

UNIVERSITE DE SAAD DAHLEB DE BLIDA

Faculté des sciences de l'ingénieur
Département d'Electronique

MÉMOIRE DE MAGISTER

Spécialité : Contrôle

LOCALISATION D'UN ROBOT MOBILE PAR LA
METHODE DE MISE EN CORRESPONDANCE DES
GRILLES AMÉLIORÉE PAR LA LOGIQUE FLOUE

Par

Mme LAKHDARI Wahida

Devant le jury composé de

H. SALHI	Maître de Conférence, U. de Blida	Président
A.GUESSOUM	Professeur. U. de Blida	Examineur
K. ACHOUR	Directeur de recherche au CDTA	Examineur
B. KAZED	Chargé de cours, U. de Blida	Examineur
R. OUIGUINI	Maître de recherche, ASAL. Alger	Rapporteur

Blida, juin 2005

RÉSUMÉ

Le présent travail porte sur la localisation absolue d'un robot mobile dans un environnement d'intérieur modélisé en utilisant une ceinture de 12 capteurs à ultrasons. L'environnement de navigation du robot est à base de grilles d'occupation.

L'approche adoptée est composée de trois étapes : acquisition d'une vue des capteurs à partir de la position courante du robot, mise en correspondance de la grille définissant l'espace de travail avec celle décrivant l'espace du robot perçu par les capteurs et calcul de la position et de l'orientation du robot mobile. La mise en correspondance est évaluée par des translations et des rotations d'une grille par rapport à l'autre autour d'une position estimée déduite du bloc de navigation.

La recherche de la meilleure correspondance entre les deux grilles est très coûteuse en temps de calcul. Pour cela, un système flou a été développé afin de réduire la zone de calcul. Ainsi qu'une méthode de recherche hiérarchisée a été mise en œuvre pour rendre la localisation réalisable en un temps relativement court.

ABSTRACT

An absolute mobile robot self localization method in a completely modeled indoor environment is presented.

A grid representation of the environment is adopted. We have set up an ultrasonic sensor model based on occupancy grids allowing to have for each of them, an equal contribution for the localization whatever the detected distance. Then, this model has been adapted for the use of a belt of 12 sensors.

The localization approach is composed of three steps : acquisition of information from robot current position and orientation thanks to the measures of sensors, the grid represents the environment of the robot which is directly perceivable from its position and the robot's evolution environment is represented in a grid describing limits of the free space are compared and the determination of robot's position and orientation in its environment. The correspondence is evaluated by translations and rotations between local grid and global grid surrounded of estimation position deduced from navigation bloc.

The research of the best matching between two grids is very expensive in time of calculation. This is why, a fuzzy system has been developed to limit calculating region and a graded research method has been achieved to give the localization's result in a relatively short time.

REMERCIEMENTS

Je tiens à remercier Monsieur RACHID OUIGUINI, Maître de recherche à l'ASAL, mon directeur de thèse, pour m'avoir fait bénéficier de ses conseils et de sa rigueur scientifique.

Je remercie Monsieur HASSEN SALHI, Maître de conférence à l'université de Blida, d'avoir accepté la présidence de mon jury.

Je tiens à exprimer ma reconnaissance à Monsieur K. ACHOUR, Directeur de recherche au CDTA, et Monsieur B. KAZED, chargé de cours à l'université de Blida, qui m'ont fait l'honneur d'examiner mon mémoire avec rigueur et attention.

Mes remerciements s'adressent également à Monsieur A.GUESSOUM, Professeur à l'université de Blida, qui a accepté de s'intéresser à ce travail et de participer au jury.

TABLE DES MATIERES

RESUME

REMERCIEMENTS

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

INTRODUCTION. 9

1. GUIDAGE ET LOCALISATION « MÉTHODES ET OUTILS »

1.1 La stratégie de guidage.	12
1.2 Planificateur de trajectoires utilisé.	15
1.3 Localisation en robotique mobile : techniques et capteurs	22
1.4 Différentes représentations d'environnement	34
1.5 Types de capteurs.	37
1.6 Conclusion.	42

2. MÉTHODE DE LOCALISATION UTILISÉE

2.1 Introduction	44
2.2 Étapes de la localisation.	45
2.3 Modélisation des capteurs à ultrasons	47
2.4 Modélisation en grille	50
2.5 Modélisation de l'espace navigable.	55
2.6 La méthode de mise en correspondance des grilles	58
2.7 Conclusion	64

3. INTÉGRATION DE LA LOGIQUE FLOUE

3.1 Introduction.	65
3.2 Structure générale d'un contrôleur flou.	66
3.3 Les systèmes flous développés.	70
3.4 Conclusion.	77

4. OPTIMISATION DE LA MÉTHODE ET RÉSULTATS DE SIMULATION

4.1 Introduction	78
4.2 Réduction des grilles	79
4.3 Résultats de simulation.	81
4.4 Conclusion.	95

CONCLUSION. 96

APPENDIX. 98

REFERENCE. 106

LISTE DES ILLUSTRATIONS, GRAPHIQUE ET TABLEAUX

- Figure 1.1 : Décomposition exacte (a) et approchée (b)
- Figure 1.2 : S_{11} , S_{12} , S_{21} et S_{22} sont les sommets à considérer
- Figure 1.3 : Les angles $B(i)$ et P
- Figure 1.4 : Projection des sommets sur la droite robot-objectif
- Figure 1.5 : Trajectoire obtenue après calcul pour l'exemple de la figure 1.2
- Figure 1.6 : Formalisme du problème de localisation d'un robot mobile
- Figure 1.7 : Transfert d'un modèle local à un autre [1]
- Figure 1.8 : La trilatération
- Figure 1.9 : Principe de la triangulation
- Figure 1.10 : Calcul de la position à partir de données goniométriques
- Figure 1.11 : Localisation grâce à deux points
- Figure 1.12 : Exemple de représentation topologique d'un environnement constitué de couloir (C) et de pièces (P)
- Figure 1.13 : Calcul de la position grâce à l'odométrie
- Figure 1.14 : Répartition des capteurs ultrasons sur la ceinture du robot
- Figure 2.1 : Deux obstacles renvoyant la même mesure
- Figure 2.2 : Modélisation d'un capteur idéal
- Figure 2.3 : Modélisation d'un capteur réel
- Figure 2.4 : Réflexion multiple
- Figure 2.5 : Phénomène de spécularité
- Figure 2.6 : Phénomène de diaphonie
- Figure 2.7 : Grille pour un capteur idéal
- Figure 2.8 : Grille d'un capteur en prenant en compte l'ouverture angulaire
- Figure 2.9 : Distribution du capteur à ultrason
- Figure 2.10 : Probabilité du capteur à ultrason en 3D [16]
- Figure 2.11 : Les valeurs affectées aux cellules libres et occupées
- Figure 2.12 : Exemple d'un plan d'une pièce
- Figure 2.13 : La disposition des capteurs autour du robot et leurs lectures à un moment donné
- Figure 2.14 : Exemple de la construction d'une grille locale

- Figure 2.15 : Projection de l'espace occupé par le robot sur la grille
- Figure 2.16 : Mise en correspondance des grilles
- Figure 2.17 : Mise en correspondance des cellules
- Figure 2.18 : Organigramme de localisation
- Figure 2.19 : Organigramme général
- Figure 3.1 : Structure générale d'un contrôleur basé sur la logique floue
- Figure 3.2 : Fuzzification avec sept fonctions d'appartenance
- Figure 3.3 : Structure générale de la partie floue du contrôleur
- Figure 3.4 : Interprétation graphique du calcul du résultat d'un contrôleur flou de type Mamdani
- Figure 3.5 : Interprétation graphique du calcul du résultat d'un contrôleur flou de type Sugeno
- Figure 3.6 : Fonction d'appartenance de la distance
- Figure 3.7 : Fonction d'appartenance de la vitesse
- Figure 3.8 : Vitesse générée par le système flou en fonction de la distance et la vitesse(v)
- Figure 3.9 : Fonctions d'appartenance de l'écart en position
- Figure 3.10 : Fonction d'appartenance d'entrée et de sortie
- Figure 3.11 : La zone de recherche générée par le système flou en fonction l'écart en position
- Figure 4.1 : Principe de la recherche du robot
- Figure 4.2 : Organigramme de localisation hiérarchisée
- Figure 4.3 : Déplacement du robot et la vue des capteurs au moment de la demande de la position du 1^{er} plan d'environnement
- Figure 4.4 : Grille locale construite à l'arrêt du robot
- Figure 4.5 : Les résultats de la « position en X » pour le premier exemple (40x40)
- Figure 4.6 : Les résultats de la « position en Y » pour le premier exemple (40x40)
- Figure 4.7 : Les résultats de « l'orientation » pour le premier exemple (40x40)
- Figure 4.8 : « Erreur en position » pour le premier exemple (40x40)
- Figure 4.9 : « Erreur en orientation » pour le premier exemple (40x40)
- Figure 4.10 : Exemple 2, le déplacement du robot et la grille locale construite au moment de la demande de la position du robot
- Figure 4.11 : Grille locale construite à l'arrêt du robot (environnement 70*100 cellules)

- Figure 4.12 : Les résultats de la « position en X » pour le 2^{ème} exemple (100x70)
- Figure 4.13 : Les résultats de la « position en Y » pour le 2^{ème} exemple (100x70)
- Figure 4.14 : Les résultats de « l'orientation » pour le 2^{ème} exemple (100x70)
- Figure 4.15 : « Erreur en position » pour le 2^{ème} exemple (100x70)
- Figure 4.16 : « Erreur en orientation » pour le 2^{ème} exemple (100x70) en utilisant les deux systèmes flous
- Figure 4.17 : Exemple de trajectoire réelle parcourue par le robot et la trajectoire estimée correspondante (environnement 70*100 cellules)
- Figure 4.18 : Les résultats de la « position en X » pour le 2^{ème} exemple (100x70) en utilisant les deux systèmes flous
- Figure 4.19 : La « position en Y » pour le 2^{ème} exemple (100x70) en utilisant les deux systèmes flous
- Figure 4.20 : Les résultats de « l'orientation » pour le 2^{ème} exemple (100x70) en utilisant les deux systèmes flous
- Figure 4.21 : « Erreur en position » pour le 2^{ème} exemple (100x70) en utilisant les deux systèmes flous
- Figure 4.22 : « Erreur en orientation » pour le deuxième exemple (100x70) en utilisant les deux systèmes flous
- Figure 4.23 : Exemple 2, le déplacement du robot et la grille locale construite au moment de la demande de la position du robot
- Figure 4.24 : grille locale construite à l'arrêt du robot (environnement 70*100 cellules)
- Figure 4.25 : Exemple de trajectoire réelle parcourue par le robot et la trajectoire estimée correspondante (environnement 70*100 cellules)
- Figure 4.26 : Les résultats de la « position en X » pour le 3^{ème} exemple (100x70) en utilisant les deux systèmes flous
- Figure 4.27 : La « position en Y » pour le 3^{ème} exemple (100x70) en utilisant les deux systèmes flous
- Figure 4.28 : Les résultats de « l'orientation » pour le 3^{ème} exemple (100x70) en utilisant les deux systèmes flous
- Figure 4.29 : « Erreur en position » pour le 3^{ème} exemple (100x70) en utilisant les deux systèmes flous
- Figure 4.30 : « Erreur en orientation » pour le 3^{ème} exemple (100x70) en utilisant les deux systèmes flous

INTRODUCTION

Les robots mobiles sont largement utilisés dans les environnements industriels pour le transport de produits par exemple. Le plus souvent ces tâches sont répétitives et suivent un chemin bien défini, parfois même matérialisé par des lignes fixées au sol ou par des amers artificiels.

Il y a actuellement une forte tendance à élargir les milieux où évoluent les robots à des environnements de bureaux ou à des environnements domestiques. Les types d'applications possibles sont innombrables. Cela peut être des tâches de nettoyage et d'entretien ou encore une assistance à une personne handicapée dans des tâches d'exploration et de préhension. Un robot peut également servir de guide pour la visite d'un musée. On parle alors, de façon générale, de robotique d'intérieur.

Un tel cadre d'utilisation requiert que le système robotisé dispose d'un niveau minimum d'autonomie et de facilités de navigation. Pour ce faire, le système doit généralement accomplir trois tâches de base qui sont la localisation, la planification et la navigation. Parmi ces tâches, la localisation relativement à l'environnement occupe une place de choix puisqu'elle détermine le bon déroulement des deux autres. Elle consiste à calculer et à maintenir à jour la connaissance de la position et de l'orientation du robot dans un repère absolu lié à l'environnement de travail. On se restreint ici au cas de systèmes navigant sur un plan. Le robot est alors complètement localisé par deux paramètres de position et un paramètre d'orientation.

Parmi les différentes approches de localisation absolue existantes, une technique très répandue est celle basée sur un modèle. En effet la localisation dans un environnement intérieur utilise le plus souvent une carte, ou modèle, de l'environnement. Le principe général est que le robot perçoit en ligne et grâce à ces capteurs extéroceptifs un modèle local de son espace de travail qu'il met en rapport avec le modèle global puis estime la position et l'orientation courantes à partir desquelles il a acquis ces données.

Le modèle peut se présenter sous diverses formes comme les grilles d'occupation, les modèles à primitives géométriques ou encore les modèles topologiques. Le type de primitives utilisées dépend entre autre du type de capteur.

En effet, les informations issues du capteur et celles stockées dans le modèle doivent pouvoir être comparées.

Quel que soit le capteur utilisé, la localisation basée sur un modèle passe par trois étapes principales qui sont : Acquisition de l'information capteur, la mise en correspondance des données observées avec celles du modèle et l'estimation de la position et de l'orientation.

Cette succession de plusieurs traitements, qui généralement sont complexes ou mettent en jeu de grandes quantités de données, rend les méthodes basées sur le modèle coûteuses en temps de calcul.

La plupart des systèmes adoptent une approche qui consiste à combiner ces calculs avec une localisation relative (généralement odométrique). Cette dernière permet de maintenir à jour la connaissance de la position et de l'orientation avec un très haut niveau d'échantillonnage mais sa précision se détériore au fil du déplacement à cause de l'accumulation d'erreurs. Ce système permet de réduire la périodicité du recours à la localisation absolue. De plus, il lui fournit une estimation de la position et de l'orientation pour réduire la complexité et le temps de calcul. On parle alors de localisation incrémentale. Cependant, il arrive que cette estimation ne soit pas disponible (comme lors de la mise en service ou s'il y a défaillance de l'odométrie). Dans ce cas l'information provient du module de navigation.

La localisation avec la construction d'une représentation de l'espace sous forme d'une grille d'occupation à l'aide des capteurs télémétriques apparaît comme un choix idéal pour atteindre l'objectif de modélisation en temps réel d'un environnement dynamique. Dans ce cas, les données acquises sont des ondes acoustiques issues d'une ceinture des capteurs à ultrasons. L'utilisation des capteurs ultrasonores en robotique mobile a toujours séduit les chercheurs du domaine en raison de leurs prix, de leurs faibles consommations, de leur simplicité et de leur compacité.

En supposant que le modèle du capteur ultrasonore est connu et que le modèle de l'environnement est disponible, la localisation par mise en correspondance des grilles grâce à la ceinture des capteurs à ultrasons suit le même schéma général composé des trois étapes citées plus haut :

- ❖ Acquisition de l'information capteur : Le robot saisit une vue panoramique de son environnement à partir de sa position courante.
- ❖ Mise en correspondance des observations avec le modèle : Vise à trouver pour des appariements entre le modèle perçu et le modèle connu à priori.

- ❖ Calcul de la position et de l'orientation du robot mobile : Se fait grâce aux appariements établis dans l'étape précédente.

Le présent travail a pour but la mise en oeuvre d'un système de localisation absolue destiné à une application de la robotique mobile comme moyen d'aide à la mobilité des personnes handicapées moteur.

Il existe plusieurs contraintes dans ce genre d'application. La première est sans aucun doute le coût du système proposé. Cela implique des limites sur la performance des systèmes de commande et de perception du robot. La seconde contrainte vient de l'environnement qui ne doit pas être modifié. Enfin, la présence d'une personne sur le robot mobile rend le champ d'actions limité.

L'originalité de ce travail réside dans l'intégration de la logique floue dans l'étape de calcul de position. A cet effet, nous avons développé deux systèmes flous afin d'offrir plus de flexibilité à la zone de recherche et la position estimée afin d'avoir la capacité d'être variable et mieux adapté à la taille de l'environnement tout en maintenant son efficacité sur le temps de calcul.

Le manuscrit est organisé comme suit :

Le Chapitre 1 est une introduction aux problématiques de la navigation et de la localisation en robotique mobile. Les techniques les plus utilisées sont répertoriées. La localisation basée sur le modèle y est abordée. En fin de chapitre, les différents capteurs utilisés dans la localisation des robots mobiles sont introduits.

Dans le Chapitre 2 nous nous sommes focalisé sur la localisation par mise en correspondance des grilles. L'acquisition des information capteur (la première étape de la méthode) nécessite une modélisation du capteur à ultrasons. Nous détaillons les capteurs que nous utilisons pour notre étude. Nous donnons les principes physiques qui régissent leur fonctionnement. Nous proposons ensuite une étude détaillée de chaque étape de la méthode utilisée.

Selon les conclusions déduites dans le chapitre 2. Dans le Chapitre 3 nous proposons une solution qui consiste à utiliser deux systèmes flous. Le premier contrôle la vitesse de déplacement du robot, le second génère la zone de recherche nécessaire dans le calcul de la position et l'orientation du robot mobile.

Dans la première partie du Chapitre 4, on aborde le problème de la recherche d'une optimisation sur le temps de calcul de la méthode. La deuxième partie présente les résultats de simulation du système de localisation développé.

CHAPITRE 1

GUIDAGE ET LOCALISATION

« MÉTHODES ET OUTILS »

Ce chapitre est une introduction au domaine du guidage et de localisation en robotique mobile. La problématique du guidage et de la localisation et l'intérêt de la connaissance de la position et de l'orientation du robot par rapport à son environnement y sont présentés. Les différentes approches du guidage et de localisation ainsi que les représentations d'environnement associés sont énoncés. La localisation en environnement intérieur est ensuite abordée. Enfin, les capteurs utilisés pour la localisation sont présentés.

1.1 La stratégie de guidage

1.1.1 Introduction

Les principales capacités d'un robot mobile sont la perception, la décision et l'action. La perception lui permet de fournir une base (une représentation de l'espace qui l'entoure) au module de décision, pour que celui-ci puisse déterminer les actions à accomplir afin d'exécuter les tâches qu'on lui a confiées (par exemple une surveillance, une manipulation ou un déplacement).

L'objectif de la robotique est de développer des systèmes autonomes, c'est-à-dire capables d'exécuter des tâches à partir de descriptions de haut niveau. De telles descriptions indiquent ce que l'on veut obtenir plutôt que comment on peut l'obtenir, laissant le système libre de trouver la manière d'exécuter cette tâche. La réalisation d'un tel objectif demande la résolution d'un certain nombre de problèmes plus spécifiques. Bien souvent, l'exécution d'une tâche nécessite un déplacement, que ce soit du robot ou d'un autre objet. Le module de décision doit donc être capable de planifier un mouvement au sein de son environnement.

Dans le cas général, le problème de la planification de mouvement se présente comme suit: on veut savoir si, partant d'un point donné en environnement avec une certaine vitesse, le robot pourra-t-il rejoindre un autre point de l'environnement et l'atteindre avec une autre vitesse. Dans l'affirmative, nous voudrions trouver la solution qui minimise un certain critère (distance, temps de parcours, etc . . .).

Nous nous sommes intéressés à ce problème dans un cadre assez particulier. On considère en effet un environnement plan (\mathbb{R}^2) caractérisé par des obstacles fixes, et qui est connu (avec une précision suffisante). On place ensuite dans cet environnement notre robot qui est du type voiture, c'est-à-dire un objet rigide de forme simple (rectangulaire), et qui peut se déplacer en restant soumis à des contraintes :

- Des contraintes physiques liées à sa structure (par exemple, celle sur sa manoeuvrabilité) ;
- Des contraintes liées à sa nature (par exemple, limitant les forces qu'il peut subir) ;
- Des contraintes données par ses capacités (par exemple, ses limites en vitesse et accélération).

1.1.2 Les Différentes méthodes

On peut regrouper les principales méthodes de planification de chemins en trois grandes classes [32], [40], les deux premières définissant un graphe dans l'espace des configurations (respectivement entre des points ou des secteurs singuliers) alors que la troisième construit incrémentalement un chemin à partir des données.

1.1.2.1 Les méthodes de types 'squelette'

Le terme anglais ('roadmap') est plus parlant. Le principe est simple : on définit dans l'espace libre un ensemble de courbes (structurées en graphe), puis on complète ce graphe pour raccorder les configurations initiale et finale. On cherche alors le plus court chemin dans ce graphe entre ces configurations. Un tel graphe peut provenir d'un graphe de visibilité, d'un diagramme de Voronoi ou d'une construction similaire.

Dans le cas d'un graphe de visibilité, les obstacles sont supposés être polygonaux, et leurs sommets forment, avec l'origine et le but, les nœuds du graphe. Les arêtes sont les segments qui relient les nœuds sans intersection avec l'intérieur des obstacles. Le plus court chemin dans ce graphe comme un chemin de contact entre

l'origine et le but, c à d un chemin contenant au moins une configuration sur la frontière d'un obstacle.

Une autre méthode utilise pour définir le graphe une fonction continue de l'espace libre dans une partie de dimension un de lui-même, et dont la restriction à cette partie est l'identité (une telle fonction est appelée une rétraction en topologie, d'où le nom de la méthode).

1.1.2.2 Les décompositions en cellules

Cette méthode est probablement celle qui a été le plus étudiée. Elle consiste (comme son nom l'indique) en une décomposition de l'espace libre en régions ou (cellules) au sein desquelles les chemins peuvent être générés sans contraintes. La recherche est alors transposée dans le graphe de connectivité, c à d le graphe non orienté représentant les relations d'adjacence entre ces cellules. On déduit de la liste des cellules sélectionnée le chemin optimal (par exemple, on effectue la transition d'une cellule à sa voisine en passant par le milieu de leur intersection).

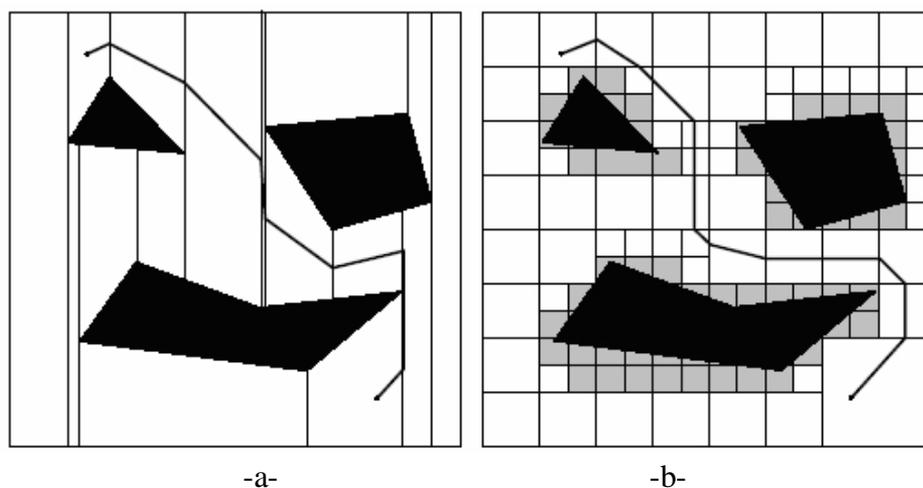


Figure 1.1 : décomposition exacte (a) et approchée (b)

Cette décomposition peut-être exacte (si l'union des cellules est égale à l'ensemble décomposé) ou au contraire approchée (si cette union est strictement incluse dans l'espace décomposé). Les méthodes exactes sont complètes (elles sont assurées de trouver une solution tant qu'il en existe) pourvu que l'algorithme d'exploration du graphe de connectivité le soit, mais sont souvent plus compliquées à implanter : la forme des cellules est alors plus complexe (pour pouvoir suivre les contours des

obstacles), et la détermination de leur limites difficile. Les méthodes approchées, quant à elles, utilisent généralement des cellules de forme et de dispositions régulières, aussi ne s'adaptent-elles pas parfaitement aux obstacles. Elles ont alors recours à une décomposition hiérarchiques de l'espace des configurations, en redécomposant localement les cellules jusqu'à un niveau de résolution donné. Le choix de la forme ou de la disposition conduit à une variété de solutions proposées. La plupart de ces méthodes sont dites complètes en résolution : s'il existe une solution, elle pourront la trouver moyennant un niveau de décomposition assez profond.

1.1.2.3 Les méthodes de type 'potentiel'[32]

Alors que les précédentes méthodes visaient à recouvrir l'espace des configurations libres grâce à un graphe (exploré par la suite), les méthodes dites du type 'potentiel' utilisent un tout autre principe. En effet, elles consistent à considérer le robot mobile comme une particule chargée dans l'espace des configurations, et soumise à l'influence d'un potentiel : elle est attirée par le but, et repoussée par les obstacles. Un tel potentiel a la propriété d'être indépendant de la position initiale. Il est composé d'une partie 'attractive', proportionnelle au carré de la distance au but, et d'une partie 'répulsive', qui tend vers l'infini quand la distance à un obstacle tend vers zéro. Le chemin optimal est alors déterminé itérativement, de façon à suivre la plus grande 'pente' de la fonction potentiel. L'inconvénient de telles méthodes est que, si le chemin construit mène toujours à un minimum, celui-ci peut être local et ne pas correspondre au but désiré. Une première solution pour résoudre ce problème est d'utiliser pour le potentiel une fonction qui, dans la partie connexe de l'espace libre qui contient le but, n'admette pas d'autre minimum. On peut aussi avoir recours à des mécanismes variés pour sortir des minima locaux.

