lidar,» [En ligne]. Available: <https://fr.wikipedia.org/wiki/Lidar>.
- [33] T. Ridene, «Co-registration of 3D heterogeneous geo-referenced data: contributions to the correction of mobile mapping systems surveys,» 2010.
- [34] V. Fox, J. Highwater, L. L. S. D et B. G., «Bayesian filtering for location estimation,» *IEEE pervasive computing*, vol. 2, n° 13, pp. 24-33, 2003.
- [35] Y. Kim et B. H., «Introduction to Kalman filter and its applications,» *Introduction and Implementations of the Kalman Filter. IntechOpen*, 2018.
- [36] J. Zhu, Z. Nanning et Y. Zejian, «An improved technique for robot global localization in indoor environments,» *International Journal of Advanced Robotic Systems*, vol. 8, n° 11, p. 7, 2011.
- [37] P. J. Besl et N. D. McKay, «Method for registration of 3-D shapes,» *Sensor fusion IV: control paradigms and data structures*, vol. 1611, pp. 586-606, 1992.
- [38] M. Greenspan et M. Yurick, «Approximate kd tree search for efficient ICP,» *Fourth International Conference on 3-D Digital Imaging and Modeling*, pp. 442-448, 2003.
- [39] A. Nüchter, J. Elseberg, P. Schneider et D. Paulus, «Study of parameterizations for the rigid body transformations of the scan registration problem,» *Computer Vision and Image Understanding*, vol. 114, n° 18, pp. 963-980, 2010.
- [40] F. Lu et E. Milios, «Globally consistent range scan alignment for environment mapping,» *Autonomous robots*, vol. 4, n° 14, pp. 333-349, 1997.

- [41] A. Censi, «An accurate closed-form estimate of ICP's covariance,» *Proceedings 2007 IEEE international conference on robotics and automation*, pp. 3167-3172, 2007.
- [42] F. Aghili et C. Y. Su, «Robust relative navigation by integration of ICP and adaptive Kalman filter using laser scanner and IMU,» *IEEE/ASME Transactions on Mechatronics*, vol. 21, n° 14, pp. 2015-2026, 2016.
- [43] S. Liu, M. M. Atia, Y. Gao et A. Noureldin, « Adaptive covariance estimation method for LiDAR-aided multi-sensor integrated navigation systems,» *Micromachines*, vol. 6, n° 12, pp. 196-215, 2015.
- [44] A. Soloviev, D. Bates et F. Van Graas, «Tight Coupling of Laser Scanner and Inertial,» *Navigation*, vol. 54, n° 13, pp. 189-205, 2007.
- [45] «ROS,» [En ligne]. Available: <https://www.ros.org/>.
- [46] «Documentation,» [En ligne]. Available: <http://wiki.ros.org/>.
- [47] «urdf,» [En ligne]. Available: <http://wiki.ros.org/urdf>.
- [48] «Divel,» [En ligne]. Available: <https://www.noovelias.com/agv-asrs>.
- [49] «Gazebo Plugins in ROS,» [En ligne]. Available: [http://gazebosim.org/tutorials?tut=ros\\_gzplugins](http://gazebosim.org/tutorials?tut=ros_gzplugins).
- [50] «Safety laser scanners,» [En ligne]. Available: <https://www.sick.com/ag/en/opto-electronic-protective-devices/safety-laser-scanners/microscan3/c/g295657>.
- [51] «Gazebo,» [En ligne]. Available: <http://gazebosim.org/>.
- [52] M. Filipenko et I. Afanasyev, «Comparison of various slam systems for mobile robot in an indoor environment,» *International Conference on Intelligent Systems*, pp. 400-407, 2018.

- [53] B. Gerkey, «gazebo\_ros\_2Dmap\_plugin,» [En ligne]. Available: [https://github.com/marinaKollnitz/gazebo\\_ros\\_2Dmap\\_plugin](https://github.com/marinaKollnitz/gazebo_ros_2Dmap_plugin).
- [54] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä et L. Meester, A Modern Introduction to Probability and Statistics: Understanding why and how, Springer Science & Business Media, 2005.
- [55] «dwa\_local\_planner,» [En ligne]. Available: [http://wiki.ros.org/dwa\\_local\\_planner](http://wiki.ros.org/dwa_local_planner).