1.1.3 Planificateur de trajectoires utilisé

Un itinéraire global est planifié sous la forme d'une suite d'emplacements, qui conduit le robot, de sa position actuelle à la destination donnée, cet itinéraire peut être le chemin le plus court du réseau pour atteindre le but final, l'algorithme de planification de trajectoire que nous avons utilisé, se place dans le cadre de la première classe des méthodes de planification de trajectoires, ou nous l'avons organisé en quatre procédures :

- limitation des sommets des obstacles.
- calcul du sommet le plus proche
- test d'existence des obstacles sur la ligne source-destination.
- calcul du but partiel

1.1.3.1 Limitation des sommets des obstacles

x_r, y_r : coordonnées du robot

x_s, y_s : coordonnées des sommets

x_{objet}, y_{objet} : coordonnées de l'objectif.

Une fois les coordonnées du robot et celles de l'objectif sont connues, le planificateur procède à la limitation des sommets des obstacles connus, ceci dans l'intérêt de minimiser le temps d'exécution du planificateur.

La connaissance des coordonnées du robot et celles de l'objectif, nous permettent d'évaluer la position du robot par rapport au sommet des obstacles connus, ensuite les limites par les formules suivantes :

- 1- Si $x_{objet} \geq x_r$ et $y_{objet} \geq y_r$, il faut considérer $x_s \geq x_r$ et $y_s \geq y_r$.
- 2- Si $x_{objet} \geq x_r$ et $y_{objet} \leq y_r$, il faut considérer $x_s \geq x_r$ et $y_s \leq y_r$.
- 3- Si $x_{objet} \leq x_r$ et $y_{objet} \leq y_r$, il faut considérer $x_s \leq x_r$ et $y_s \leq y_r$.
- 4- Si $x_{objet} \leq x_r$ et $y_{objet} \geq y_r$, il faut considérer $x_s \leq x_r$ et $y_s \geq y_r$.

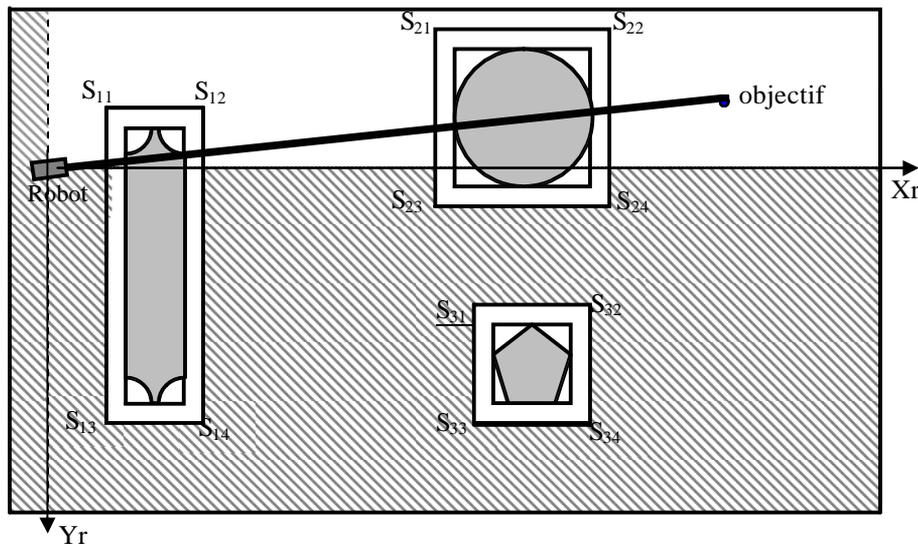


Figure 1.2 : S_{11} , S_{12} , S_{21} et S_{22} sont les sommets à considérer

1.1.3.2 Calcul du sommet le plus proche

Après avoir limité les sommets, on procède aux calculs des distances qui séparent le centre du robot aux sommets à considérer, évaluées par l'équation suivante :

$$D = \sqrt{(x_r - x_s(i))^2 + (y_r - y_s(i))^2} \quad (1.1)$$

x_r , y_r : coordonnées du robot.

x_s , y_s : coordonnées des sommets à considérer

i : numéro du sommet.

La distance minimum nous donnera le sommet le plus proche au robot.

1.1.3.3 Test d'existence des obstacles sur la trajectoire

La connaissance du sommet le plus proche conduit à la connaissance de l'obstacle le plus proche dont il faut vérifier sa présence sur la trajectoire en utilisant le calcul des angles suivants:

x_s , y_s : coordonnées des sommets de l'obstacle le plus proche.

x_r , y_r : coordonnées de l'objectif.

i : numéro du sommet.

P : l'angle que fait le robot avec l'objectif.

$B(1)$, $B(2)$, $B(3)$, $B(4)$: les différents angles que fait le robot avec les sommets de l'obstacle le plus proche.

Si $x_{\text{objet}} \neq x_r$ et $y_{\text{objet}} \neq y_r$.

$$P = \arctan((x_r - x_{\text{objet}}) / (y_r - y_{\text{objet}})) \quad (1.2)$$

$$B(i) = \arctan((x_r - x_s(i)) / (y_r - y_s(i))).$$

En considérant toujours l'exemple précédent voir la Figure 1.3.

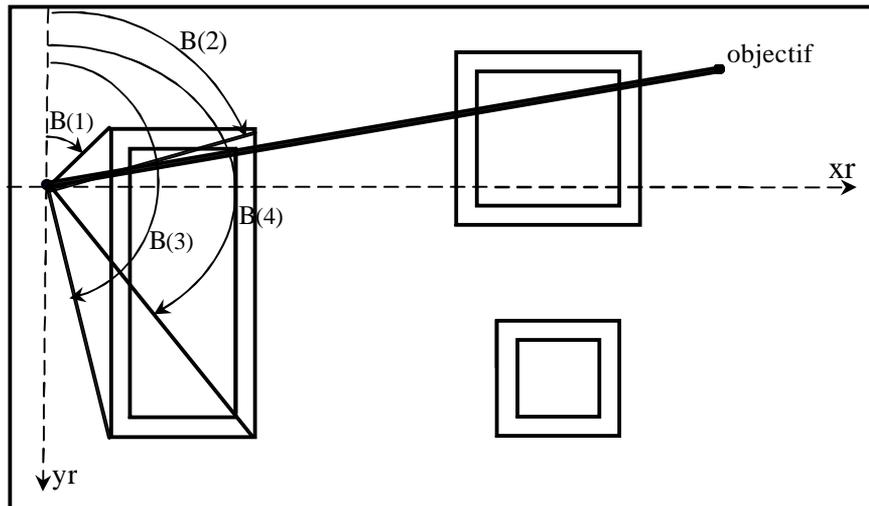


Figure 1.3 : les angles $B(i)$ et P

La figure 1.3 présente les différents angles que fait le robot avec l'objectif et les différents sommets de l'obstacle le plus proche.

$$F1 : B(1) < P < B(4) \quad (1.3)$$

$$F2 : B(2) < P < B(3). \quad (1.4)$$

- Si l'une des conditions F1 ou F2 est vérifiée donc l'obstacle est sur la trajectoire, alors il faut l'éviter.
- Sinon on procède à la recherche du nouveau sommet le plus proche, après avoir éliminer le sommet le plus proche obtenu auparavant.

1.1.3.4 Calcul du but partiel

Si l'obstacle est sur la trajectoire, il faut l'éviter en déterminant le but partiel, c.à.d le sommet par qui transite le robot en exécutant son chemin.

Cette procédure se réalise en deux parties:

Classification des sommets

C'est la détermination des sommets se trouvant au-dessus et au-dessous de la droite robot objectif.

$A(i) = A(1), A(2), A(3), A(4)$

$B(i) = B(1), B(2), B(3), B(4)$

$B(i)$: l'angle que fait le robot avec les sommets de l'obstacle le plus proche.

P : l'angle que fait le robot avec l'objectif.

$$A(i) = p - B(i) \quad (1.5)$$

Si $A(i) > 0$, les sommets se trouvent au-dessus de la droite robot objectif.

Si $A(i) < 0$, les sommets se trouvent au-dessous de la droite robot objectif.

Calcul de L_{min}

Une fois la classification obtenue, on projette les sommets de l'obstacle le plus proche se trouvant sur la trajectoire, sur la droite robot-objectif, comme le montre la figure 1.4 en considérant toujours l'exemple de la Figure 1.2.

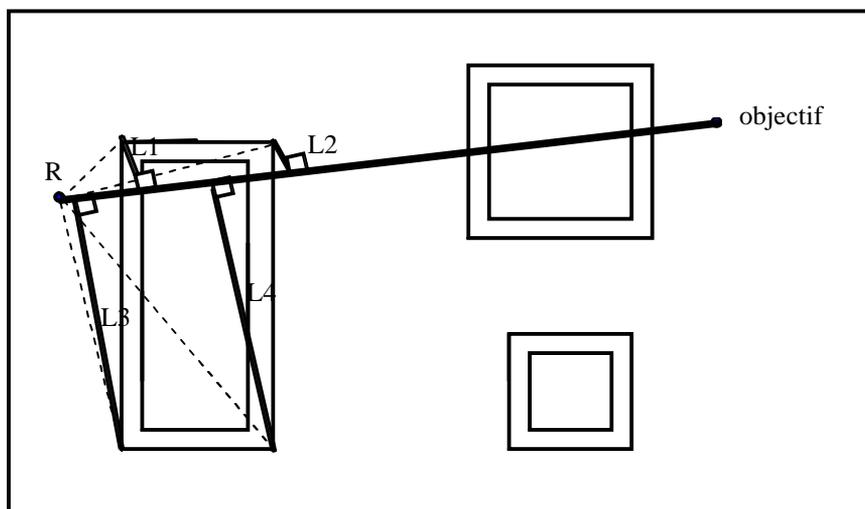


Figure 1.4 : projection des sommets sur la droite robot-objectif

Les calculs s'effectuent comme suit:

x_r, y_r : coordonnées du robot.

$x_s(m,i), y_s(m,i)$: coordonnées des sommets.

$d(m,i)$: la distance qui sépare le robot des différents sommets avec :

$$d(m,i) = \sqrt{(x_r - x_s(m,i))^2 + (y_r - y_s(m,i))^2} \quad (1.6)$$

P : l'angle que fait le robot avec l'objectif.

B(m,i) : l'angle que fait le robot avec différents sommets.

m : numéro d'obstacle.

i : numéro du sommet.

$$\sin(A(m, i)) = L(m, i)/d(m, i) \quad (1.7)$$

$$L(m, i) = \sin(A(m, i)) \cdot d(m, i) \quad (1.8)$$

En remplaçant $d(m,i)$, on aura :

$$L(m, i) = \sin(A(m, i)) \cdot \sqrt{(x_r - x_s(m, i))^2 + (y_r - y_s(m, i))^2} \quad (1.9)$$

Une fois les $L(m,i)$ calculés, on procède aux choix de la manière suivante:

- A. On choisit la valeur maximale des $L(m,i)$ correspondant au sommet se trouvant au-dessus de la droite robot objectif, ainsi que la valeur maximale des $L(m,i)$ correspondant aux sommets se trouvant au-dessous de la droite robot objectif ($L(m,i)_1$, $L(m,i)_2$).
- B. ensuite, on choisit la valeur minimale des deux valeurs maximales des $L(m,i)$ choisit auparavant.

$$L_{\min} = \min[L(m,i)_1, L(m,i)_2] \quad (1.10)$$

La valeur L_{\min} choisit correspond au sommet minimum, qui correspond à son tour au but partiel.

Le robot atteint le but partiel, et refait les mêmes procédures en considérant les coordonnées du but partiel celles du robot.

En considérant toujours l'exemple de la figure 1.2, la trajectoire obtenue est montrée dans la figure 1.5.

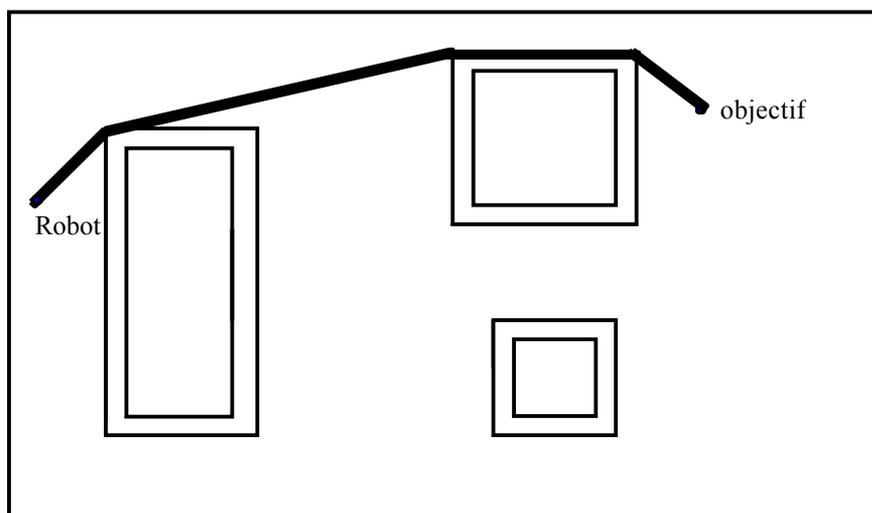


Figure 1.5 : trajectoire obtenue après calcul pour l'exemple de la figure 1.2

1.1.3.5. Algorithmes

- 1) Entrée : x_r, y_r : coordonnées du robot.
 x_{objet}, y_{objet} : coordonnées de l'objectif.
- 2) Limitation des sommets des obstacles connus.
- 3) Calcul du sommet le plus proche spp_i ($i = 1 \dots n$), avec: $n =$ nombre de sommets.
- 4)
 - a. Si l'obstacle n n'est pas sur la ligne directe à l'objectif et $i \neq n$, alors éliminer le spp_i et refaire 3.
 - b. Si $i = n$, et la ligne directe à l'objectif n'est interrompue par aucun obstacle, aller à 6.
- 5) Calcul du but partiel et le mettre dans la liste des solutions.
$$\begin{array}{l} x_r \longleftarrow x_{bp_i} \longrightarrow \\ y_r \longleftarrow y_{bp_i} \longrightarrow \end{array} \quad \text{aller à 2}$$

$i =$ nombre de but partiel
- 6) Exécuter le chemin au but final en passant par les buts partiels.
- 7) Atteindre l'objectif et terminer avec succès.

1.2 Localisation en robotique mobile : techniques et capteurs utilisés

Quelque soit le domaine d'application pour lequel il est destiné, un robot mobile, pour être utilisable, doit comporter un système permettant un certain niveau d'autonomie dans la localisation et la navigation. Pour schématiser, il doit être capable de répondre à trois types de questions : « où suis-je ? », « où vais-je ? » et « comment y aller ? ». La première question soulève le problème de la localisation. Les deux autres sont liées à la planification de trajectoire et à la navigation proprement dite. La bonne exécution des deux dernières tâches est fortement liée à la première.

Cette étude se restreint aux robots mobiles navigant sur un plan (2-D), ce qui englobe une très large partie des systèmes existants. Localiser le robot revient alors à déterminer trois paramètres : deux coordonnées cartésiennes pour la position et un angle pour l'orientation. De façon plus formelle, la tâche de localisation consiste à calculer la transformation de passage d'un repère lié au robot à un repère lié à l'environnement figure 1.6.

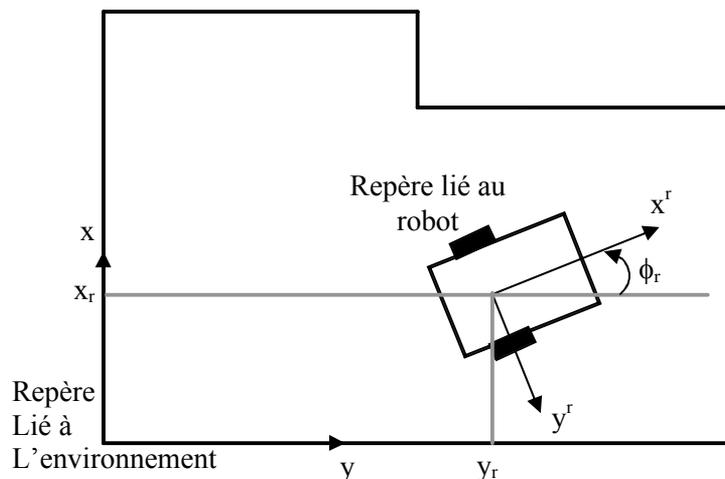


Figure 1.6 : formalisme du problème de localisation d'un robot mobile

Plusieurs techniques et méthodes ont été développées pour assurer la connaissance exacte et de façon autonome de la position d'un robot mobile dans son environnement. A ce jour, ces techniques peuvent être regroupées en deux catégories principales : les méthodes de localisation relatives et les méthodes de localisation absolue.

1.2.1 Localisation relative

La position du robot est calculée en incrémentant sa position précédente de la variation mesurée grâce à des capteurs proprioceptifs. Les deux principales méthodes de localisation relative sont la localisation grâce à l'odométrie et la localisation inertielle.

1.2.2 Localisation absolue

La position est calculée par rapport à des repères fixes grâce à des capteurs extéroceptifs. Ceci requiert souvent la connaissance de l'environnement. Les différentes techniques peuvent être distinguées par la nature des repères utilisés ou par la méthode de calcul. Selon la nature des repères utilisés, les approches les plus connues sont la localisation grâce aux compas magnétiques, la localisation grâce à des repères actifs, la localisation grâce à des repères passifs et la localisation basée sur le modèle. Selon les techniques de calcul utilisées on distingue, entre autres, les méthodes basées sur la trilatération (ou multilatération), les méthodes basées sur la triangulation.

En général, les techniques de localisation absolue découlent de deux catégories de base :

- Approches utilisant les balises.
- Approches de mise en correspondance.

1.2.2.1 Localisation par balises actives

Des balises sont disposées à des emplacements connus de l'environnement de travail. Elles sont facilement détectées par le robot et avec un faible coût de calcul. La base mobile à localiser est dotée d'un émetteur et les balises de récepteurs. L'avantage d'un tel système est le taux d'échantillonnage élevé. Les inconvénients sont la difficulté à disposer les balises avec précision et le coût d'installation et de maintenance ([33], [12]). Le calcul de la position et de l'orientation est basé sur la trilatération ou la triangulation.

1.2.2.2 Localisation grâce à des repères passifs

On distingue deux types de repères passifs, les balises passives et les amers. Contrairement aux balises actives, les balises passives se contentent de réfléchir un signal provenant de l'équipement de mesure. Il peut s'agir par exemple de miroirs. Ces balises peuvent également comporter des informations plus évoluées (code-barres).

Elles doivent avoir une position connue dans l'environnement et doivent être facilement identifiables grâce à un contraste suffisant par rapport au « fond ».

On appelle amers des éléments distincts de l'environnement que le robot peut reconnaître aisément grâce à ses capteurs extéroceptifs. Ces amers peuvent être par exemple des formes géométriques (rectangles, lignes, cercles). Ce sont des éléments déjà existant dans l'environnement et qui ont une fonction autre que celle de localiser le robot. Ils présentent l'avantage de ne pas modifier l'environnement de travail ce qui est important dans une application comme l'assistance à des personnes dans un appartement par exemple. Cependant, leur détection et leur identification peuvent être plus difficiles et les risques d'ambiguïté ou de fausses détections plus élevés. Le plus souvent, le capteur utilisé est visuel.

L'avantage de ce type de repère par rapport aux balises actives est le faible coût. Leurs inconvénients sont la portée réduite et la difficulté d'identification.

1.2.2.3 Localisation par mise en correspondance

Dans cette technique, un plan ou modèle de l'espace de travail du robot est stocké en mémoire. A un instant donné, le robot perçoit et construit grâce à ces capteurs un plan local de son environnement. Le système effectue alors une mise en correspondance entre le plan local et la partie correspondante dans le modèle global. Si une telle mise en correspondance est trouvée, alors la position et l'orientation du robot peuvent être calculées.

Ces approches permettent de localiser le robot quelque soit la qualité de l'information reçue par les capteurs ; en comparant, les informations reçues par une vue panoramique des capteurs avec une cartographie de l'environnement, cette dernière est connue soit a priori [19], [13], [14], [15], soit construite au fur et a mesure de l'évolution du robot dans l'environnement qui peut-être subdivisé en deux types :

- Environnement statique : dans ce type d'environnement, tous les obstacles sont fixes, il n'y a pratiquement rien de mobile. Cette hypothèse simplifie beaucoup le travail de localisation, de cartographie et de navigation. La plupart des travaux en robotique mobile considèrent ce type d'environnement.

- Environnement dynamique : un environnement dynamique varie dans le temps. Il y a, par exemple, des obstacles qui se déplacent, des portes qui s'ouvrent et qui se ferment, ou encore des humains qui se déplacent.

Récemment, plusieurs techniques compétitives sur les méthodes de mise en correspondance sont apparues, et plus particulièrement, nous citons deux méthodes qui permettent de localiser le robot en présence du bruit :

- Localisation de Markov.
- Mise en correspondance des scrutations (scan-matching).

En terme probabilistique, la localisation est le process de déterminer la probabilité de trouver le robot à la position l , sachant les lectures des capteurs : $s_1, s_2, \dots, s_n = S^n$; et $a_1, a_2, \dots, a_n = A^n$ les lectures précédentes de position à partir des odomètres. En pratique, il est difficile de déterminer la lecture conjointe de tous les capteurs et les odomètres. En revanche, une approximation récursive est considérée :

$$p(l/S^n, A^n) = \alpha \cdot \int p(l/s_n, a_n, l') \cdot p(l'/S^{n-1}, A^{n-1}) \cdot dl' \quad (1.11)$$

Où :

l' : position précédente du robot.

α : facteur normalisé choisi de telle manière que :

$$p(l/S^n, A^n) \text{ tend vers } 1 \text{ pendant l'intervalle } [0, l].$$

Toutes les informations du passé du robot sont représentées par la distribution :

$$p(l'/S^{n-1}, A^{n-1}).$$

Chacune des deux méthodes de localisation doit être capable de traiter les cas suivants:

- La représentation de la distribution antérieure..
- Le calcul de la distribution postérieur $p(l/s_n, a_n, l')$

Pour représenter les probabilités antérieures, la localisation de Markov utilise soit des grilles [10], [13], soit un graphe topologique pour couvrir l'espace de positions du robot, et ensuite garder la probabilité de chaque élément de cet espace.

La mise en correspondance des scrutations (scan matching) [19], [23], [24], [9], utilise une simple distribution gaussienne. Malgré la divergence dans la représentation de la distribution antérieure, les deux méthodes utilisent les mêmes techniques pour calculer la distribution postérieure :

- Prédire la position l du robot et son incertitude, à partir de la position précédente l' et l'information des odomètres.
- Mettre à jour la position du robot (incertitude) en utilisant l'information du capteur correspondante à celle de la carte de l'environnement.

La première étape augmente l'incertitude dans la position du robot, alors que la seconde la diminue. L'étape de prédiction est modélisée par une probabilité conditionnelle $p(l/a_n, l')$ qui signifie que la probabilité de l'action a_n , exécutée à la position l' , mène le robot à la position l .

La position est calculée par :
$$p(l) \propto \int p(l/a_n, l') \cdot p(l') \cdot dl' \quad (1.12)$$

Dans les algorithmes considérés ici, les erreurs possibles des odomètres (c.à.d. $p(l/a_n, l')$) sont modélisées par une distribution normale.

Dans l'étape de mise à jour, la nouvelle position du robot suit la formule de Bayes:

$$p(l/s_n) = \alpha \cdot p(s_n/l) \cdot p(l) \quad (1.13)$$

Le modèle du capteur $p(s_n/l)$ représente la probabilité d'avoir l'information s_n du capteur sachant que le robot est à la position l .

1.2.2.3.1 La méthode de mise en correspondance des scrutations (MCS)

La méthode (MCS) est la manière d'effectuer des translations et des rotations sur les scrutations (lecture des informations des capteurs d'une vue panoramique) dans le but d'avoir le maximum de ressemblance avec une cartographie connue à priori basés sur des scrutations, comme il est présenté dans la figure I.4.

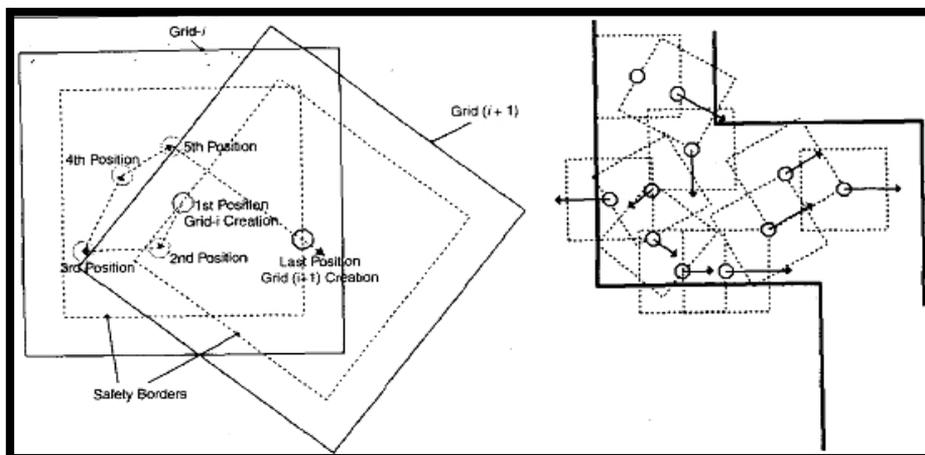


Figure 1.7 : Transfert d'un modèle local à un autre [1]

La plupart des méthodes de MCS présument une position initiale estimée proche de la position réelle, généralement prise par les odomètres, afin de déduire l'espace de recherche.

La position du robot et sa mise à jour à partir de la MCS suivent la distribution Gaussienne, ceci à l'avantage à ce que la position du robot sera calculée avec une grande précision et l'étape de mise à jour peut-être traiter par un filtre de Kalman [34].

Il existe un très grand nombre de méthodes de MCS, nous citerons en particulier:

1. Mise en correspondance point-segment

Cox en 1991 [19] a essayé de mettre en correspondance des points pris par scrutation des capteurs, avec des segments du modèle de l'environnement connu à priori, en se basant sur le voisinage le plus proche. Il effectue ensuite des translations et des rotations afin de réduire le carré de la distance entre les points de scrutations et leurs segments cibles.

Cette approche est aussi employée à partir des mesures issues d'une ceinture de capteurs à ultrasons pour la localisation en cours de mouvement développée dans le cadre du projet V.A.H.M [17]. Dans ces travaux, le mode d'association d'une mesure au segment le plus proche est réalisé en définissant une ellipse d'incertitude pour chaque segment, et en cherchant pour un point donné son appartenance à une ellipse.

Il est clair que cette méthode ne pourra être efficace qu'avec des environnements présentant une structure polygonale.

2. Utilisation de la fonction de corrélation (c c f)

Cette approche utilise une représentation stochastique (histogramme) des deux scrutations (référence et courante), la mise en correspondance est résolue par la recherche du maximum d'une fonction de corrélation.

Là aussi la méthode présente un intérêt particulier qu'avec des environnements ayant une structure polygonale [23].

3. Mise en correspondance point par point

Ici la mise en correspondance se fait point par point, c.à.d un point de la scrutation de référence est mis en correspondance avec un point de la scrutation courante. Cette technique, utilise un algorithme connu sous le nom « idc » (iterative dual correspondance) applicable sur n'importe quelle structure d'environnement (polygonale ou non polygonale) [23].

4. Mise en correspondance segment par segment

Dans les cartes composées de segments de droites, le processus de mise en correspondance entre le modèle local et le modèle global consiste à rechercher la similarité entre les segments de droite suivant trois critères : orientation, position et longueur. Si plusieurs segments passent les trois tests, le plus long d'entre eux est choisi. La position du robot est corrigée dans une direction perpendiculaire au segment identifié et l'orientation est corrigée en fonction de la différence d'orientation entre les segments détectés et modélisés [15].

5. Mise en correspondance des scrutations combinées

Gutman et al en 1996, ont testé les trois méthodes cités précédemment dans un environnement réel par la plate-forme expérimentale AMOS (Autonomous Mobile System), ces tests ont montré que pour des structures polygonales la méthode de Cox est plus rapide avec une précision égale à celle de la c c f alors que pour les structures non polygonales, il est évident que la seule méthode qui peut donner des résultats directement exploitables c'est la méthode de mise en correspondance point par point. En partant de ces observations Gutman et al ont

développé un algorithme qui combine la méthode de Cox et la méthode de mise en correspondance point par point [23].

1.2.2.3.2 La méthode de MARKOV

Parmi les différentes variantes de Markov, on cite les méthodes qui utilisent les grilles d'occupation. Cette technique a été développée pour la première fois en 1985 par Elfes et Moravec [21], pour modéliser l'environnement et représenté les obstacles. Cette méthode a été utilisée principalement comme outil de base pour l'évitement des obstacles dans la robotique mobile.

Moravec et Elfes modélisent l'environnement à partir de la position courante du robot et les informations reçues des capteurs, Burgard et al [10] utilisent le même principe des grilles d'occupation en inversant la procédure, c.à.d connaissant le modèle de l'environnement et l'information issue des capteurs, ils estiment la position du robot. La méthode de localisation consiste à trouver la meilleure correspondance entre deux modèles. Le premier, représentant l'environnement partiel ou dans sa totalité, est appelé carte globale. Le second, correspondant à la partie de l'environnement perçue par le robot depuis sa position, est appelé carte locale.

Les cartes globale et locale sont constituées de cellules dont les valeurs indiquent la présence ou l'absence d'un obstacle. Pour ce faire Elfes affecte des valeurs positives aux cellules représentant l'obstacle et des valeurs négatives aux cellules correspondant à l'espace libre. En superposant les deux grilles, le produit de deux cellules superposées indique s'il y a correspondance ou non, puisque le produit de deux cellules identiques est positif et qu'il est négatif quand elles sont de signe opposé. Ce produit est calculé sur toute la grille et la somme de tous ces produits est évaluée. Ainsi, en calculant cette somme pour différentes positions et orientations de la carte locale par rapport à la carte globale, l'emplacement qui fournit la somme maximale et donc qui donne la meilleure correspondance entre les deux grilles permet de localiser le robot.

Pour corriger les erreurs odométriques Yamauchi [44] à développé une méthode de mise en correspondance qui nécessite non pas une carte globale mais deux cartes locales créés à deux instants différents. Ces cartes sont superposées pour calculer un facteur de correspondance M . Ce facteur augmente à chaque fois que deux cellules identiques sont superposées.

Une autre méthode à base de grilles utilise une représentation topologique pour modéliser l'environnement [18], chaque pièce est représentée par un nœud, et le couloir par un arc. Pour déterminer la pièce de travail au cour de la navigation, le robot construit de nouvelles grilles. En comparant les paramètres de la carte topologique créent au début à ceux extrait des grilles construites, la région de travail dans laquelle le robot se trouve est déterminée.

1.2.3 Méthodes de calcul de la position et de l'orientation

1.2.3.1 La trilatération

Le calcul de la position se fait grâce à la connaissance de la distance de la base à trois références de l'environnement au moins. Le temps de vol de l'onde émise peut par exemple être utilisée pour le calcul de ces distances. La figure 1.8 montre un exemple de calcul de la position du robot grâce à la connaissance des distances respectives aux balises B_1 , B_2 et B_3 .

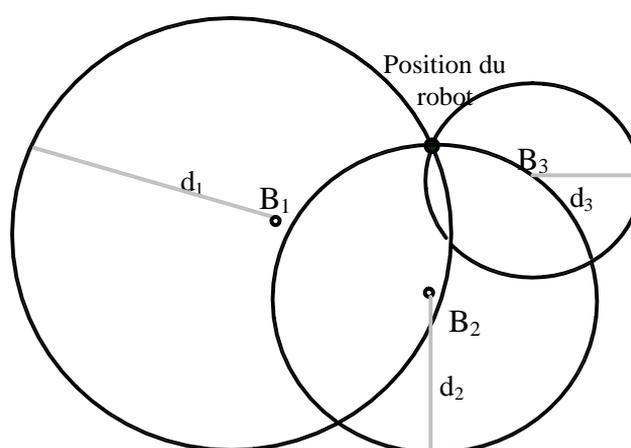


Figure 1.8 : la trilatération

1.2.3.2 La triangulation

Dans ce cas les mesures utilisées sont goniométriques. Les angles $\lambda_1, \lambda_2, \lambda_3, \dots$ avec lesquels sont vus les repères sont calculés (figure 1.9). Ces mesures peuvent par exemple être effectuées grâce à un capteur (récepteur ou émetteur) rotatif embarqué sur le robot ou encore grâce à une caméra panoramique.

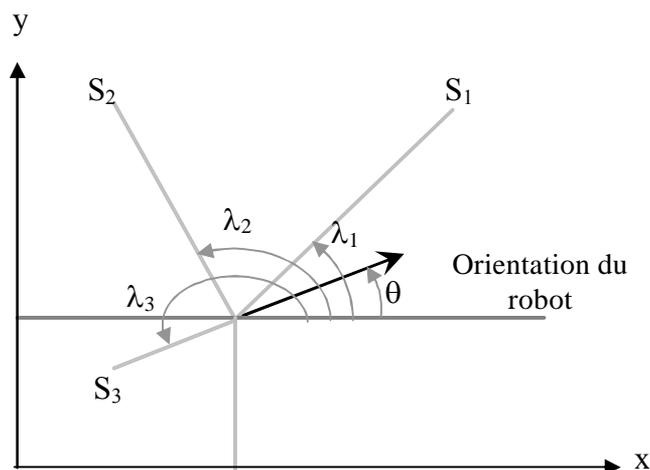


Figure 1.9 : principe de la triangulation

Un exemple de calcul de la position à partir de ces angles avec un minimum de trois mesures est représenté en figure 1.10. Chaque couple de mesures contraint la position du robot à appartenir à un arc de cercle défini grâce à la différence ω des angles de vue de chaque amer. La position finale retenue correspond à l'intersection de tous les arcs de cercles calculés grâce à tous les couples possibles de mesures.

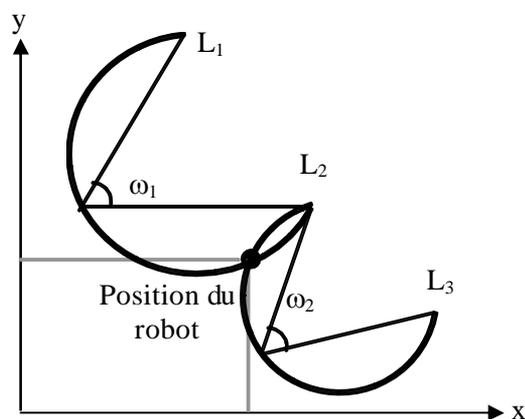


Figure 1.10: Calcul de la position à partir de données goniométriques

Un algorithme de calcul de la pose par triangulation est présenté dans [4]. Il est basé sur une représentation des amers par des nombres complexes ce qui permet d'obtenir un système d'équations linéaires. La complexité de l'algorithme est linéairement proportionnelle au nombre de mesures.

1.2.3.3 Localisation grâce à deux points

Une autre technique de calcul est citée dans [26]. Une caméra et un plan laser sont utilisés. Une coupe de l'environnement est effectuée grâce au laser. Deux points $R(x_r, y_r)$ et $L(x_l, y_l)$ dans le repère du robot R_r , sont extraits de cette coupe. Un repère R_{ij} est défini grâce à ces deux points (figure 1.11). La matrice de passage de R_{ij} à R_r s'écrit:

$${}_{R_r}T_{R_{ij}} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & -x_l \cos\theta - y_l \sin\theta \\ -\sin\theta & \cos\theta & 0 & x_l \cos\theta - y_l \sin\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.14)$$

L'orientation et la position sont alors obtenues comme suit :

$$\begin{cases} \theta = \arctan\left(\frac{y_r - y_l}{x_r - x_l}\right) & \text{si } x_l < x_r \\ \theta = -\frac{\pi}{2} & \text{si } x_l = x_r \\ \theta = \arctan\left(\frac{y_r - y_l}{x_r - x_l}\right) - \pi & \text{si } x_l > x_r \end{cases} \quad (1.15)$$

et

$$\begin{cases} i = -x_l \cos\theta - y_l \sin\theta \\ j = x_l \cos\theta - y_l \sin\theta \end{cases} \quad (1.16)$$

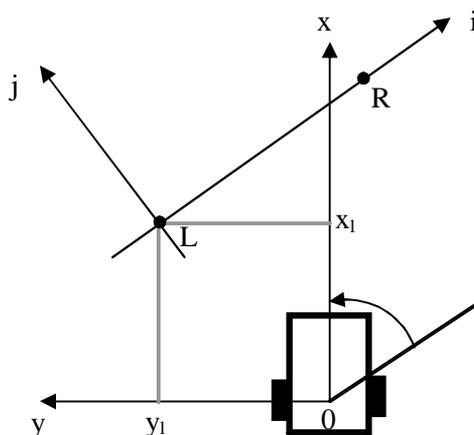


Figure 1.11 : localisation grâce à deux points

1.2.4 Localisation en environnement intérieur

Nous nous intéressons à présent au problème de localisation dans un environnement intérieur. Ce problème est apparu suite au développement de projets de robots mobiles connus sous le nom de robots de service. Les premiers robots mobiles étaient appelés AGVs (Automatic Guided Vehicles). Ce sont des systèmes destinés presque exclusivement à des tâches industrielles répétitives telles que le transport de charges dans un environnement structuré en suivant un chemin unique. Cette tâche était exécutée grâce à des bandes réfléchissantes, à une ligne blanche tracée sur le sol ou grâce à des câbles électromagnétiques enterrés. Depuis, les robots mobiles sont de plus en plus développés pour des applications telles que l'assistance à une personne handicapée en appartement [27], le nettoyage et l'entretien, ou encore le guidage d'une visite dans un musée. Ces applications ont en commun un environnement que l'on peut qualifier de semi-structuré.

S'il est permis d'équiper un environnement industriel par des balises ou des amers artificiels, ceci est difficilement envisageable dans un environnement domestique ou des bureaux. Il paraît clairement que la localisation basée sur un plan ou un modèle et, dans une moindre mesure, les méthodes utilisant des amers naturels soient les mieux adaptées à ce type de localisation. Il faut noter que dans ce contexte, la limite entre les deux techniques est assez floue. Un aspect important de la localisation en intérieur est la manière de représenter l'environnement, autrement dit, la structure du modèle la plus

adéquate en terme de richesse d'information et de concision. A la lumière des travaux étudiés, deux grandes représentations émergent. Il s'agit des modèles topologiques et des modèles métriques (à primitives géométriques ou avec des grilles d'occupation).

L'approche de localisation et le type du modèle sont fortement liés. Quelque soit l'approche utilisée, la localisation dans le contexte d'un environnement interne peut être divisée en deux problèmes que l'on désignera par localisation incrémentale et localisation globale. La localisation incrémentale suppose l'existence à tout moment d'une estimation même grossière de la position et de l'orientation du robot ce qui permet de faciliter la tâche de mise en correspondance avec le modèle. Dans certains cas aucune estimation n'est disponible. Le problème de localisation, dit global, est alors plus complexe à cause des symétries et des ambiguïtés.

Dans les sections suivantes nous présentons les trois principales représentations de l'environnement ainsi que les approches de localisation associées à chacune.

1.25 Différentes représentations d'environnement

Il existe plusieurs façons de représenter l'environnement. Dans [43] deux grandes représentations sont distinguées. Il s'agit des grilles d'occupation et des cartes topologiques.

Cependant cette distinction ne fait pas apparaître les modèles à primitives géométriques ([37], [35], [36],...). On peut donc regrouper les façons de représenter l'environnement en deux grandes familles : les représentations métrique avec un ensemble de primitives géométriques ou avec des grilles d'occupation composées de cellules et les représentations topologiques (non métriques). Le modèle peut être pré-établi et stocké en mémoire dans certains cas. Dans d'autres cas il peut être inexistant au départ. C'est le robot qui le construit grâce à ses capteurs par exploration.

1.2.5.1 Représentation topologique

L'environnement est représenté par un graphe constitué de places reliées par des connecteurs. Selon la définition donnée dans [11], une place est un espace qui peut être soit topologique soit fonctionnel. Par exemple, une pièce ou un corridor sont des unités topologiques, une imprimante peut représenter une unité fonctionnelle. Des exemples de connecteurs entre les places sont les portes ou les escaliers. L'avantage d'une telle

représentation est la concision et la facilité de construction d'un modèle puisque elle ne nécessite pas une connaissance très pointue de l'environnement.

Pour se localiser, le robot a seulement besoin de savoir sur quel connecteur il se déplace en quittant une place. Cependant il doit être capable de distinguer facilement les places. Dans [Ulrich00], le capteur utilisé est une caméra. Des images couleur sont prises à une fréquence de 1Hz et libellées avec l'endroit où elles ont été prises. Les vues sont comparées grâce à des histogrammes à une dimension. Un exemple de modèle topologique est donné en figure (1.12).

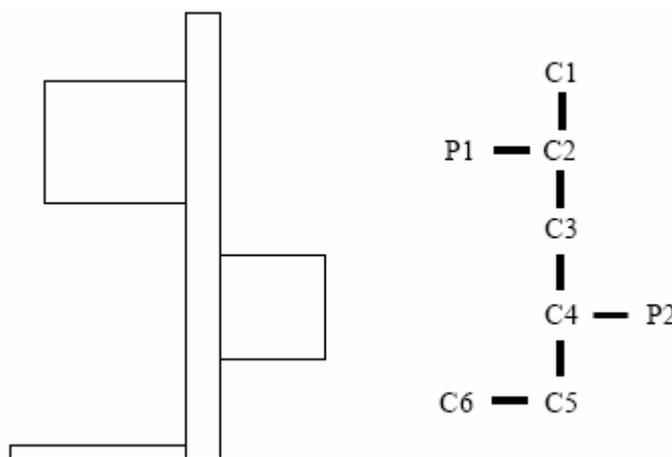


Figure 1.12: Exemple de représentation topologique d'un environnement constitué de couloir (C) et de pièces (P)

1.2.5.2 Représentation géométrique

L'idée de départ est qu'un environnement semi-structuré est constitué de primitives de base détectables par les capteurs extéroceptifs tels que les caméras, les capteurs à ultrasons ou les télémètres laser. Les exemples de primitives les plus courants sont les lignes, les coins, les plans ou les points. Ces primitives ont des positions et des dispositions connues et peuvent être paramétrées par des attributs tels que la longueur, la couleur, la texture, etc.

On peut citer plusieurs travaux utilisant des modèles à primitives géométriques. Dans [11] le modèle de l'environnement est un plan polygonal. Un capteur laser est utilisé pour détecter des lignes et réaliser un appariement avec le modèle. Un principe similaire est utilisé dans [29] avec des capteurs à ultrasons. Dans [31] les arêtes et coins

de l'environnement détectés grâce à un sonar sont considérés comme amers pour la localisation. Horn [28] extrait des facettes planes à partir d'une image laser.

Une telle représentation présente deux avantages. D'une part, l'utilisation de ce type de primitives permet facilement d'incorporer l'incertitude liées aux capteurs ou au modèle. D'autre part, elle est particulièrement adaptée aux opérations de changement de repères.

1.2.5.3 Les représentations en grilles d'occupation

L'approche de construction d'une représentation de l'espace sous forme d'une grille d'occupation à l'aide des capteurs télémétriques apparaît comme un choix idéal pour atteindre l'objectif de modélisation en temps réel d'un environnement dynamique. La représentation en grille, proposée au départ par Moravec et Elfes, souffre malheureusement de certains inconvénients liés, d'une part, à sa nature même et, d'autre part, au type des capteurs employés. En effet, cette approche est surtout associée aux capteurs à ultrasons, ce n'est que récemment que cette représentation a été également construite à partir de capteur laser à balayage horizontal ayant des vitesses d'acquisition élevées.

De nombreux auteurs limitent l'utilisation de grilles d'occupation et des capteurs à ultrasons à la construction de l'espace libre autour du robot.

Grâce à un bitmap glissant, permettent une mémorisation à court terme des obstacles proches. Le terme « glissant » provient du fait que la carte suit les déplacements du robot : après chaque déplacement, les informations que l'on possède sur les cellules frontières de la grille sont perdues. Il convient donc de ne pas limiter la grille à la zone directement perceptible par les capteurs.

Le principal inconvénient des modèles par grilles, où par bitmap, est le volume de mémoire nécessaire pour coder les informations. Une étude intéressante est présentée par Schiele et Crowley [42], où les auteurs comparent différentes méthodes de localisation qui met en correspondance :

- Grille locale avec grille globale,
- Modèle local extrait des segments avec une grille globale,
- Grille locale avec un modèle global extrait des segments,
- Modèle local et global extrait des segments.

Schiele et Crowley ont conclu aussi que les résultats en utilisant les grilles sont comparables et même plus précises que celles obtenues dans les travaux précédents qui utilisent le modèle paramétrique des segments extraient directement des données sonores.

1.2.6 Types de capteurs

Tout robot mobile autonome a besoin de moyens de perception pour lui permettre d'appréhender son environnement et de pouvoir y fonctionner en toute sécurité. Ces fonctions de perception permettent d'apporter une information pour aider à prendre une décision ou encore permettre d'agir efficacement sur son milieu homogène ou hétérogène.

En robotique mobile [40], les capteurs utilisés pour la localisation sont présentés dans deux parties, suivant qu'ils sont de types proprioceptifs, employés pour la localisation relative en estimant la distance parcourue par le robot, ou extéroceptifs, employés pour la localisation absolue à partir de la perception de l'environnement extérieur au robot.

1.2.6.1 Les capteurs proprioceptifs

Dans le cas des robots mobiles évoluant dans des environnements d'intérieurs, la localisation est souvent obtenue à partir de mesures du déplacement des roues fournies par des codeurs incrémentaux (odométrie). Dans le cas des robots évoluant en extérieur, le processus de localisation utilise des mesures fournies par des capteurs inertiels.

1.2.6.1.1 L'odométrie

C'est l'un des systèmes les plus utilisés en robotique mobile car il présente beaucoup d'avantages comme le coût financier, un très haut niveau d'échantillonnage de la mesure, une très bonne précision à court terme et une très grande facilité de mise en œuvre ([19], [26], [6]). L'idée fondamentale de ce système est l'intégration de l'incrément de la position, calculé grâce à des encodeurs montés sur les roues, par rapport au temps. Les déplacements dU_g et dU_d des roues droite et gauche permettent de calculer la variation de l'orientation ainsi que la variation de la position dU entre l'état n et l'état $n+1$ (figure 1.13).

Il en résulte malheureusement une accumulation non bornée de l'erreur. L'erreur en orientation induit d'importantes erreurs en position et les deux erreurs croissent en fonction de la distance parcourue. Les erreurs peuvent être regroupées en deux catégories selon leur source.

Les erreurs systématiques résultent des imperfections du modèle géométrique du robot (diamètres des roues différents, incertitude sur les dimensions des axes de la base,...). Les erreurs non systématiques résultent de l'interaction entre le robot et son environnement telle que les glissements ou les chocs qui ne sont pas pris en compte dans la mesure du mouvement effectué.

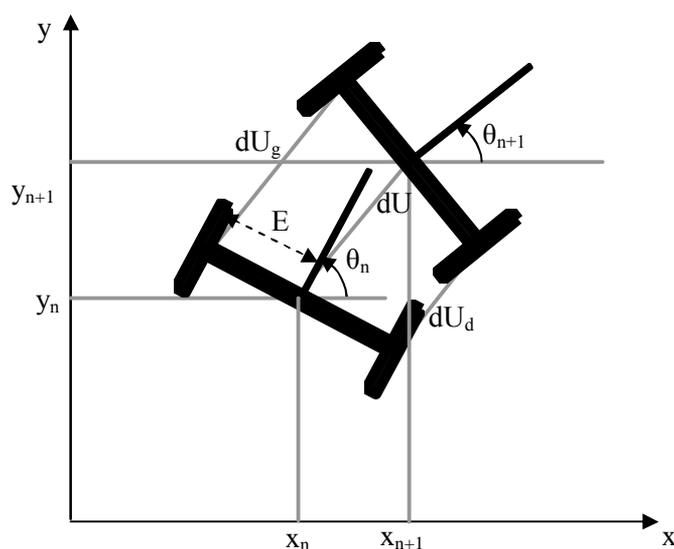


Figure 1.13 : calcul de la position grâce à l'odométrie

1.2.6.1.2 Les capteurs inertiels

Les capteurs inertiels utilisés en robotique mobile, notamment en robotique d'extérieur, sont de deux types : les gyromètres, qui fournissent une mesure de la vitesse instantanée de rotation, et les accéléromètres, qui fournissent une mesure de l'accélération instantanée en translation. L'association de ces deux types de capteurs constitue une centrale inertielle.

1.2.6.1.3 Capteurs à effet Doppler

Les capteurs à effet Doppler permettent une vitesse relative entre le véhicule et une surface (par exemple le sol) en utilisant l'effet de décalage en

fréquence entre une onde émise et le faisceau réfléchi sur la surface durant le déplacement. Les capteurs utilisent des ultrasons ou des ondes hyperfréquences et permettent d'estimer la vitesse avec une résolution d'environ 2 cm/s, pour des gammes de mesures comprises entre 0 et 20 m/s. Ces techniques présentent l'avantage d'être insensible au glissement des roues, mais leurs performances se dégradent rapidement en cas de mesure de faibles vitesses contrairement aux capteurs optiques à corrélation. Puisque la mesure s'effectue selon une direction, ces capteurs sont principalement utilisés pour estimer la vitesse d'avancement du véhicule.

1.2.6.1.4 Capteurs d'attitude

Les capteurs d'attitude sont très utilisés sur des robots se déplaçant en terrain accidenté, afin de mesurer le cap par rapport au nord magnétique, et les angles de roulis et de tangage. Les inclinomètres sont précis à l'arrêt, mais la précision se dégrade lorsque la mesure est acquise durant les mouvements.

1.2.6.1.5 Les capteurs optiques

Les capteurs optiques à corrélation exploitent des images du sol acquises généralement par une caméra CCD durant le déplacement du robot. En utilisant une technique de corrélation entre deux images consécutives de la séquence, il est possible de mesurer le mouvement apparent dans l'image, résultant de la projection du mouvement réel. Connaissant la position de la caméra par rapport au sol, il est alors possible de reconstruire une estimée du mouvement réel. Cette méthode est souvent appelée odométrie optique.

Le principal inconvénient de cette méthode réside dans la nécessité d'avoir des images du sol suffisamment texturées pour permettre un bon fonctionnement de l'algorithme de corrélation, elle est surtout adaptée pour la localisation d'un robot évoluant sur terrain naturel.

1.2.6.2 Les capteurs extéroceptifs

Les capteurs extéroceptifs, utilisé en robotique mobile délivrent une information qui caractérisent l'environnement, on peut les regrouper en deux classes: les capteurs passifs et les capteurs actifs.

1.2.6.2.1 Les capteurs passifs

L'énergie perçue est dans la majorité des cas la lumière mais elle peut aussi être constituée par la chaleur, par le son, ou par une autre forme d'énergie.

Les technologies des caméras sont suffisamment évoluées pour permettre d'interpréter une scène avec une bonne résolution. De même, le volume d'une caméra assure un encombrement minimal, ce qui permet de l'embarquer sur tout type de robot mobile.

Une caméra recueille sur une puce un plan image d'une scène qui peut être enrichie en analysant ses caractéristiques (perspective des objets, ombres, gradation des textures...). Par contre la distance ou la profondeur est déterminée par une association de plusieurs images décalées dans l'espace : la stéréovision.

En robotique mobile, les systèmes de vision ont été utilisés dans la navigation et la modélisation de l'environnement par fusion avec des ultrasons ou des télémètres laser. La localisation des robots mobiles est un domaine de recherche dans lequel ce capteur est très employé.

1.2.6.2.2 Les capteurs actifs

L'information sur l'environnement est portée par l'énergie réfléchi par les obstacles. Ce type de capteur englobe essentiellement les télémètres dont l'objectif consiste à déterminer la distance qui sépare le robot de son environnement. Les capteurs diffèrent par le type d'énergie employée et par la forme structurée ou non de l'émission.

Les principaux procédés couramment utilisés en robotique mobile sont : les ultrasons, les infrarouges, les lasers, et le GPS.

- Les ultrasons

Les capteurs à ultrasons permettent la mesure du temps de vol d'une onde ultrasonore entre son émission et sa réception après réflexion. A partir de ce temps et connaissant la vitesse de propagation de l'onde, la distance séparant le capteur de l'obstacle qu'il perçoit est directement disponible sans analyses complémentaires. Les acquisitions des mesures sont rapides, ils peuvent donc être utilisés pour les processus

qui demandent un traitement en temps réel comme l'évitement d'obstacles ou l'estimation de la position [5].

Parmi les caractéristiques de ces capteurs, la distance maximale mesurée peut aller de quelques centimètres jusqu'à quelques dizaines de mètres, ainsi que leur cône d'émission assez large (de 20° à 30°) qui ne permet pas de connaître précisément la position de l'obstacle dans le cône d'émission du capteur.

Les premiers travaux utilisant ce type de capteurs sont ceux de Crowley [13]. Il a utilisé une ceinture autour du robot afin de percevoir une grande partie de l'environnement (Figure 1.14). Ceci peut aussi être obtenu par un capteur tournant [20].

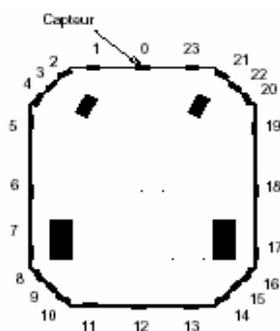


Figure 1.14 : répartition des capteurs ultrasons sur la ceinture du robot

- Les télémètres lasers

Le principe de mesurer une distance par un télémètre laser est le suivant :

- Lorsqu'il émet en mode continu, la distance est déterminée à partir du déphasage entre l'onde émise et l'onde réfléchiée par l'obstacle.
- Lorsqu'il émet en mode discret, la distance est proportionnelle au temps de vol de l'onde entre son émission et sa réception.

La distance maximale mesurable des lasers peut aller d'une dizaine de mètres jusqu'à quelques centaines de mètres.

La précision angulaire des systèmes laser est meilleure que celle des capteurs à ultrasons, ils sont cependant moins utilisés car plus coûteux, plus fragiles et plus délicats à mettre en œuvre.

- Les capteurs infrarouges

Les capteurs à infrarouges fonctionnent suivant le même principe que les capteurs laser et ultrasonores. Ils se composent d'un couple émetteur - récepteur.

Les capteurs à infrarouges sont d'un faible encombrement mais sensibles aux perturbations lumineuses ainsi qu'à la nature des surfaces réfléchissantes. Leurs portées n'étant que de quelques mètres, ils sont utilisés essentiellement pour détecter les obstacles proches.

- Le GPS

La navigation est le but initial du système GPS. Elle consiste à localiser des véhicules sur terre, sur la mer ou dans l'espace [8].

Le principe est de mesurer des distances d'une ou plusieurs stations au sol à plusieurs satellites simultanément et d'en déduire, à partir des coordonnées satellites soit la position des stations, soit la différence de coordonnées entre les stations.

Une technique consiste à mesurer au moins 4 pseudo distances sur 4 satellites à partir d'un seul récepteur. La durée d'observation est de quelques secondes. Le résultat est une position absolue, instantanée, en temps réel, avec une précision de 10 à 100 mètres selon que les dégradations sont actives ou non.

1.3 Conclusion

Notre objectif est la contribution au développement d'un fauteuil électrique autonome et plus précisément, l'établissement d'une méthode de localisation statique du robot. Pour cela nous avons présenté dans les sections précédentes, les approches de planifications de trajectoires existantes dans la littérature, nous avons ensuite, décrit la méthode choisie pour faire la simulation.

Nous avons abordé la problématique de la localisation en robotique mobile ainsi que les différentes approches et techniques (capteurs et représentations d'environnement) utilisées pour l'effectuer. Nous nous sommes ensuite focalisé sur la robotique dans un environnement intérieur.

Premièrement il ressort de cette étude l'aspect de complémentarité entre les deux grandes approches de localisation que sont la localisation relative et la localisation absolue. Selon l'étude faite par Schield et Crowley [42] aboutissant à la fin de leurs travaux à la conclusion suivante :

Les résultats de la mise en correspondance de modèle des grilles sont comparables et même plus précises que celles obtenues dans les travaux qui utilisent le modèle géométrique. Pour cela, on a choisit le modèle des grilles pour représenter

l'environnement d'évolution du robot. L'environnement est connu et de dimension relativement faible (quelques mètres, à une dizaine de mètres).

La localisation grâce à des grilles d'occupation, bien qu'étant discutable en terme de précision, présente l'avantage d'être très peu dépendante de la nature du capteur utilisé [25]. Plusieurs techniques de décompositions existent. Elles peuvent être regroupées en deux catégories, les méthodes exactes et les méthodes approchées [26]. Le calcul de localisation quant à lui se fait grâce à un regroupement d'informations capteurs. Plusieurs méthodes de mise à jour de l'état des cellules existent. Dans [5] une méthode simple permettant de réduire le temps de calcul de la mise à jour est présentée.

Le choix des capteurs ultrasonores pour positionner le robot utilisé comme aide aux handicapés moteur se justifie par leurs caractéristiques physiques et de coût. Le rôle du capteur est de percevoir les limites de l'espace libre entourant le robot, cela a souvent été réalisé au moyen de capteurs rotatifs de différentes natures : télémètre laser, caméra sur un miroir tournant, ultrasons, ou infrarouge. Ce type d'équipement rotatif s'adapte bien aux robots industriels, mais dans notre cas, la présence de la personne sur le fauteuil cache une partie du panorama et rend impossible l'exploitation de ce type de matériel. D'autres projets de robotique mobile utilisent une ceinture de capteurs à ultrasons fixes. Ce dispositif délivre une information de distance (limite de l'espace libre dans la direction de chaque capteur) directement exploitation pour la localisation mais peu précise. En effet, l'obstacle détecté se situe dans le cône d'émission de l'onde ultrasonore émise par le capteur mais il est impossible de savoir précisément dans quel axe. Par ailleurs, ces capteurs US sont légers, peu chers, de petit encombrement, et de faible consommation énergétique, nous les avons donc choisi car ces caractéristiques les rendent bien adaptés à la problématique de notre projet. Nous avons donc équipé la base mobile d'une ceinture de 12 capteurs répartis autour de celle-ci.

Les chapitres suivants concernent les choix effectués et les contributions dans chacune des étapes du localisateur énoncés ci-dessus.

CHAPITRE 2

MÉTHODE DE LOCALISATION UTILISÉE

Après avoir survolé les aspects de la localisation en robotique mobile puis de la localisation en environnement intérieur, nous nous focalisons à présent sur le problème de la localisation par mise en correspondance des grilles. Autrement dit, nous nous intéressons au cas où la représentation de l'environnement est basée sur les grilles d'occupation.

Pour cela, nous présentons un modèle du capteur à ultrasons basé sur les grilles d'occupation. Ce modèle est ensuite adapté pour l'utilisation d'une ceinture de 12 capteurs en tenant compte des superpositions de plusieurs mesures et des différentes orientations de chaque capteur. La grille ainsi obtenue représente l'espace perceptible par le robot depuis sa position (grille locale). L'environnement d'évolution du robot quant à lui est représenté par une grille décrivant les limites de l'espace libre (grille globale). Les deux grilles sont mises en correspondance pour déterminer la position et l'orientation du robot mobile dans son environnement.

2.1. Introduction

Le processus classique de localisation par mise en correspondance des grilles se présente comme une séquence de trois étapes qui sont :

➤ Acquisition de l'information capteur

Le robot saisit une vue panoramique de son environnement par des capteurs ultrasonores à partir de sa position courante afin de créer la grille locale.

➤ Mise en correspondance des observations avec le modèle

Trouver chaque grille locale perçue par la ceinture des capteurs ultrasonores la grille globale correspondante dans le modèle de l'environnement connu à priori.

- Calcul de la position et de l'orientation du robot

Se fait grâce aux correspondances établies dans l'étape précédente.

Deux types de connaissances a priori sont nécessaires pour réaliser les trois étapes. Il s'agit du modèle de l'environnement et le modèle du capteur. Chacune de ces étapes de localisation et chacune des connaissances a priori représente en soit un vaste domaine de recherche sur lesquelles il existent un grand nombre de travaux publiés. Dans ce qui suit nous présentons brièvement le rôle de chaque étape.

2.1.1 Modélisation de l'environnement

Nous avons abordé dans le chapitre 1, les représentations par grilles d'occupation de l'environnement. Nous avons vu que ce modèle peut être soit construit par l'humain et fourni au système soit construit grâce aux capteurs extéroceptifs du robot.

2.1.2 Modélisation du capteur

Les données observées par les capteurs ultrasonores sont représentées dans un espace à deux dimensions formant la grille locale.

Pour permettre la mise en correspondance entre le modèle perçu par les capteurs et le modèle connu à priori ou le calcul de la position et de l'orientation du robot mobile il faut modéliser les capteurs qui fournissent l'information sur l'environnement. Autrement dit, choisir une modélisation qui permette de faire la simulation.

Deux modèles des capteurs ultrasonores existent. On distingue en effet, le modèle idéal et le modèle réel. Le choix d'un modèle doit tenir compte de la qualité du capteur et des dimensions de l'environnement perçu.

2.2 Etapes de la localisation

2.2.1 Acquisition de l'information capteur

Le but de cette étape est de distinguer parmi la grande quantité d'informations que véhicule une vue d'une ceinture de capteurs à ultrasons, celles qui peuvent correspondre au modèle de l'environnement. Le capteur à ultrasons présente l'avantage

de donner directement une information sur la distance. En effet, l'angle d'ouverture de l'onde acoustique introduit un facteur d'incertitude concernant la direction dans laquelle se situe l'obstacle perçu. Pour une mesure donnée par le capteur, en considérant un espace à deux dimensions et en négligeant les incertitudes sur la mesure, la région correspondant aux positions possible de l'obstacle détecté prend la forme d'un arc centré sur le capteur. Les dimensions de l'objet sont aussi masquées par cette propriété du capteur à ultrasons. Ensuite, le modèle perçu par une ceinture de capteurs est enregistré dans une grille ou on affecte une valeur à chaque cellule de la grille.

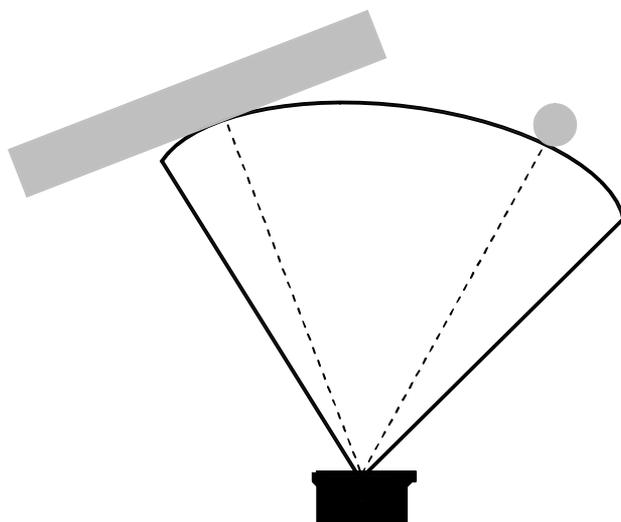


Figure 2.1 : Deux obstacles renvoyant la même mesure

2.2.2 Mise en correspondance des observations avec le modèle

Le modèle de l'environnement et le modèle créé par la ceinture des capteurs à ultrasons sont constitués de cellules dont les valeurs indiquent la présence ou l'absence d'un obstacle. Pour ce faire, des valeurs sont affectées aux cellules représentant les obstacles et d'autres valeurs aux cellules correspondant aux espaces libres. En superposant les deux grilles, un facteur de mise en correspondance augmente à chaque fois que deux cellules identiques sont superposées. Ce calcul est effectué pour toutes les translations et rotations possibles d'une grille par rapport à l'autre, l'emplacement qui fournit un facteur de mise en correspondance maximal et donc qui donne la meilleure correspondance entre les deux grilles permet de localiser le robot.

2.2.3 Calcul de l'orientation et de la position du robot

Après avoir établi une correspondance entre le modèle perçu et le modèle de l'environnement, il reste à calculer la position et l'orientation du robot mobile par rapport au repère du modèle de l'environnement.

2.3 Modélisation des capteurs à ultrasons

Tout système complexe observant ou inter-réagissant avec son environnement doit avoir la capacité de le percevoir, pour cela il doit disposer de capteurs fournissant les informations plus ou moins représentatives du monde, cela afin de se construire une représentation de ce monde. De nombreux capteurs existent pour remplir cette tâche de collecte d'informations et ainsi assurer le bon fonctionnement du système.

Modéliser un capteur de profondeur consiste à déterminer une fonction densité de probabilité est une distribution de Dirac:

$$P(r/d) = \begin{cases} 1 & \text{si } r=d \\ 0 & \text{si } r \neq d \end{cases} \quad (2.1)$$



Figure 2.2 : Modélisation d'un capteur idéal

En réalité, les capteurs à ultrasons émettent dans un cône de l'espace, l'incertitude sur les mesures intervient au niveau de la profondeur r , et aussi au niveau de l'angle θ par rapport à l'axe de tir, un modèle gaussien est bien adapté pour représenter son fonctionnement.

$$P(r/d, \theta) = (1/2\pi\sigma_r\sigma_\theta) \exp\left[-\frac{1}{2}\left(\frac{(r-d)^2}{\sigma_r^2} + \frac{\theta^2}{\sigma_\theta^2}\right)\right] \quad (2.2)$$

Où:

σ_r et σ_θ : résolution radiale et angulaire du capteur.

L'équation (2.2) modélise le fonctionnement du capteur en fonction de la distance et de l'angle par rapport à l'axe de tir.

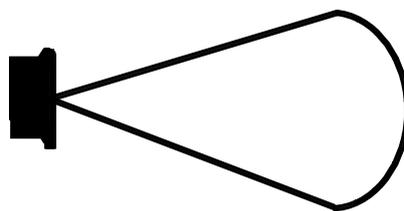


Figure 2.3 : Modélisation d'un capteur réel

Généralement, les capteurs à ultrasons possèdent une bonne résolution radiale (incertitude sur la distance mesurée) de l'ordre de quelques centimètres.

L'angle d'ouverture introduit un facteur d'incertitude concernant aussi la direction (résolution angulaire), dans la quel se situe l'obstacle perçu. Pour une mesure donnée par le capteur, en considérant un espace à deux dimensions et en négligeant l'incertitude sur la mesure, la région correspondante aux positions possibles de l'obstacle détecté prend la forme d'un arc (figure 2.3).

Dans le domaine de la perception, les capteurs à ultrason possèdent des défauts spécifiques qui restreint leur utilisation :

Les réflexions multiples (figure 2.4) : le signal ultrason peut-être réfléchi sur plusieurs surfaces d'incidence avant de retourner au capteur.

Ce phénomène fait paraître l'objet plus loin que sa position réelle.

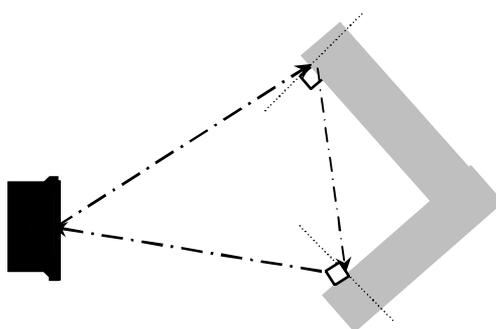


Figure 2.4 : Réflexion multiple

Forme des obstacles présents, leurs textures de la surface réfléchissante de l'objet. Les surfaces rugueuses retournent l'onde ultrasonore produite par le capteur

quel que soit l'angle d'incidence, contrairement aux surfaces lisses qui ne renvoient que les ondes ayant un angle d'incidence proche d'un angle droit (figure 2.5).

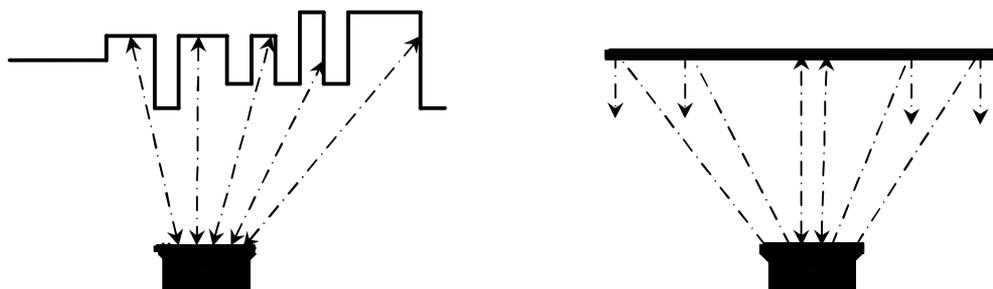


Figure 2.5 : Phénomène de spécularité

De même, une mesure incorrecte peut-être due à la diaphonie. Cette dernière intervient quand un capteur émet une onde ultrasonore, le cône émis est renvoyé vers le robot. Si un autre capteur se trouve en mode de réception, il interprétera l'onde émise par le capteur x et renvoyé par l'obstacle comme étant la sienne, et percevra un objet plus proche qu'il ne devrait (figure 2.6).

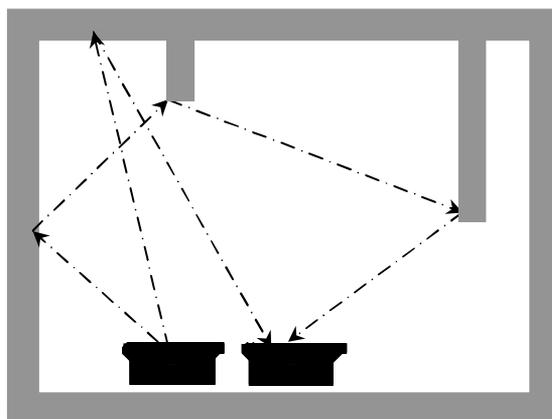


Figure 2.6 : Phénomène de diaphonie

Tous ces phénomènes sont des phénomènes parasites plus ou moins imprévisibles contrairement aux incertitudes angulaire et radiale qui sont toujours présentes dans la mesure.

2.4. Modélisation en grille

Une grille d'occupation est une représentation discrète 2D d'un espace navigable supposé plan. L'environnement est discrétisé en cellules ; à chaque cellule est attribuée une valeur décrivant son état.

En enregistrant le modèle idéal d'un capteur ultrasons dans une grille, seul les cellules situées sur l'axe acoustique du capteur à la distance mesurée sont modifiées (figure 2.7).

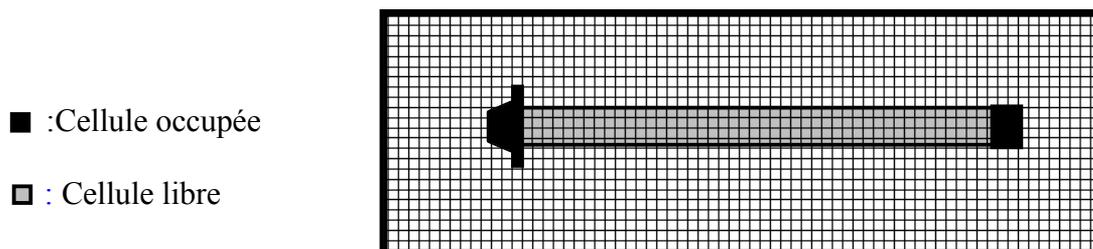


Figure 2.7 : Grille pour un capteur idéal

Ce modèle a été utilisé pour réaliser la localisation d'un robot mobile [18]. Les cellules de la grille représentant l'environnement sont initialisées à la valeur 0. Puis, au cours du déplacement du robot, les valeurs des cellules correspondant à l'espace libre sont décrémentées tandis que la valeur de la cellule représentant l'obstacle est incrémentée par 3, les valeurs des cellules restantes n'étant pas modifiées.

Ce modèle a été utilisé aussi pour effectuer des tâches d'évitement d'obstacles pour robot mobile [5].

L'inconvénient de ces modèles est qu'ils ignorent l'incertitude angulaire des capteurs à ultrasons. Un capteur à ultrason réel a un cône d'émission d'ouverture entre 20° à 30° , les réflexions retournées au capteur peuvent provenir d'objets situés dans la périphérie du cône, l'incertitude sur les mesures intervient donc sur la profondeur r , mais aussi sur l'angle θ par rapport à l'axe de tir.

Dans [21], le cône ultrasonore est décomposé en deux volumes de l'espace : l'un représentant l'espace libre, et l'autre représentant l'espace occupé en se basant sur les probabilités (Figure 2.8).

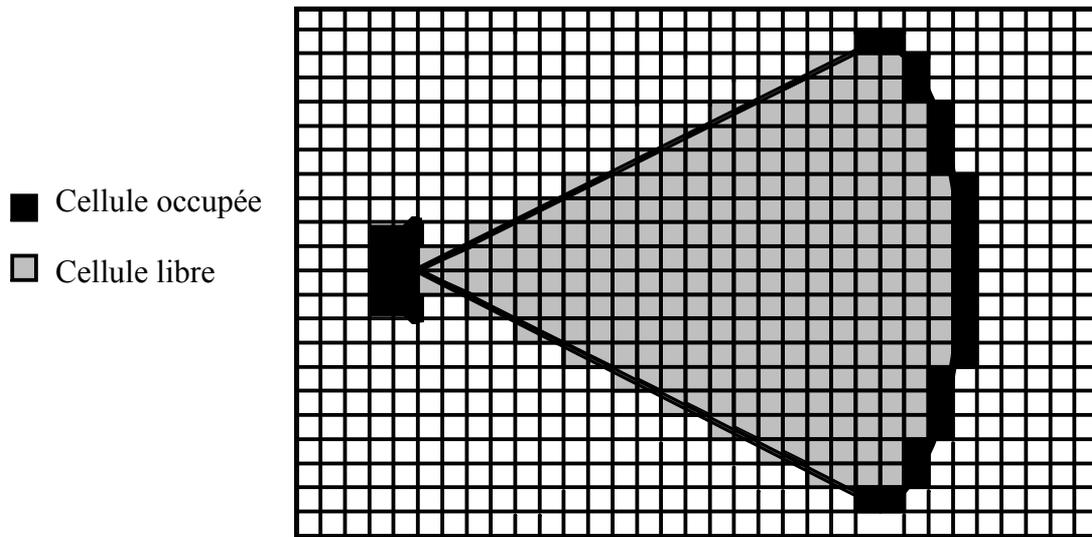


Figure 2.8 : Grille d'un capteur en prenant en compte l'ouverture angulaire

Région libre: inclue les points situés dans la région certainement vide ($r < R - \varepsilon$ et $\theta \leq \Omega/2$), qui ont une probabilité d'être libres définie par :

$$P_L = p[\text{point}(x, y, z) \text{ est libre}] = L_r \cdot L_a(\theta) \quad (2.3)$$

Avec :

$$L_r = \begin{cases} 1 - [(r - R_{\min}) / (R - \varepsilon - R_{\min})]^2 & \text{pour } r \in [R_{\min}, R - \varepsilon] \\ 0 & \text{ailleurs} \end{cases} \quad (2.4)$$

$$\text{Et } L_a = 1 - [2\theta / \Omega]^2 \quad \text{pour } \theta \in [-\Omega/2, \Omega/2] \quad (2.5)$$

r et θ désignent les coordonnées polaires d'un point dans un référentiel placé sur le capteur, R correspondant à la distance mesurée, R_{\min} représente la distance minimale détectable par le capteur, ε désigne l'incertitude sur la mesure et Ω décrit l'ouverture angulaire du capteur. Région occupée: inclue les points situés dans la région certainement occupée ($r \in [R - \varepsilon, R + \varepsilon]$ et $\theta \leq \Omega/2$), qui ont une probabilité d'être occupées définie par :

$$P_o = p[\text{point}(x, y, z) \text{ est occupée}] = O_r \cdot O_a(\theta) \quad (2.6)$$

avec :

$$O_r = \begin{cases} 1 - [(r-R)/\varepsilon]^2 & \text{pour } r \in [R - \varepsilon, R + \varepsilon] \\ 0 & \text{ailleurs} \end{cases} \quad (2.7)$$

$$\text{Et } O_a = 1 - [2\theta / \Omega]^2 \quad \text{pour } \theta \in [-\Omega/2, \Omega/2] \quad (2.8)$$

La figure 2.8 montre les distributions des capteurs à ultrason.

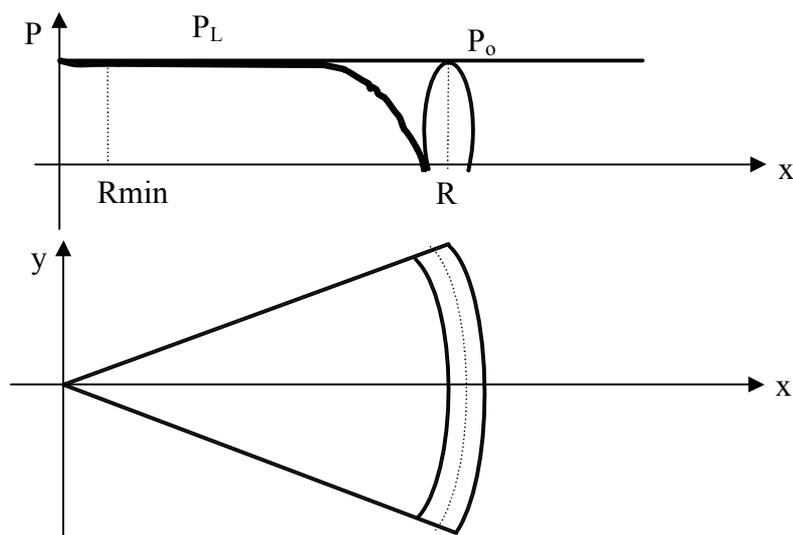


Figure 2.9 : distribution du capteur à ultrason

La figure 2.10 représente les probabilités en 3D.

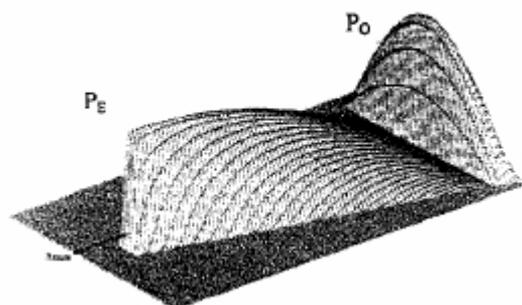


Figure 2.10 : Probabilité du capteur à ultrason en 3D [16]

A la fin, à chaque cellule est attribuée une valeur décrivant son état. Elfes [21] a proposé d'associer un vecteur à chaque cellule, afin de conserver plusieurs.

A l'initialisation, une valeur particulière est affectée à chaque cellule exprimant la connaissance courante que l'on a sur son état. Ainsi, dans le cas où l'environnement est totalement inconnu, une même valeur est attribuée à chaque cellule afin de représenter le manque de connaissance sur son état courant.

C'est le processus de mise à jour de la grille d'occupation. Pour cela, ELFES a proposé d'utiliser la formulation de Bayes qui prend en considération l'état courant de chaque cellule et le vecteur de mesures disponible pour calculer la nouvelle valeur de la probabilité d'occupation.

Dudek a proposé une méthode de regroupement des données ultrasoniques suivant un critère d'homogénéité afin de pouvoir extraire un segment à partir de chaque groupe de points par une approximation aux moindres carrés. La normale à chaque segment est calculée et son orientation vis-à-vis du repère de référence est rajoutée à l'état de chaque cellule couverte par ce segment.

La mise à jour de la grille se fait à chaque instant. Si une cellule est perçue dans le même état aux instants t et $t+1$. Alors l'état de cette cellule est confirmé : la probabilité d'occupation est décrétementée (libre) ou incrémentée (occupée). Si une cellule étiquetée obstacle à l'instant t est perçue comme étant libre à l'instant $t+1$, le test de specularité permet de vérifier si l'obstacle détecté auparavant en cette cellule est spéculaire ou non vis-à-vis de la position courante du capteur, si le résultat du test est positif, l'état de la cellule n'est pas modifié ; sinon sa probabilité d'occupation est décrétementée. Enfin si une cellule étiquetée libre à l'instant t est perçue comme étant obstacle à l'instant $t+1$, la probabilité d'occupation de la cellule est incrémentée.

Bulata propose d'effectuer la mise à jour des cellules de la grille d'occupation dans le sens capteur \longrightarrow obstacle, en décomposant le lobe en un ensemble de droites. Pour chaque droite, la probabilité des cellules est mise à jour dont les coordonnées vérifient l'équation de cette droite. Si le test de specularité est positif sur une cellule, toutes les autres situées après celle-ci par rapport au capteur ne seront pas modifiées.

Pagac[39] a utilisé la modélisation suivante. Le cône de mesure est composé d'un secteur représentant l'espace libre (contient m cellules) et d'un arc représentant la présence d'un obstacle (contient n cellules).

Les probabilités de chaque cellule du secteur sont données par : $\begin{cases} P_o=0 \\ P_L=1/m \end{cases}$ (2.9)

Les probabilités de chaque cellule de l'arc sont données par : $\begin{cases} P_o=1/n \\ P_L=0 \end{cases}$ (2.10)

Pour pouvoir ensuite utiliser les règles des probabilités pour la mise à jour de la carte, une cellule doit avoir la somme de ses probabilités égale à 1 ce qui n'est pas le cas. Il a interprété les données ultrasonores par la théorie de Dempster-Shafer, la somme des probabilités devient : $P_\emptyset + P_L + P_o + P_{L_o} = 1$ (2.11)

P_\emptyset : est la probabilité que la cellule ne soit ni vide ni occupé.

P_{L_o} : est la probabilité que la cellule soit vide ou occupé.

Pour chaque cellule, il est impossible qu'une cellule ne soit ni vide ni occupée (c.à.d $P_\emptyset = 0$), dans ces conditions il suffit de déterminer les probabilités P_L et P_o , puis la valeur de P_{L_o} est calculée par la relation : $P_{L_o} = 1 - P_L - P_o$. Ces trois valeurs suffisent pour mettre à jour les probabilités d'une cellule.

Courcelle [16] à créer la grille locale en affectant aux cellules que des valeurs entières, ensuite, il a modifié le modèle en affectant des valeurs réelles aux cellules, pour n_1 cellules de l'arc la valeur affectée à ces cellules est égale à $1/n_1$, et pour n_2 cellules du secteur la valeur affectée est égale à $-1/n_2$, et les cellules non explorées la valeur 0.

Dans notre travail, on va utiliser la modélisation suivante, les valeurs affectées sont : 0 pour indiquer que la zone n'est pas perçue par les capteurs ultrasonores, -1 pour indiquer l'espace libre (zone de secteur) et la valeur 1 pour indiquer la présence d'un obstacle (zone de l'arc).

Ces choix proviennent du fait que ce modèle à l'avantage d'éviter le traitement des mesures à l'aide de lois de probabilités lourdes à gérer. La zone occupée est une limite donnée par les capteurs à ultrasons, c'est pourquoi l'arc n'est réalisé que sur une épaisseur d'une ligne ; du point de vue du robot la zone à l'intérieur des obstacles, bien qu'occupée est considérée comme inconnue. Une seule carte est nécessaire pour représenter l'espace occupé et l'espace libre ; la différence se fait sur le signe de la valeur de la cellule.

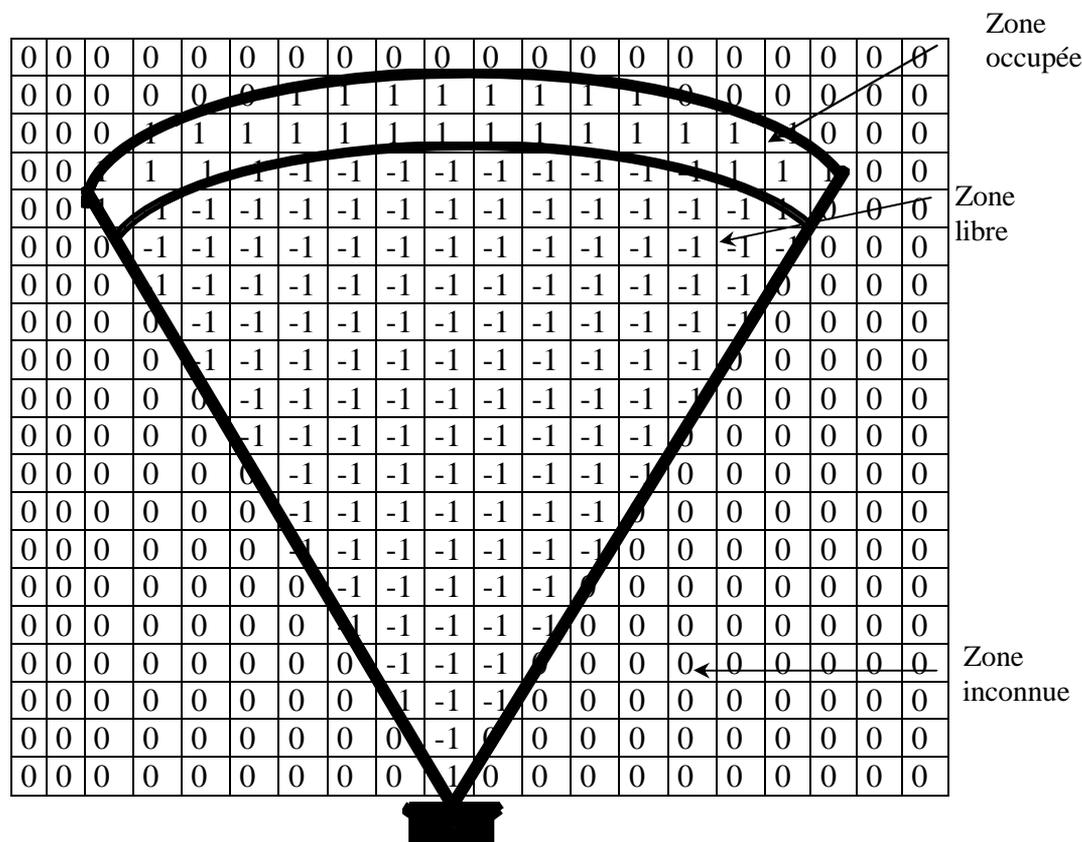


Figure 2.11 : les valeurs affectées aux cellules libres et occupées

2.5 Modélisation de l'espace navigable

2.5.1 Grille globale

La grille globale définit l'environnement dans lequel le robot va se mouvoir. Notre application étant destinée à un fauteuil roulant électrique pour des personnes physiquement handicapées, l'environnement correspondant à des pièces telles que celles rencontrées dans un appartement.

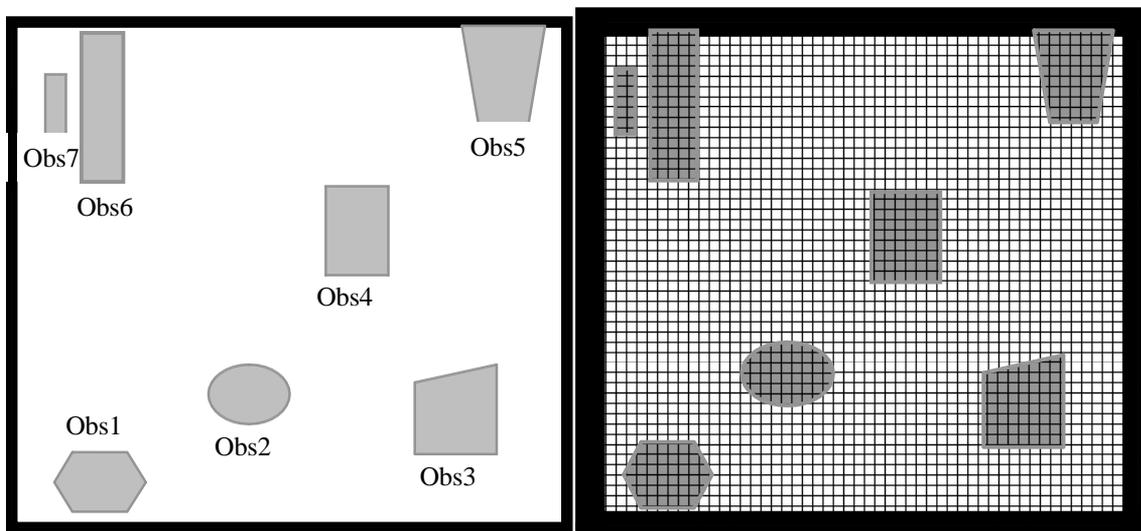
Pour la navigation ou la localisation, il a été démontré qu'un modèle 2D du sol est suffisant dans de très nombreux cas (sol plat). Les représentations géométriques ont souvent été employées pour la modélisation des environnements structurés (ou d'intérieur). Ces environnements comportent de nombreuses entités facilement modélisables par des facettes ou des segments de droite (arrêtes entre murs, facettes planes, etc..) et sont naturellement adaptés à une représentation polygonale. L'approche de construction d'une représentation de l'espace sous forme d'une grille d'occupation à

l'aide des capteurs télémétriques apparaît comme un choix idéal pour atteindre l'objectif de modélisation en temps réel d'un environnement dynamique.

Dans notre travail l'espace navigable est défini par une grille globale à 2 dimensions qui représente un environnement statique correspondant à des pièces telle que celles rencontrées dans un appartement (figure 2.12.a).

La grille globale est construite à partir d'un plan d'environnement décomposé en cellules, on attribue les valeurs $C_g = -1$ à toutes les cellules représentant le sol de la pièce (espace libre) et les portes, et on affecte les valeurs $C_g = 1$ à toutes les cellules recouvertes par des objets existant dans l'environnement (figure 2.12.b).

L'exemple de la figure 2.12.a représente une pièce composée de 7 objets :



-a- Exemple d'environnement

-b- Représentation en grille

Figure 2.12 : exemple d'un plan d'une pièce

2.5.2 Grille locale

A partir des mesures obtenues sur les 12 capteurs autour du robot, nous créons une grille locale d'occupation où les -1 représentent les zones vides, les 1 les zones occupées et les 0 les zones non déterminées. La zone occupée est une limite donnée par les capteurs US (Figure 2.13). Une fois la grille locale constituée, l'objectif est de rechercher la position et l'orientation dans lesquels elle se superposera au mieux avec la grille globale.

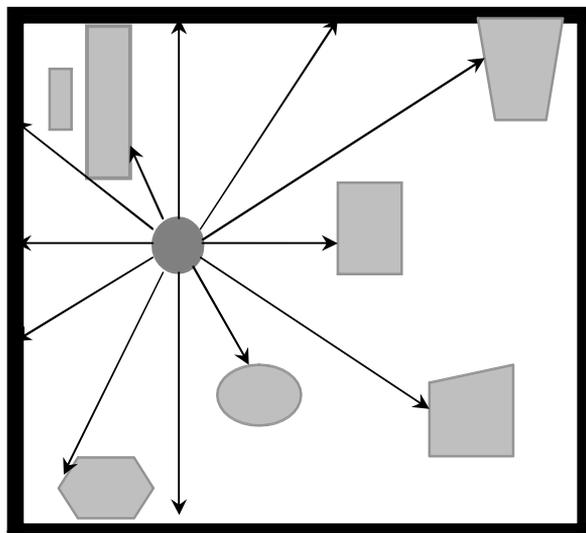


Figure 2.13 : la disposition des capteurs autour du robot et leurs lectures à un moment donné

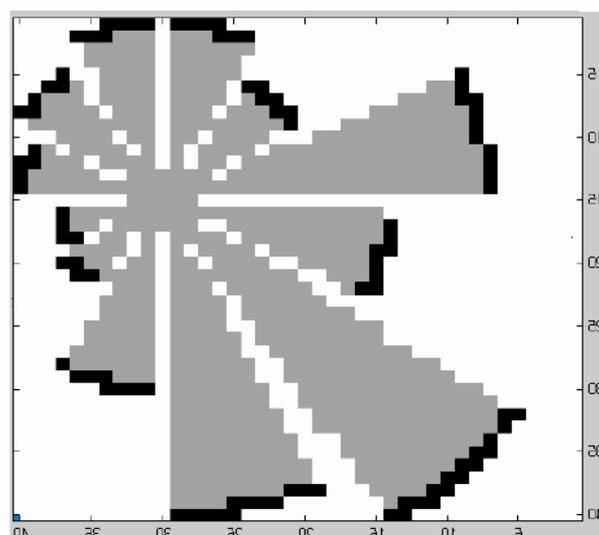


Figure 2.14 : exemple de la construction d'une grille locale

2.5.3 Représentation de l'espace occupé par le robot

L'espace occupé par le robot correspond à toutes les cellules recouvertes par la surface au sol du robot (Figure 2.15). Comme nous sommes certains que le robot n'est pas placé sur un obstacle, nous affectons la valeur $C_L = -1$ aux cellules correspondantes.

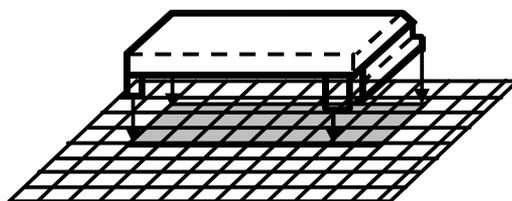


Figure 2.15 : Projection de l'espace occupé par le robot sur la grille

2.6 La méthode de mise en correspondance des grilles

Le but de la localisation est de déterminer la position et l'orientation du robot dans l'environnement. La première étape consiste à centrer le robot dans la grille locale, ensuite il faut la faire tourner par un angle qui est égal à l'orientation. La grille obtenue va subir des translations et des rotations afin d'avoir la meilleure coïncidence entre cette grille et la grille globale. En terme mathématique, on cherche la transformation F se décomposant en une translation T et une rotation R , qui permet de déterminer la cellule $C_G(k, l)$ de la grille globale M_G connaissant la cellule $C_L(i, j)$ de la grille locale M_L .

$$C_G(k, l) = F[C_L(i, j)] = T + R \cdot C_L(i, j) \quad (2.12)$$

Avec:

$T = [a, b]^T$: cellule dans la grille globale.

$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ -\sin\theta & -\cos\theta \end{bmatrix}$: Matrice de rotation.

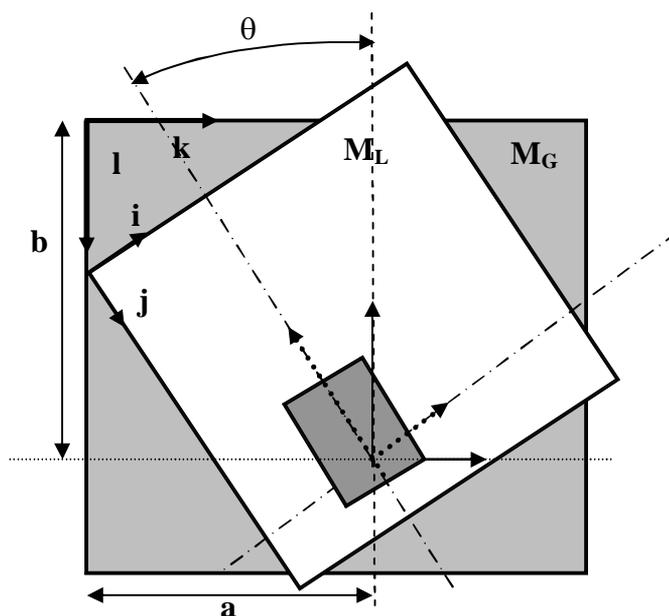


Figure 2.16 : Mise en correspondance des grilles

Les indices de la cellule de la grille globale (k et l) couverte par la cellule d'indice (i et j) de la grille locale (après l'application de la translation et de la rotation) sont données par les équations suivantes :

$$\begin{cases} k = a + (i - i_0) \cdot \cos(\theta) + (j - j_0) \cdot \sin(\theta) \\ l = b - (i - i_0) \cdot \sin(\theta) + (j - j_0) \cdot \cos(\theta) \end{cases} \quad (2.13)$$

Avec : i_0 et j_0 sont les indices de la cellule centrale de grille locale.

Cette relation fournit des valeurs réelles, et cela est expliqué par la mauvaise coïncidence entre une cellule de la grille locale et une cellule de la grille globale. Pour résoudre ce problème, il suffit de prendre l'entier le plus proche de k et l .

La position et l'orientation du robot sont calculées en trouvant l'emplacement de la grille locale dans la grille globale ce qui revient à calculer a et b qui correspondent à la position (x et y), quant à l'orientation du robot, elle est évaluée en mesurant l'angle de rotation de la grille locale par rapport à la grille globale.

Pour trouver a , b et θ , il faut déterminer parmi toutes les translations et les rotations possibles de la grille locale par rapport à la grille globale, la translation T_M et la rotation R_M qui maximisent le critère de mise en correspondance.

Considérons $C_L(i, j)$ une cellule de la grille locale, par la transformation F , cette cellule se superpose à une cellule $C_G(k, l)$ de la grille globale. Pour savoir s'il y a une bonne correspondance ou pas, on a qu'à voir le signe du produit des deux cellules superposées.

Si le signe du produit est positif, cela signifie qu'il y a une bonne correspondance (deux cellules libres ou deux cellules occupées), par contre si le signe est négatif, cela signifie que les deux cellules superposées ne se correspondent pas (cellule libre et cellule occupée ou vice versa).

Le produit est effectué sur toutes les cellules de la grille locale, la somme S de tous les produits est calculée.

$C_L \backslash C_G$	Occupée	Libre	Inconnu
Occupée	>0	<0	0
Libre	<0	>0	0
Inconnu	0	0	0

Le critère de mise en correspondance est la somme S , quand S est maximale, cela signifie que le plus grand nombre de cellules de même nature sont superposées.

$$S(a, b, \theta) = \sum C_L(i, j) \cdot C_G(k, l) \quad (2.14)$$

La position du robot sera donnée par a et b , et la rotation sera donnée par θ .

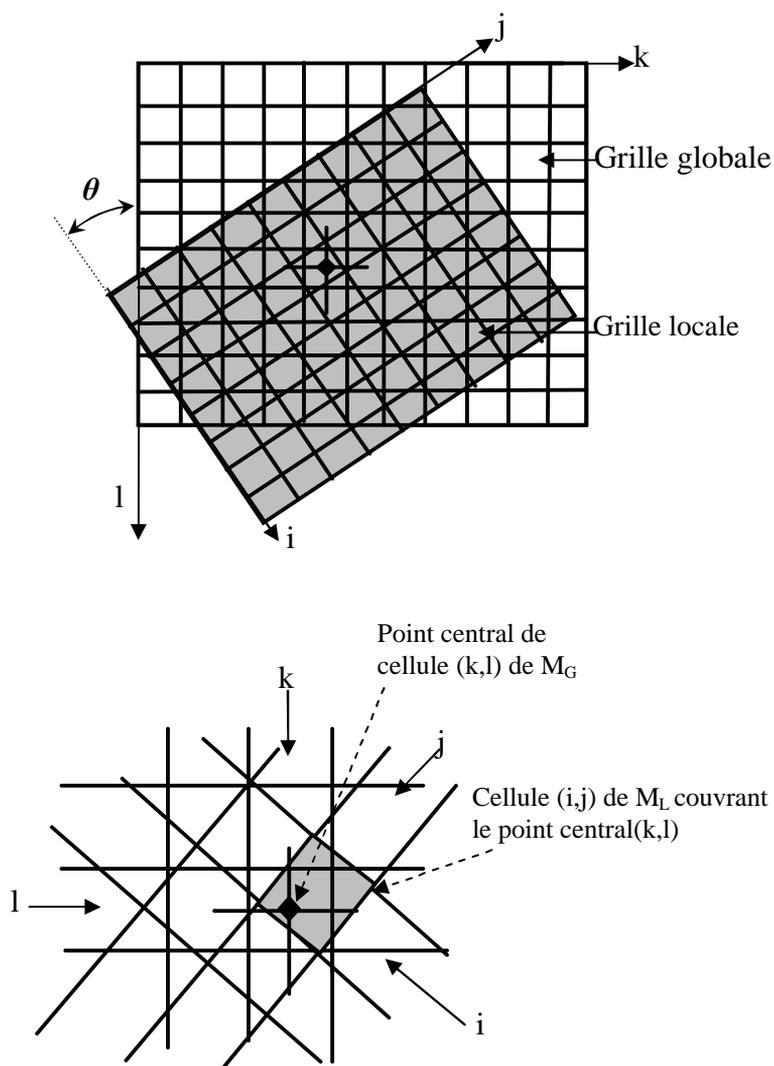


Figure 2.17 : Mise en correspondance des cellules

L'application de la cette méthode nécessite le calcul du critère de mise en correspondance pour toutes les positions et orientations possible du robot dans une pièce, et cela nécessite beaucoup de temps.

Afin de limiter le temps requis pour la localisation, on peut limiter la zone de recherche autour d'une position estimée.

2.6.1 Position estimée

La seule information disponible au début de la trajectoire, en plus les mesures des capteurs à ultrasons, est une information vague de la position du robot qu'on a appelé position et orientation estimée. Dans un premier temps, on a pris la position estimée du robot dans le centre de l'environnement (c.à.d centre de la grille globale) et l'orientation verticale ($\theta_{\text{est}} = 0$).

2.6.2 Zone de recherche

La zone de recherche est une zone de la pièce que doit englober la position réelle du robot. La recherche de la mise en correspondance doit être effectuée dans une zone de $\pm m_x$ autour de la position estimée et pour des orientations comprises dans un cône de $\pm n_x$ autour de l'orientation estimée. Ces valeurs correspondent à une précision raisonnable que peut donner l'utilisateur sur sa position. La définition de cette zone de recherche permet de limiter le temps requis pour la localisation, en effet sans cette indication il faudrait calculer le critère de mise en correspondance pour toutes les positions et orientations possible du robot dans la pièce.

Pour la réalisation de ceci, on a utilisé l'organigramme de localisation de la figure 2.18 .

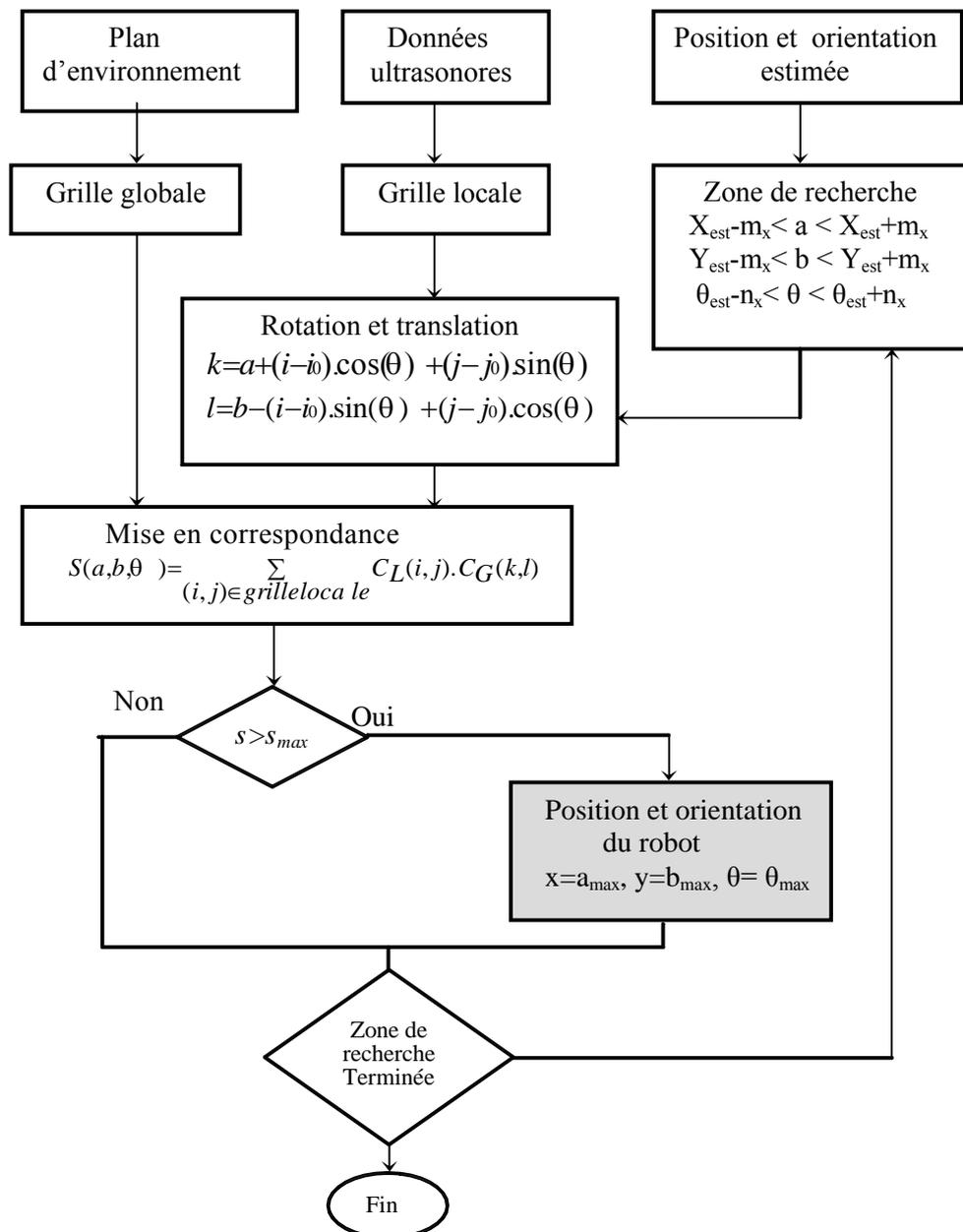


Figure 2.18 : Organigramme de localisation

L'avantage de la méthode de localisation utilisée est qu'elle ne nécessite pas beaucoup de moyens, la seule information disponible au début de la trajectoire, sont les mesures des capteurs à ultrasons.

Quant à la position estimée provient du module de navigation. La méthode de planification de trajectoire qu'on a utilisé est expliquée en détail dans le premier chapitre. Le robot se déplace en suivant des portions de droite qui mène à sa destination.

On a supposé que le robot se déplace avec une vitesse constante qu'on a appelé vitesse estimée, donc la position qu'on va y avoir à chaque moment est une position estimée $(X_{est}, Y_{est}, \theta_{est})$.

En réalité, la vitesse de déplacement n'est pas constante, pour cela on a utilisé un système flou qui génère à sa sortie la vitesse réelle du déplacement. La Figure 2.18 montre l'organigramme de fonctionnement général.

La zone de recherche permet de limiter le temps requis pour la localisation, en effet sans cette indication il faudrait calculer le critère de mise en correspondance pour toutes les positions et orientations possibles du robot dans la pièce ou même dans l'appartement. La recherche de mise en correspondance dans une zone m_x autour de la position estimée et pour des orientations comprises dans un cône de n_x degrés autour de l'orientation estimée.

Un deuxième système flou est utilisé pour générer m_x , ceci nous permet d'avoir une zone de recherche variable, quant la différence entre la position estimée et la position réelle du robot est grande, alors il faut que la zone de recherche englobe cette différence. Par contre si l'écart de position est petit, alors il suffit de chercher la correspondance dans une petite zone.

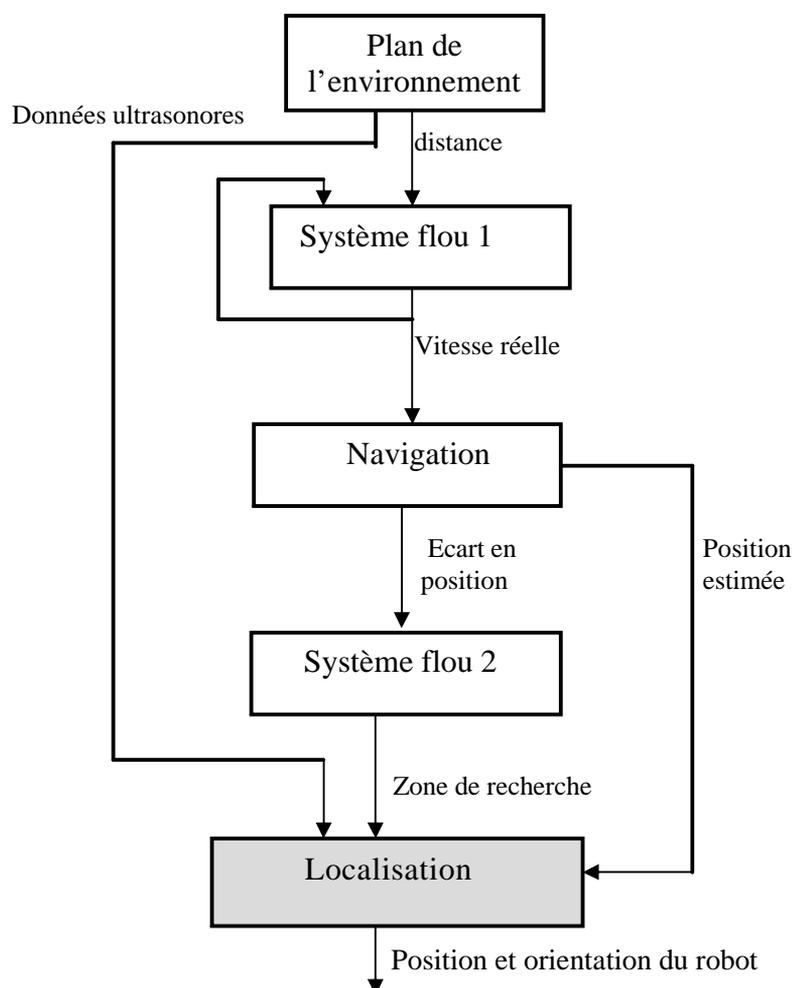


Figure 2.19 : Organigramme général

2.7 Conclusion

Dans ce chapitre, nous avons décrit la manière d'obtenir une grille globale à partir d'un plan de l'environnement, ainsi qu'une grille locale des mesures des capteurs à ultrasons disposés en ceinture.

Nous avons présenté en détails la méthode de mise en correspondance des deux grilles pour déterminer la position et l'orientation du robot.

Les essais en simulation, présentés en chapitre4, ont montré que l'efficacité de la méthode dépend de la zone de recherche et de la position estimée.

Choisir une position estimée fixe et une zone de recherche donnée au départ, suffit seulement dans le cas d'un environnement de petite surface.

Partant de l'idée qu'une position estimée et une zone de recherche fixe ne suffisent pas pour localiser le robot mobile, nous allons dans le chapitre suivant proposer une solution, qui consiste à utiliser un estimateur qui permet de calculer la position estimée, et un système flou pour générer la zone de recherche nécessaire.

CHAPITRE 3

INTÉGRATION DE LA LOGIQUE FLOUE

Pour que le processus de localisation présenté dans le chapitre précédent puisse être mené à bien, certaines connaissances doivent être établies hors-ligne. Il s'agit de la zone de recherche nécessaire dans le calcul et la position estimée du robot mobile. Pour cela nous allons décrire au cours de cette section le système flou de navigation (qui génère la vitesse de déplacement du robot) et le système flou de localisation (qui génère la zone de recherche de localisateur) que nous avons développé.

Nous nous intéresserons pour chaque système, tout d'abord au choix des variables d'entrée et de sortie, au module de fuzzification et de défuzzification, puis à la logique de décision. Nous présenterons finalement les résultats de simulation de chaque système flou.

3.1 Introduction

Le contrôle flou est un système qui transforme une donnée numérique (entrée numérique) en une action de contrôle (sortie numérique), par un traitement flou. Celui-ci consiste d'une part, en la transformation (fuzzification) de cette donnée en une information floue (entrée floue) et d'autre part, le traitement (inférence) de cette information dans une base de connaissance floue pour donner une information floue (sortie floue). Cette information sera convertie en une donnée numérique (défuzzification) qui représente l'action de contrôle.

Autrement dit, le contrôle flou remplace de la même façon le modèle mathématique par un certain nombre de règles, et utilise des inférences avec une ou plusieurs de ces règles, se basant sur des variables linguistiques.

Les principes de base de la logique floue sont présentés en annexe.

3.2 Structure générale d'un contrôleur flou

Chaque contrôleur basé sur la logique floue [41] est composé de quatre blocs principaux suivants (figure 3.1) :

1. Base de connaissances (règles et paramètres des fonctions d'appartenance).
2. Bloc de décision ou le moteur d'inférence (inférence des opérations sur les règles).
3. Fuzzification (transformation des entrées précises en degrés d'appartenance).
4. Défuzzification (transformation des résultats flous en sorties précises).

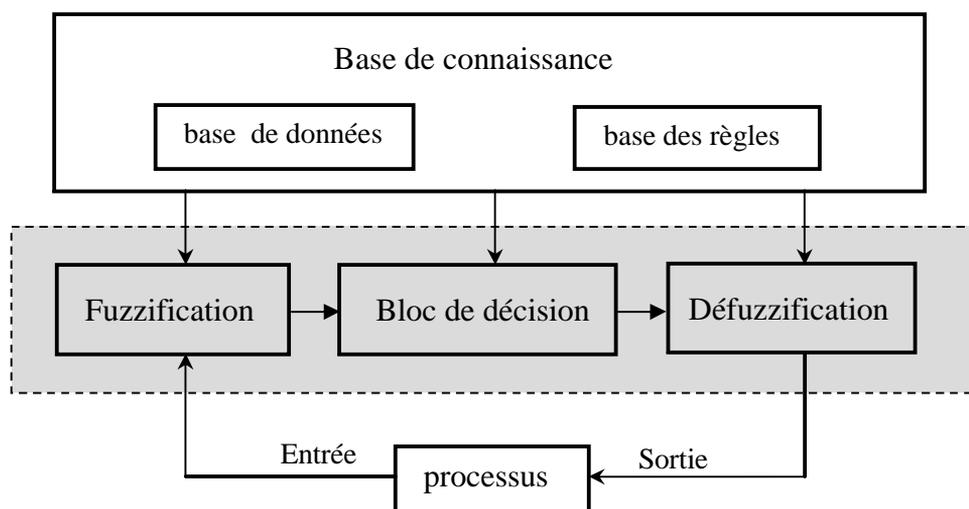


Figure 3.1 : structure générale d'un contrôleur basé sur la logique floue

3.2.1 Fuzzification

La fuzzification comprend:

- Les mesures des variables d'entrées.
- La conversion analogique/digitale, ainsi que le traitement des grandeurs mesurées.
- L'attribution d'un ensemble de degrés d'appartenance à chaque valeur d'entrée.

Tout dépend de la résolution du réglage désirée, une variable x peut prendre trois, cinq ou sept valeurs linguistiques. La désignation standard des ensembles flous est (figure 3.2) :

- NG : négatif grand
- NM : négatif moyen
- NP : négatif petit

EZ : environ zéro

PP : positif petit

PM : positif moyen

PG : positif grand

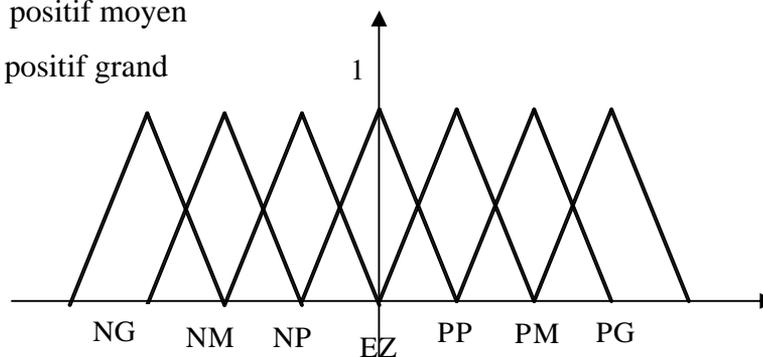


Figure 3.2 : fuzzification avec sept fonctions d'appartenance

3.2.2 Logique de décision

Nous allons considérer un contrôleur flou à n règles linguistiques, deux entrées convenablement transformées en variables linguistiques x , y et une sortie z :

Si $x=A_1$ et $y=B_1$ alors $z=C_1$

Si $x=A_2$ et $y=B_2$ alors $z=C_2$

\vdots

Si $x=A_n$ et $y=B_n$ alors $z=C_n$.

(3.1)

Où : x , y et z sont des variables linguistiques qui représentent les variables d'état du processus et la variable de contrôle. A_i , B_i et C_i ($i=1,2, \dots,n$) sont les sous-ensembles flous définis dans les ensembles de référence pour x , y , et z respectivement.

Le schéma de ce contrôleur est représenté sur la figure 3.3.

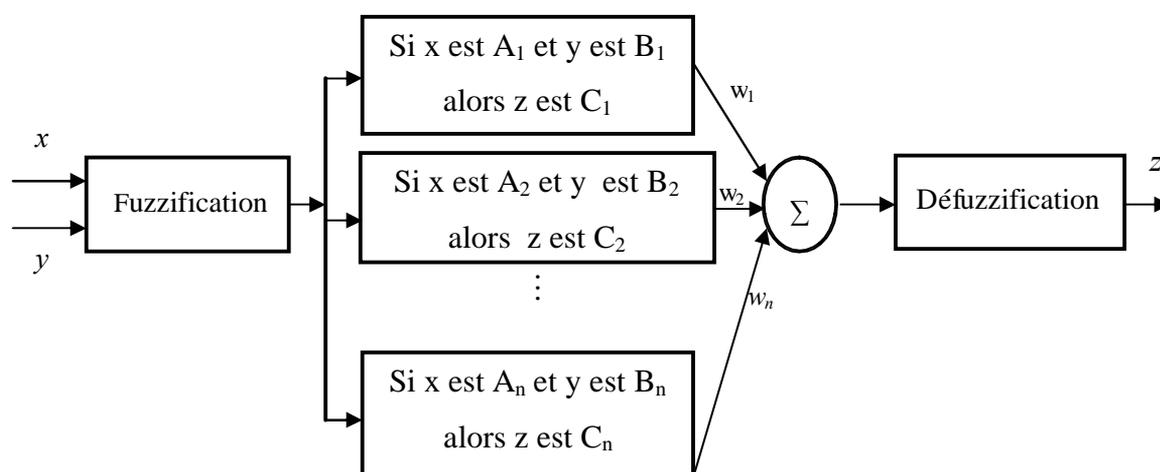


Figure 3.3 : Structure générale de la partie floue du contrôleur

Dans le sens mathématique, l'activation de règles est l'application de l'opérateur min (ou un autre opérateur T-normes) pour obtenir le poids d'activation de chaque règle. Lorsque les opérateurs d'implication sont ceux de Mamdani et de Sugeno, il est possible d'interpréter, graphiquement, le calcul de la valeur de C_i conséquence de l'ensemble des règles composant la base de connaissance. (voir figure 3.4 et figure 3.5).

Dans le cas du contrôleur de la figure 3.3, le poids w_i obtenu de la i -ème règle est :

$$w_i = \mu_{A_i}(x) \text{ et } \mu_{B_i}(y) \quad (3.2)$$

Où :

$\mu_{A_i}(x)$ et $\mu_{B_i}(y)$ sont des valeurs d'appartenance de x et y respectivement aux sous-ensemble A_i et B_i . Ce veut dire que la partie conséquente de la i -ème règle ($z=C_i$) doit être activée avec un niveau de vérité w_i (poids d'activation).

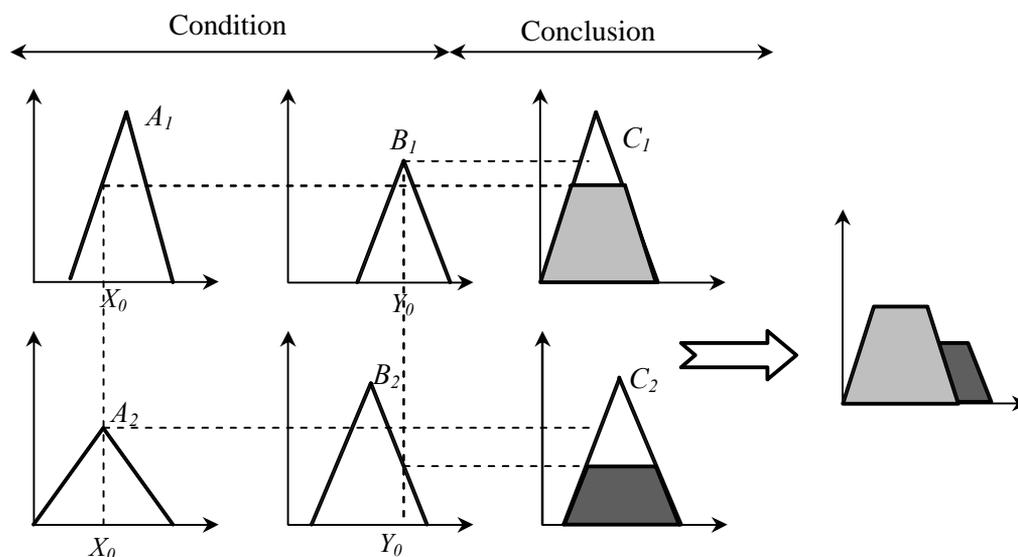


Figure 3.4 :Interprétation graphique du calcul du résultat d'un contrôleur flou de type Mamdani

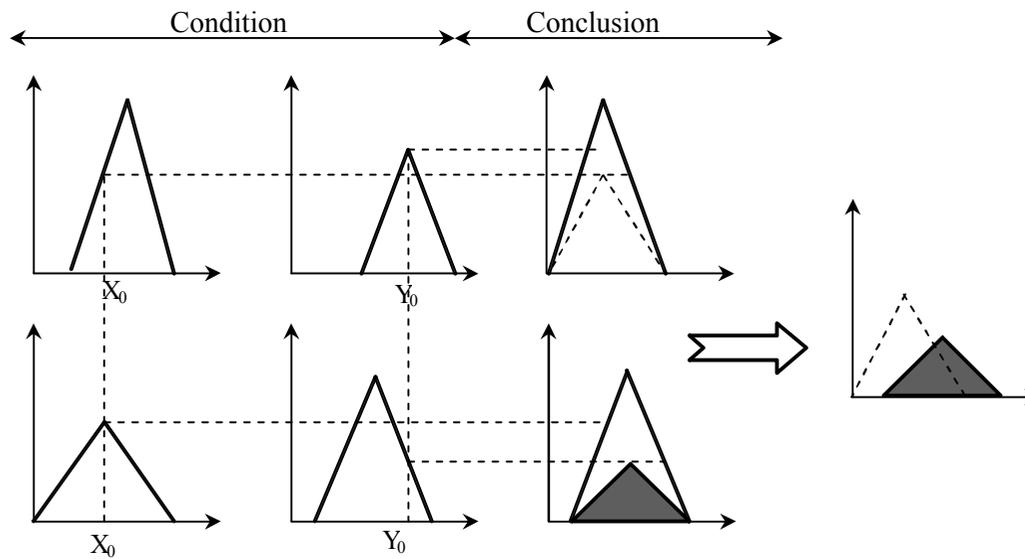


Figure 3.5 : Interprétation graphique du calcul du résultat d'un contrôleur flou de type Sugeno

3.2.3 Défuzzification

Le résultat d'une inférence floue est une fonction d'appartenance. C'est un sous-ensemble flou. Un organe de commande nécessite un signal de commande précis. La transformation d'une information floue en une information déterminée est la défuzzification (concrétisation). De plus, on doit souvent prévoir un traitement du signal et la conversion digitale /analogique.

Parmi les méthodes de défuzzification, on propose les suivantes :

3.2.3.1 Méthode du centre de gravité

C'est la méthode la plus mentionnée dans la littérature. Le centre de gravité est donnée par :

3.2.3.2 Méthode de moyenne de maximum

Cette méthode génère une commande précise en calculant la moyenne des valeurs pour lesquelles l'appartenance est maximale. La valeur défuzzifiée est donnée par:

$$z = \sum_{i=1}^L \frac{r_i}{L} \quad (3.3)$$

Où : L est le nombre de valeurs qualifiées r pour lesquelles l'appartenance est maximale.

3.2.3.3 Méthode de moyenne pondérée

Cette méthode est utilisée lorsque les sorties sont définies comme fonctions linéairement dépendantes d'entrées. En générale, la partie conséquence de la règle est : $z = f(x, y)$

Si w_i est le poids d'activation de la règle i, la valeur précise de la commande est :

$$z = \frac{\sum_{i=1}^n w_i \cdot f(x_i, y_i)}{\sum_{i=1}^n w_i} \quad (3.4)$$

Où : n est le nombre de règles activées.

3.3 Les systèmes flous développés

Nous allons décrire au cours de cette section les systèmes flous de navigation réactive que nous avons développés. Nous décrirons successivement les différents modules présentés figure 3.1. Nous nous intéresserons tout d'abord au choix des variables d'entrée et de sortie au module de fuzzification et de défuzzification puis le bloc de décision.

La stratégie envisagée pour le guidage du robot peut être résumée comme suit : le robot doit foncer tout droit avec une vitesse qui dépend de la distance qui le sépare de l'objectif (ou l'obstacle). Si la distance est très longue, le robot se déplace avec une vitesse importante, celle-ci diminuera au fur et à mesure que le robot s'approche du but.

Donc le système de navigation doit admettre en entrée la vitesse actuelle du robot ainsi que de la distance séparant le robot du but (ou but partiel). Ces données vont être codées afin de pouvoir être manipulées par les règles floues du système.

3.3.1.2 Module de fuzzification

Comme nous l'avons signalé au cours de la section 2.1, il n'existe pas de manière unique ni systématique de réaliser le fuzzification des données d'entrée.

On a utilisé pour les fonctions d'appartenance de la distance la forme triangulaire pour les prédicats C (Courte), M (Moyenne) et L (Long), et la forme trapézoïdale pour les prédicats TC (Très Courte) et TL (Très long)(Figure3.6). La distance maximale est variable selon la surface de l'environnement.

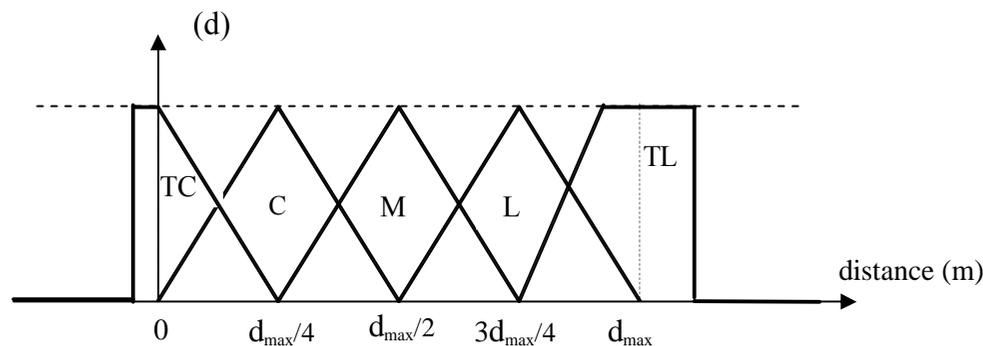


Figure 3.6. Fonctions d'appartenance de la distance

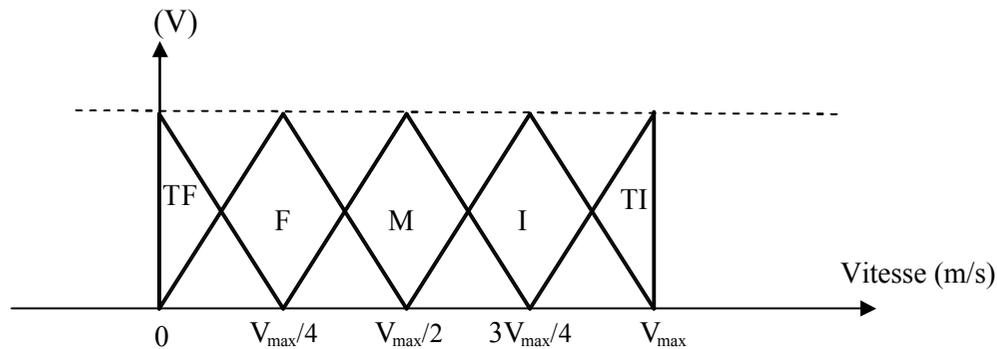


Figure 3.7 : Fonction d'appartenance de la vitesse

3.3.1.3 Détermination des règles

Il existe quatre méthodes principales permettant de dériver des règles de contrôle :

1. En interrogeant un expert. Le but du contrôle flou est alors non pas de modéliser le système à contrôler mais de modéliser le comportement de la personne en train de le contrôler.
2. Par observation des commandes fournies par un opérateur. Il existe des systèmes q'un opérateur humain est en mesure de contrôler sans pouvoir expliciter les règles utilisées.
3. En construisant un modèle flou du système. Cela consiste tout d'abord à créer une description linguistique du processus dynamique que l'on souhaite contrôler. Cette description peut être considéré comme un modèle flou de ce processus.
4. Par apprentissage. Les approches basées sur l'apprentissage ont pour objectif de créer automatiquement une base de règles ou de modifier une base déjà

- ❖ Règle1 : Si «la distance» est TC et «la vitesse» est (F ou M ou I ou TI) Alors changer «la vitesse» à TF.
- ❖ Règle2 : Si «la distance» est C et «la vitesse» est (TF ou M ou I ou TI) Alors changer «la vitesse» à F.
- ❖ Règle3 : Si «la distance» est M et «la vitesse» est (TF ou F ou I ou TI) Alors changer «la vitesse» à M.
- ❖ Règle4 : Si «la distance» est L et «la vitesse» est (TF ou F ou M ou TI) Alors changer «la vitesse» à I.
- ❖ Règle5 : Si «la distance» est TL et «la vitesse» est (TF ou F ou M ou I) Alors changer «la vitesse» à TI.
- ❖ Règle6 : Si «la distance» est TC et «la vitesse» est (TF) Alors garder «la vitesse» TF.
- ❖ Règle7 : Si «la distance» est C et «la vitesse» est (F) Alors garder «la vitesse» F.
- ❖ Règle8 : Si «la distance» est M et «la vitesse» est (M) Alors garder «la vitesse» M.
- ❖ Règle9 : Si «la distance» est L et «la vitesse» est (I) Alors garder «la vitesse»

l'avons vu dans la section 2.3, il existe trois solutions principales. Après plusieurs essais nous avons choisit la méthode de centre de gravité.

3.3.1.5 Résultat expérimental

Un des grands avantages des approches floues est la grande flexibilité des systèmes obtenus permettant de “sculpter” la fonction selon ses besoins favorisant ainsi la mise au point.

La figure 3.8 représente la commande en vitesse linéaire obtenue en fonction de la distance au but et la vitesse actuelle, qui permet la navigation dans le troisième exemple d'environnement (200x200cellule).

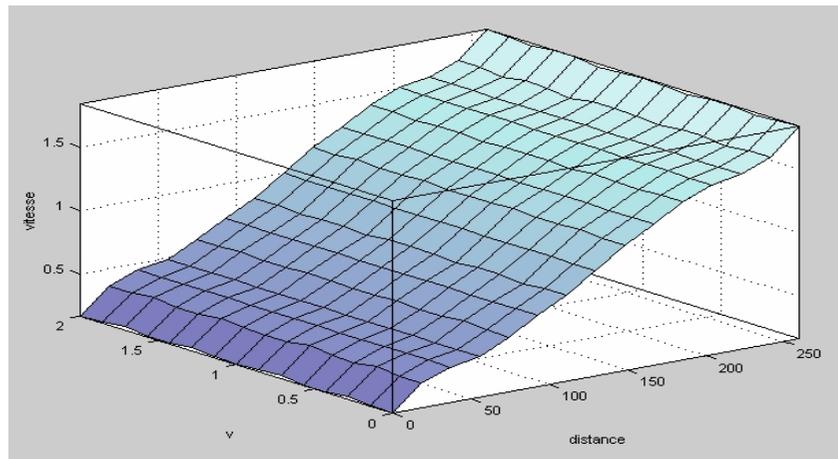


Figure3.8: vitesse générée par le système flou en fonction de la distance et la vitesse(v)

3.3.2 Système flou2

L'écart en position ce n'est pas l'erreur en position, mais c'est une valeur approximative de cette erreur.

3.3.2.2 Module de fuzzification

La phase de réglage est essentielle dans la conception d'un contrôleur flou, donc après avoir varié le nombre de règles utilisées et les fonctions d'appartenance on a abouti aux prédicats (TP, P, M, G, TG) qui sont utilisés pour l'entrée, distribués d'une manière équidistante dans le domaine $[0, E_{Pmax}]$, avec des fonctions d'appartenance d'une forme triangulaire (Figure 3.9).

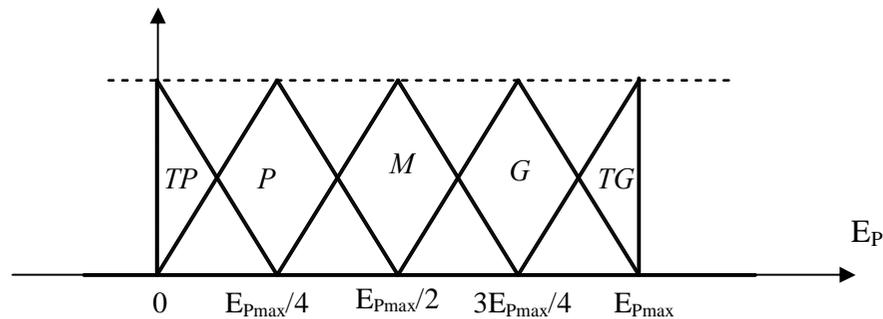


Figure 3.9 : fonctions d'appartenance de l'écart en position

On a utilisé pour les fonctions d'appartenance de la variable de sortie la forme Triangulaire pour les prédicats TP (Très Petit), P (Petit), M (Moyen), G (Grand), TG (Très Grand) (Figure 3.10).

3.3.2.3 Détermination des règles

Nous utiliserons donc la première approche aussi afin de déterminer les règles utilisées dans notre système.

Donc nous avons utilisé la liste des règles floues suivante :

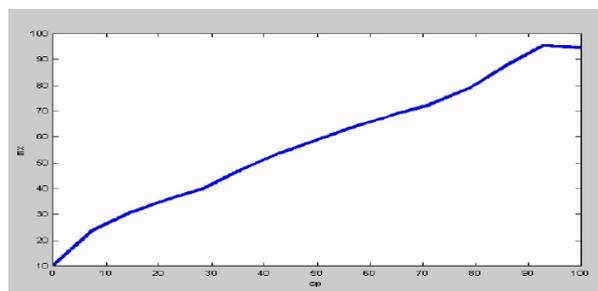
- Règle1 : Si « l'écart en position » est TP alors « la zone de recherche » est TP.
- Règle2 : Si « l'écart en position » est P alors « la zone de recherche » est P.
- Règle3 : Si « l'écart en position » est M alors « la zone de recherche » est M.
- Règle4 : Si « l'écart en position » est G alors « la zone de recherche » est G.
- Règle5 : Si « l'écart en position » est TG alors « la zone de recherche » est TG.

3.3.2.4 Module de défuzzification

Pour le module de défuzzification nous avons utilisé la même méthode que le système précédent (la méthode de centre de gravité).

3.3.2.5 Résultat expérimental

La figure 3.11 représente la zone de recherche (m_x) obtenue en fonction de l'écart en position (E_p) dans le troisième exemple d'environnement (200*200cellule).



3.4 Conclusion

La logique floue est un formalisme permettant de construire une transformation continue entre un espace d'entrée et un espace de sortie à l'aide de connaissances fournies par le concepteur et exprimées sous forme de règles.

La propriété d'approximateur universel pour les systèmes flous indique que ce formalisme à la puissance nécessaire pour décrire la solution que nous recherchons.

La logique floue nous a permis de développer un système qui génère la vitesse de déplacement du robot mobile, ainsi qu'un autre système flou permettant d'améliorer la méthode de mise en correspondance des grilles afin de localiser le robot mobile.

Une méthode de réduction des grilles est développée dans le chapitre suivant, afin de minimiser le temps de calcul, ainsi que les résultats de simulation.

CHAPITRE 4

OPTIMISATION DE LA MÉTHODE ET RÉSULTATS DE SIMULATION

4.1 Introduction

Nous avons vu que la localisation par mise en correspondance des grilles dépendait de la qualité de l'estimation initiale de la position et de l'orientation ainsi que de la zone de recherche. D'un autre côté, la localisation basée sur le modèle a un coût calculatoire important du fait de la succession de plusieurs étapes de calcul. Ceci est d'autant plus vrai pour la localisation par les grilles d'occupation car chaque étape nécessite dans ce cas le traitement d'une grande quantité de données. Lors du développement du système de localisation, nous avons visé comme objectif d'optimiser les performances de chacune des étapes de localisation en terme de sensibilité aux conditions initiales, de temps de calcul. Pour cela, nous allons décrire dans cette section, une méthode de réduction des grilles permet de réduire le temps de calcul.

En terme de précision, la plupart des systèmes de localisation basés sur les grilles d'occupation, (dans un domaine du même ordre de grandeur que celui d'un environnement d'intérieur : quelques mètres voire quelques dizaines de mètres), permettent d'atteindre une précision d'environ 8° pour l'orientation et environ 10 cm pour la position [17]. Cette précision est suffisante pour des missions de localisation d'un fauteuil électrique et peut servir de référence pour l'évaluation de notre travail.

Le chapitre suivant concerne les résultats de simulation effectués pour différents exemples d'environnement.

Dans le chapitre précédent nous avons évoqué les deux systèmes flous(le premier utilisé pour générer la vitesse réelle du déplacement du robot, et le deuxième permet d'avoir une zone de recherche variable).

Au cours de ce chapitre, nous étudions plus précisément la possibilité d'optimiser le temps nécessaire pour l'exécution du localisateur en intégrant les systèmes flous développés dans le chapitre précédent et en maintenant une précision

tolérable pour les tâches exécutées par le fauteuil. Pour mettre en évidence nos travaux nous avons appliqué la méthode pour trois types d'environnement :

- ❖ Premier environnement : pièce carrée de 4 mètres de côté, chaque cellule de la grille de base (40x40) a une dimension de 10 cm.
- ❖ Deuxième environnement : pièce de 10 mètres de longueur et 7 mètres de largeur, chaque cellule de la grille de base (70x100) a une dimension de 10 cm.
- ❖ Troisième environnement : pièce carrée de 10 mètres de côté, chaque cellule de la grille de base (200x200) a une dimension de 5 cm.

Le localisateur sera considéré comme réussit dans le cas où l'erreur ne dépasse pas trois cellules.

4.2 Réduction des grilles

Cette méthode a été proposée par O.HORN [28].

La recherche analytique de toutes les positions et orientations possibles de la grille locale par rapport à la grille globale entraînerait une charge de calcul si lourde que la méthode serait inapplicable. En effet, si les deux grilles ont des dimensions $n \times n$ pour les n^2 cases possibles de positionnement de la grille locale, il faut effectuer n^2 produits pour mesurer la ressemblance entre les deux grilles et ce dans les $2n$ orientations possibles. Le nombre de calcul requis est donc en n^5 soit si $n=64$, on a 10^9 produits à traiter.

Afin de replacer la charge de calcul dans une limite raisonnable, nous mettons en œuvre une méthode de recherche hiérarchisée, pour cela nous réalisons une compression des grilles d'occupation suivant les critères suivants :

- Quatre cellules voisines (c_1 , c_2 , c_3 et c_4) au niveau k produisent une cellule (c) au niveau $k+1$.
- Si l'une des cellules est occupée ($c_j = 1$) la cellule c sera affectée à la valeur 1.
- Si aucune cellule n'est occupée e) si l'une des quatre est vide ($c_j = -1$) la valeur affectée sera -1 .
- Si les quatre cellules sont inconnues ($c_j = 0$) la valeur affectée sera 0.

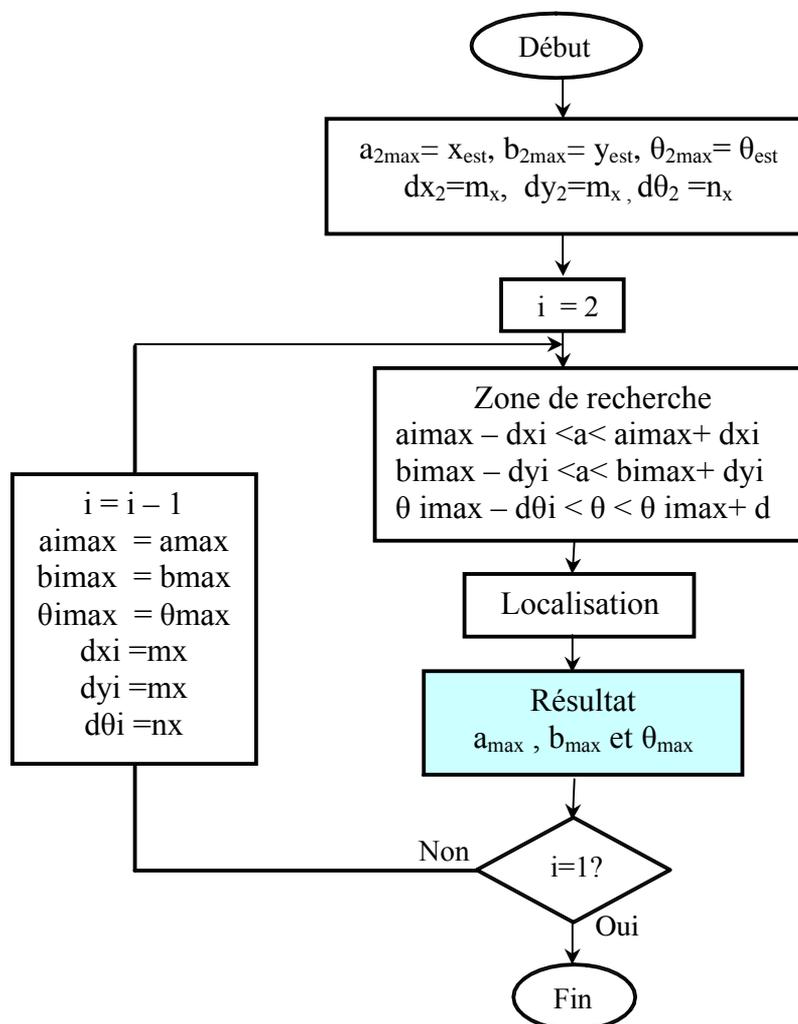


Figure 4.2 : Organigramme de localisation hiérarchisée

4.3 Résultats de simulation

L'organigramme de la méthode de localisation présenté par la figure 2.18 à été implémenté sur un Pc PentiumIII sous matlab. Nous avons en premier temps, testé les performances à partir de mesures simulées des capteurs ultrasonores, c'est-à-dire qu'à partir d'un plan du lieu et d'une position donnée du robot, nous calculons les données que les 12 capteurs US devraient délivrer. Pour chaque capteur, il s'agit de la distance qui le sépare du premier obstacle rencontré dans son cône d'émission.

Nous lançons ensuite la procédure de localisation avec comme données le plan du lieu, les valeurs simulées des capteurs et une position estimée fixe (au centre du plan de l'environnement).

4.3.1 Exemple 1

Nous avons, dans un premier temps, testé les performances de la méthode sur une pièce carrée de 4 mètres de coté et peu encombrée (figure 4.3), chaque cellule de la grille de base (40x40) a une dimension de 10 cm.

Les positions sont représentées en cellules ($X_c=21$ ce qui correspond à 20x10 cm soit 210cm). La position estimée a été fixée au centre de la grille globale ($X_{est}=20$ et $Y_{est}=20$). L'orientation estimée c'est l'angle pris par le robot pour empreinter une portion de droite en cours avec une marge de 10° (puisque le déplacement du robot est rectiligne).

Pour que le critère de la mise en correspondance s'étale sur tout l'environnement, il est donc impératif de prendre la zone de recherche de 20 cellules ($m_x=20$).

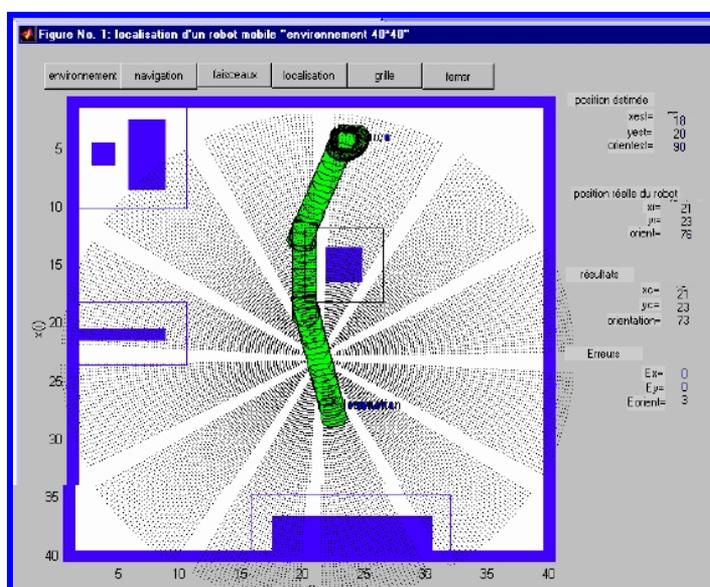


Figure 4.3: déplacement du robot et la vue des capteurs au moment de la demande de la position du 1^{er} plan d'environnement

La figure 4.4 représente la grille locale construite au moment où le robot a été arrêté afin de déterminer sa position.

Nous comparons ensuite la position réelle du robot avec celle calculée par le l'algorithme développé.

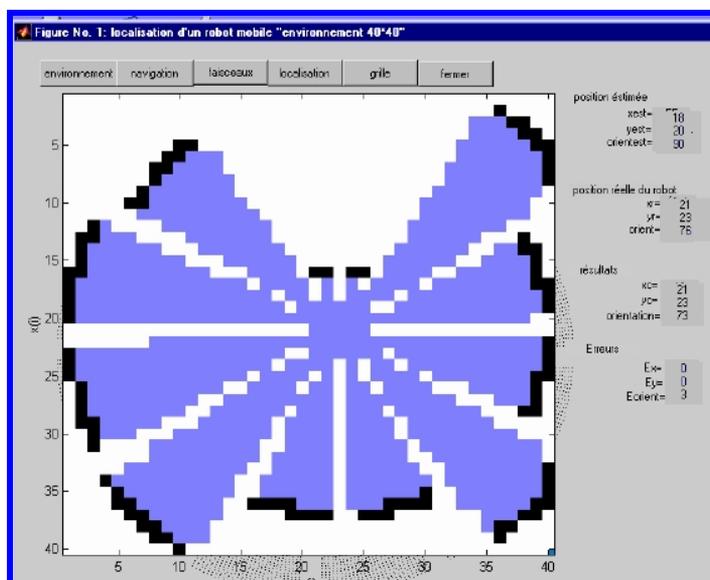


Figure 4.4 : Grille locale construite à l'arrêt du robot

Nous avons jugé que les résultats obtenus rentre dans le cadre de l'acceptable, les graphes des figures 4.5 , 4.6 et 4.7 confirme cette affirmation et montre clairement que l'erreur en position est au maximum 1 cellule (10 cm) (figure 4.8), par contre l'erreur en orientation est assez importante (8°) (figure 4.9) ceci est toléré par l'utilisation du robot.

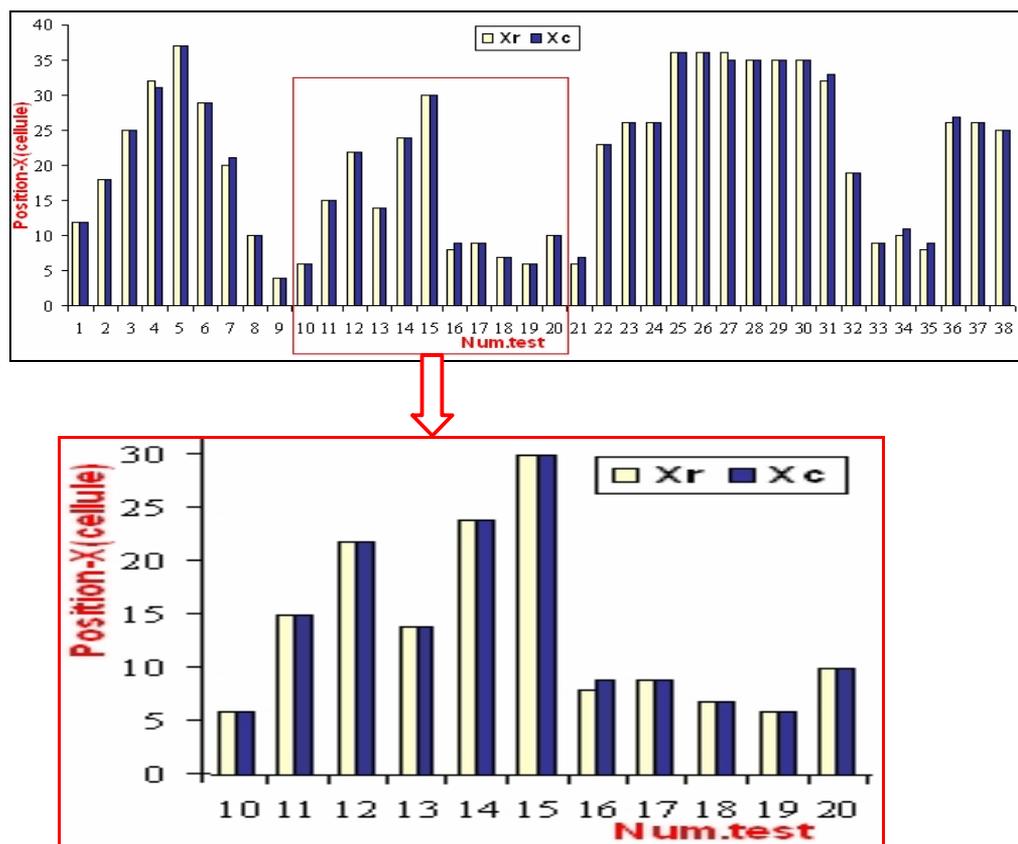


Figure 4.5 : Les résultats de la « position en X » pour le premier exemple (40x40)

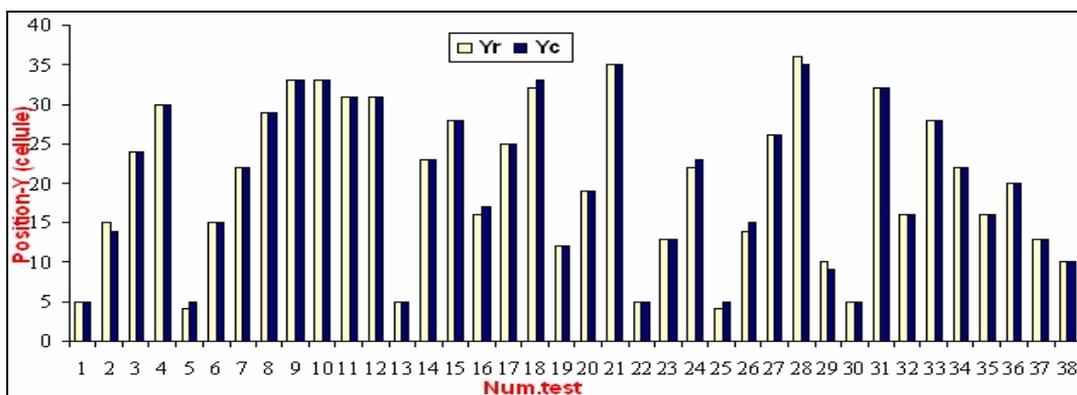


Figure 4.6 : Les résultats de la « position en Y » pour le premier exemple (40x40)

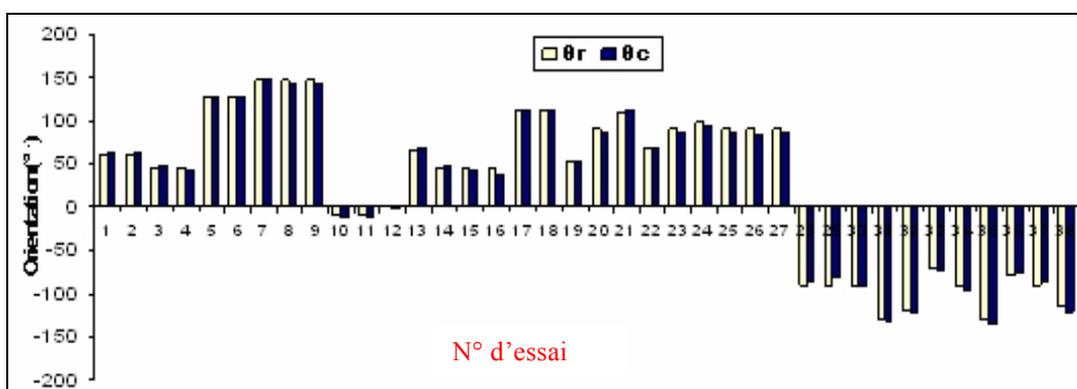


Figure 4.7 : Les résultats de « l'orientation » pour le premier exemple (40x40)

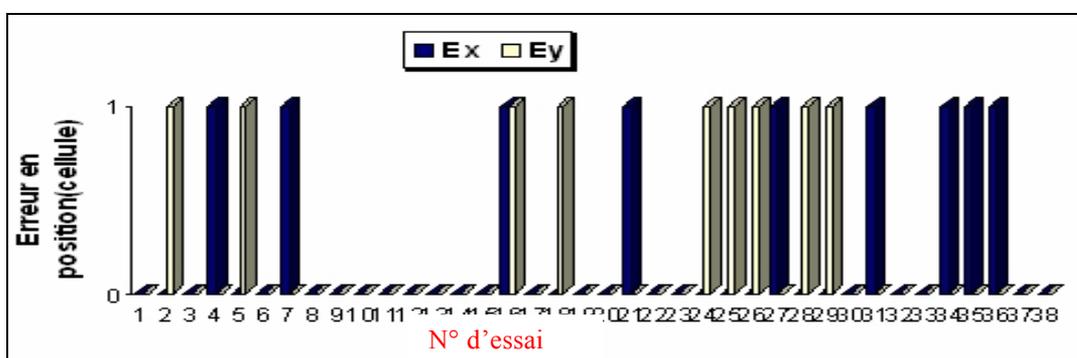


Figure 4.8 : « Erreur en position » pour le premier exemple (40x40)

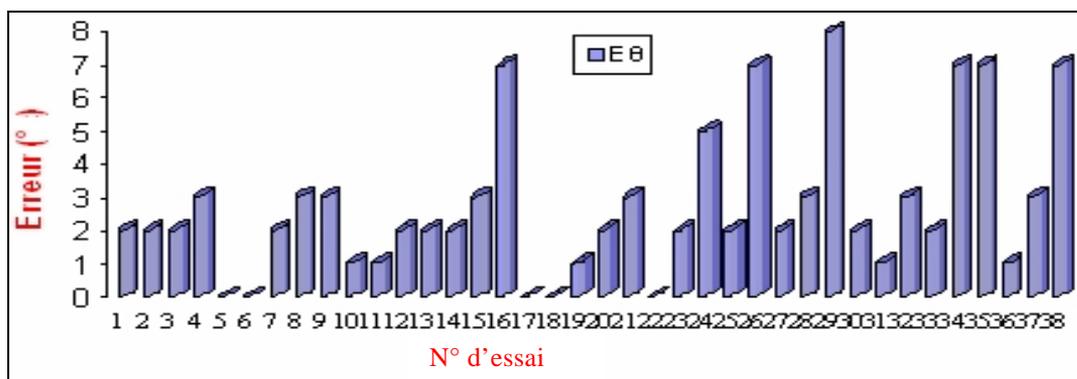


Figure 4.9 : « Erreur en orientation » pour le premier exemple (40x40)

4.3.2 Exemple2

Dans le deuxième exemple (figure 4.10), on a testé notre localisateur dans une autre pièce de longueur de 10 mètres et de largeur de 7 mètres.

Comme dans le premier exemple, le résultat présenté est obtenu à partir d'une résolution de 10x10 cm par cellule, la position estimée est fixée dans le centre de la grille globale ($X_{est}=35$ et $Y_{est}=50$), on a gardé la même zone de recherche (c à d $m_x=20$ cellules et $n_x=10^\circ$).

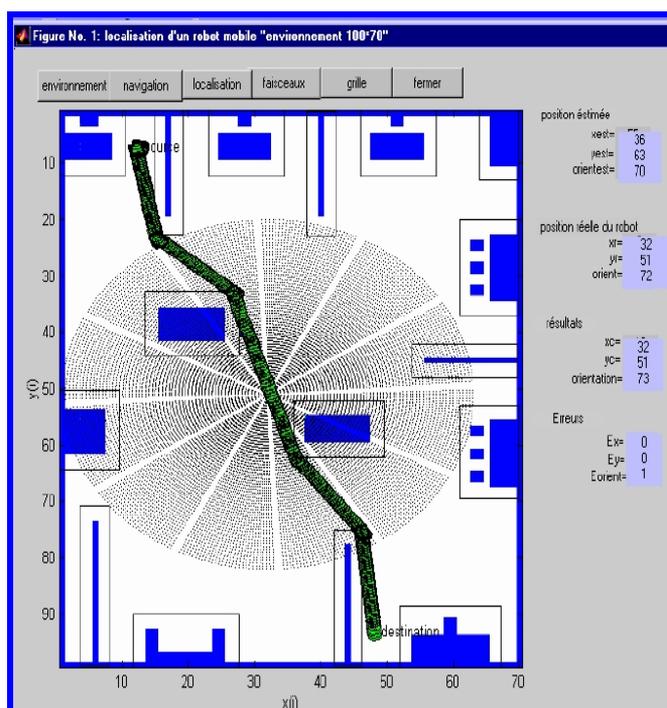


Figure 4.10 : Exemple 2, le déplacement du robot et la grille locale Construite au moment de la demande de la position du robot

Pour tester ce nouveau modèle, nous avons réalisé 37 mesures différentes représentées sur les graphes (des figures 4.12, 4.13 et 4.14) dont 14 répandent aux conditions d'une bonne localisation.

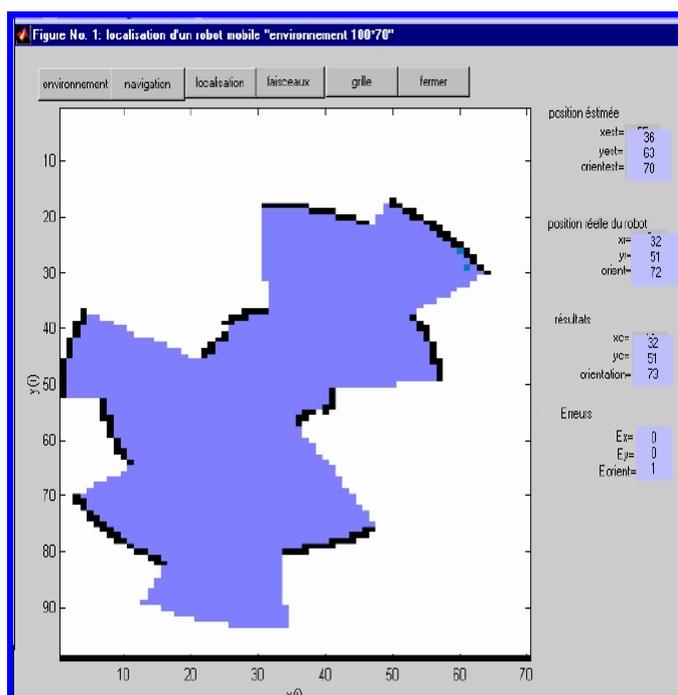


Figure 4.11 : grille locale construite à l'arrêt du robot (environnement 70x100cellules)

L'erreur maximale est d'une cellule soit ± 10 cm, et de $\pm 8^\circ$ en orientation pour des positions concentrés à proximité de la position estimée.

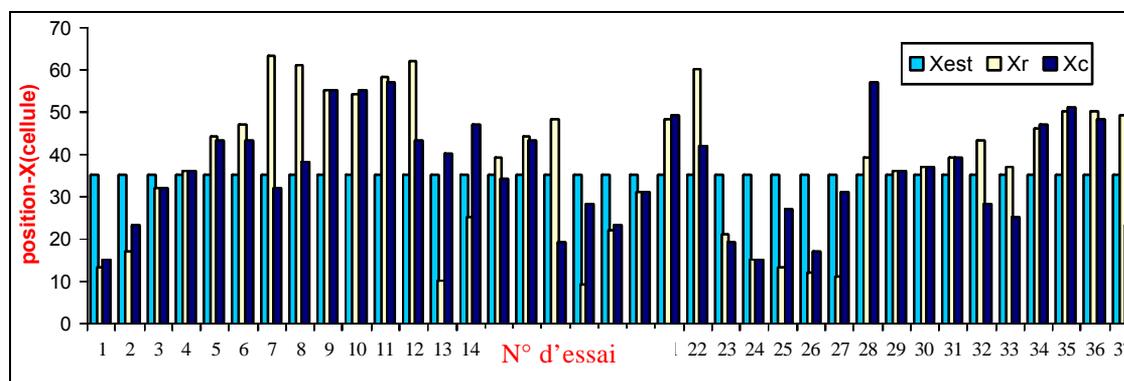


Figure 4.12 : Les résultats de la « position en X » pour le 2^{ème} exemple (100x70)

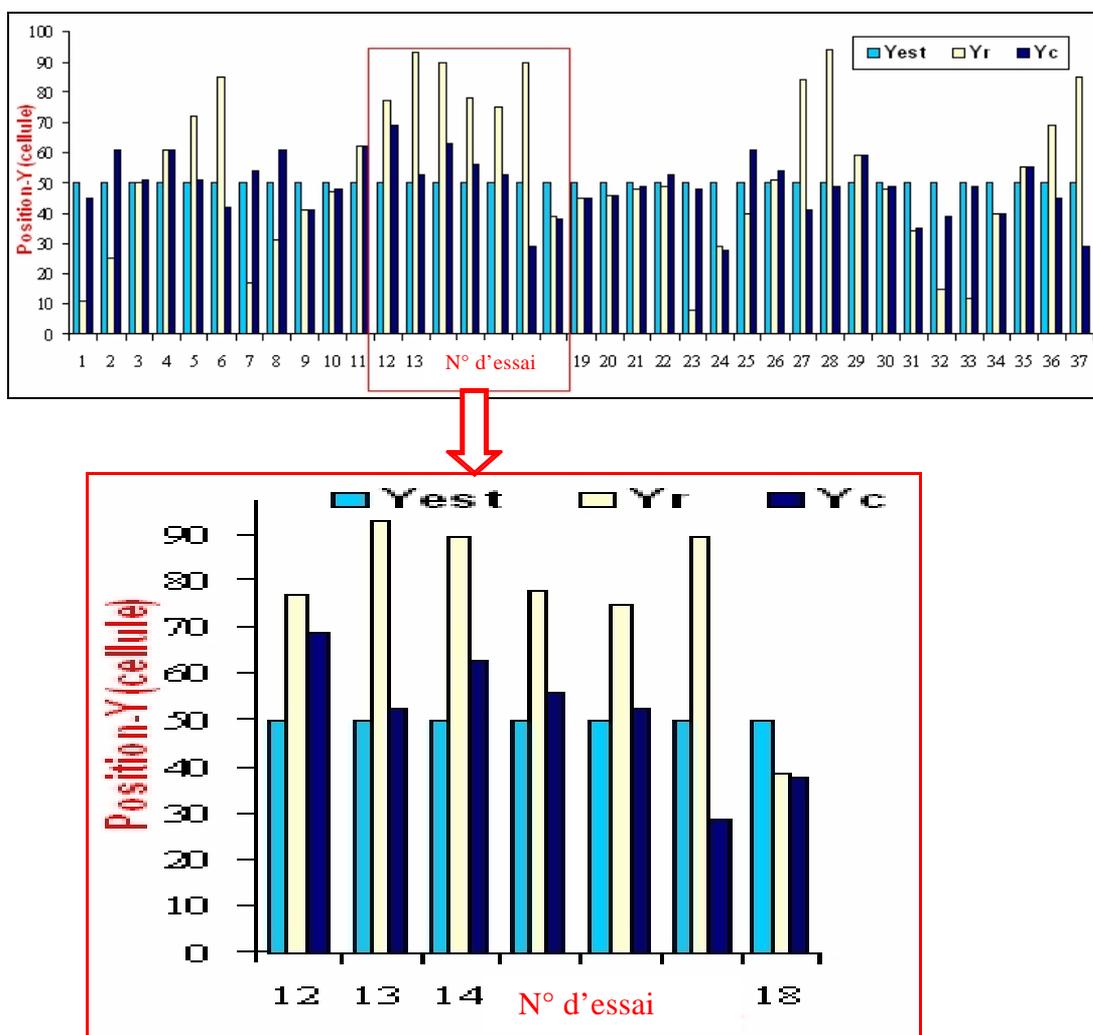


Figure 4.13 : Les résultats de la « position en Y » pour le 2^{ème} exemple (100x70)

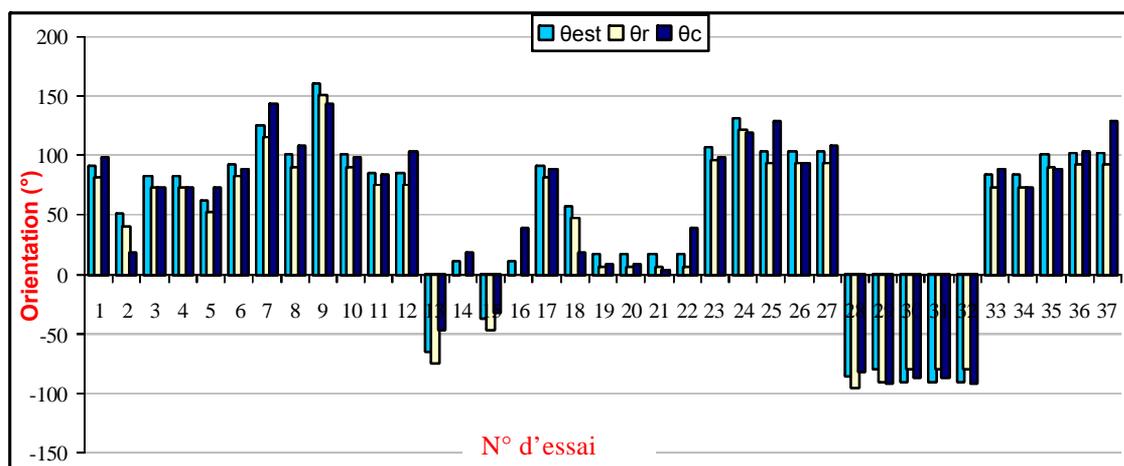


Figure 4.14 : Les résultats de « l'orientation » pour le 2^{ème} exemple (100x70)

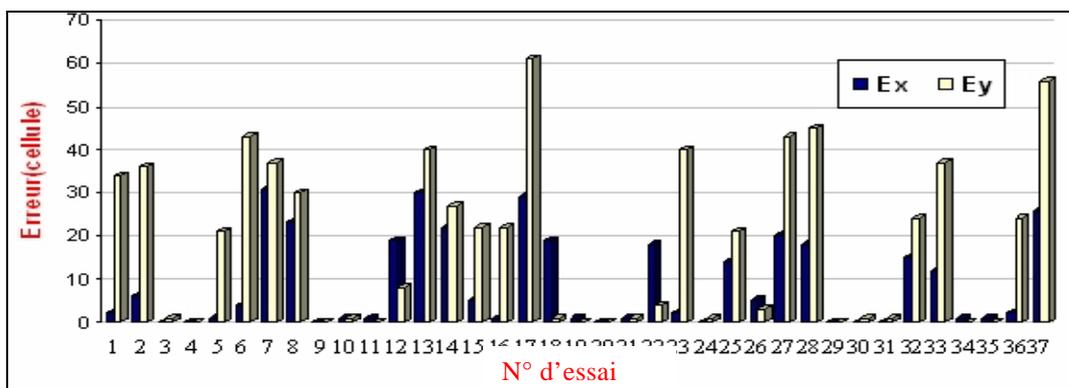


Figure 4.15 : « Erreur en position » pour le 2^{ème} exemple (100x70)

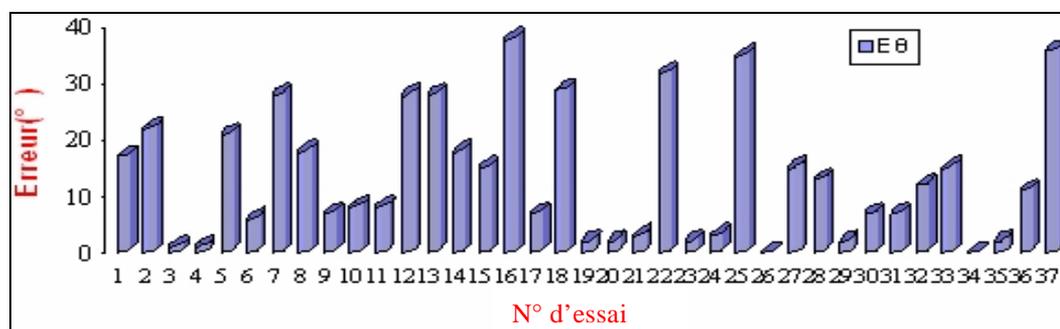


Figure 4.16 : « Erreur en orientation » pour le 2^{ème} exemple (100x70)

Il est à remarqué que lorsque la position réelle du robot s'éloigne de la position estimée, la position calculée par notre méthode est erronée.

Par ailleurs, l'analyse des échecs, en simulation, montre que ceux-ci sont liés à l'influence de la zone de recherche, car cette dernière doit englober la position réelle du robot.

Afin de compenser cette source d'erreur, nous avons proposé de rendre la zone de recherche variable et la position estimée variable aussi, d'où l'intégration de la logique floue (Chapitre3).

La figure 4.17 représente la trajectoire réelle parcourue par le robot c'est-à-dire le déplacement du robot se fait avec une vitesse générée par le système flou, alors que la trajectoire estimée c'est le déplacement du robot avec une vitesse constante.

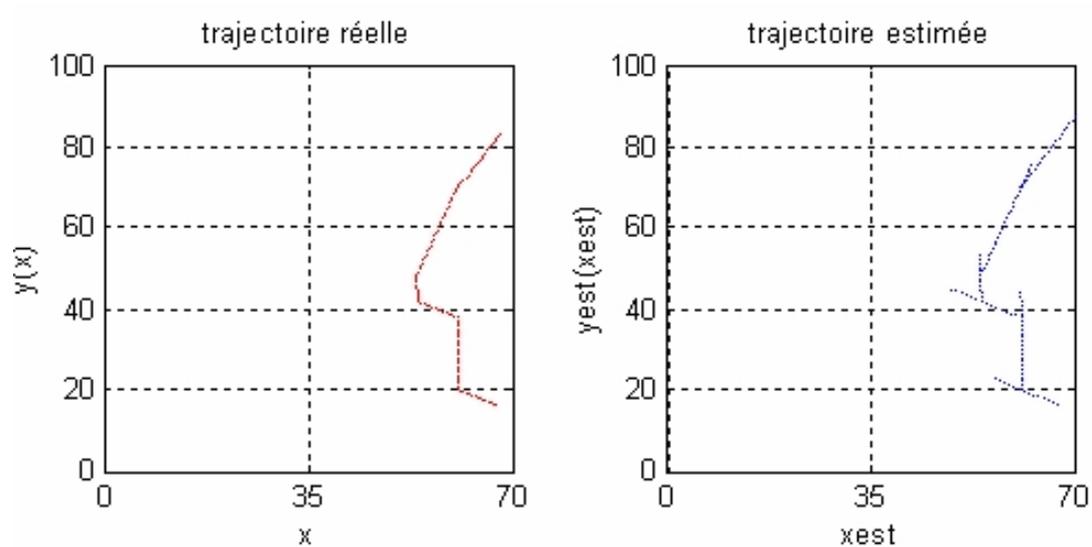


Figure 4.17 : Exemple de trajectoire réelle parcourue par le robot et la trajectoire estimée correspondante (environnement 70x100 cellules)

Moyennant les deux systèmes flous, les résultats donnés par la méthode sont d'une précision correcte : erreur maximale de deux cellules en position et de 9° en orientation.

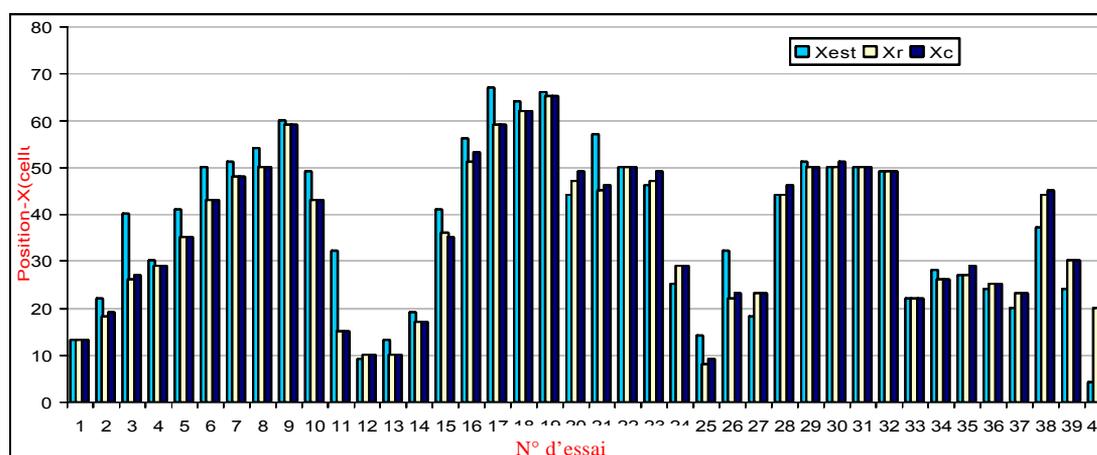


Figure 4.18 : Les résultats de la « position en X » pour le 2^{ème} exemple en utilisant les systèmes flous (100x70)

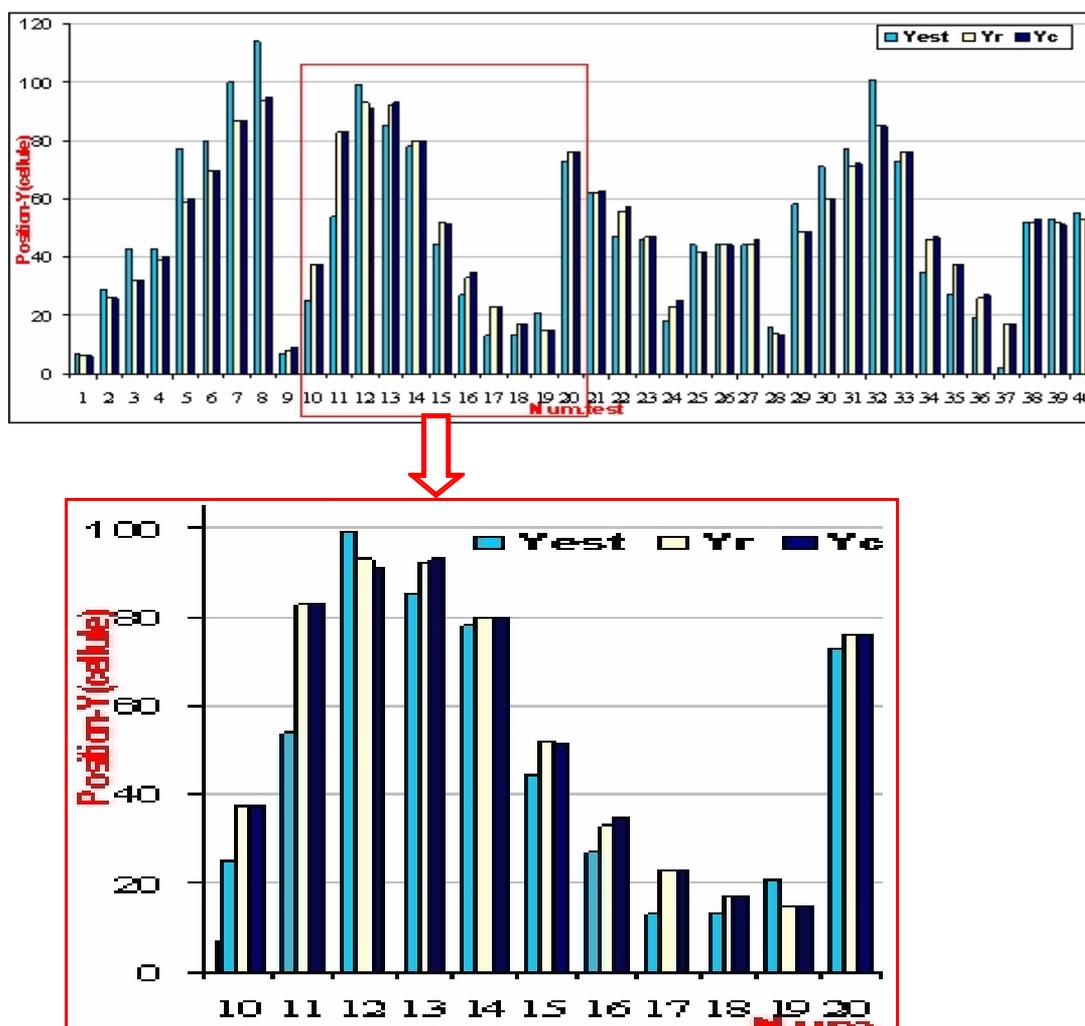


Figure 4.19 : Les résultats de la « position en Y » pour le 2^{ème} exemple en utilisant les systèmes flous (100x70)

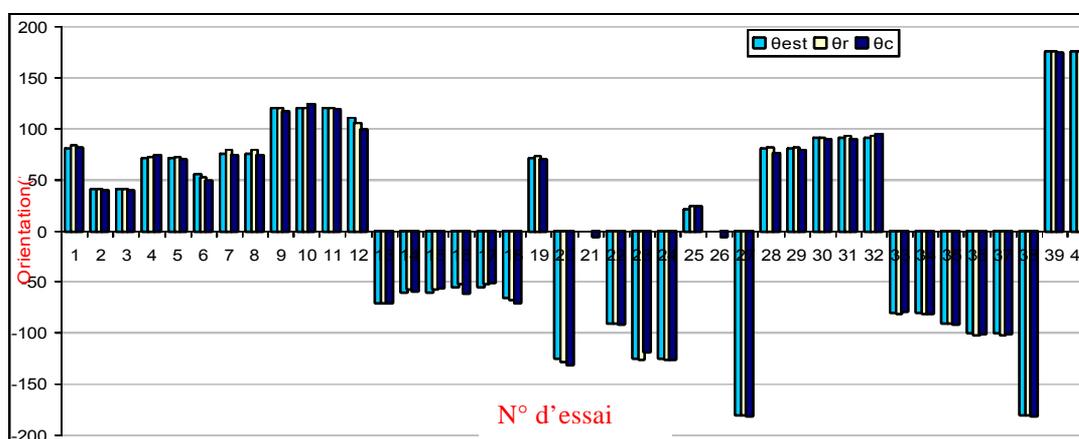


Figure 4.20 : Les résultats de « l'orientation » pour le 2^{ème} exemple (100x70) en utilisant les deux systèmes flous

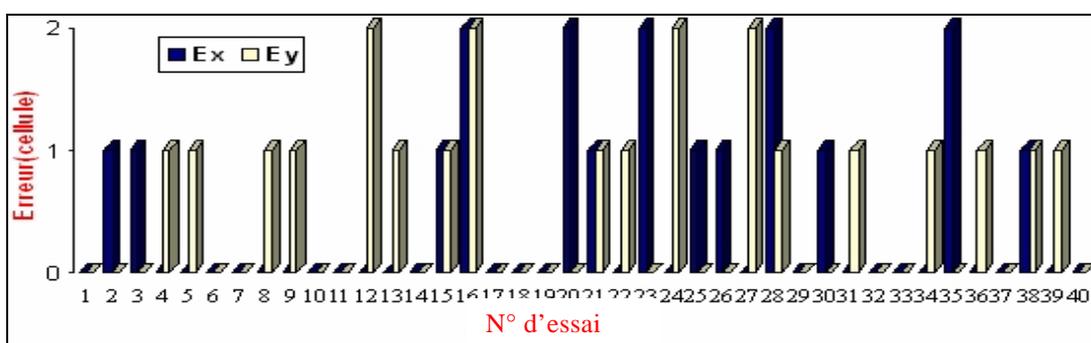


Figure 4.21 : « Erreur en position » pour le 2^{ème} exemple en utilisant les systèmes flous (100x70)

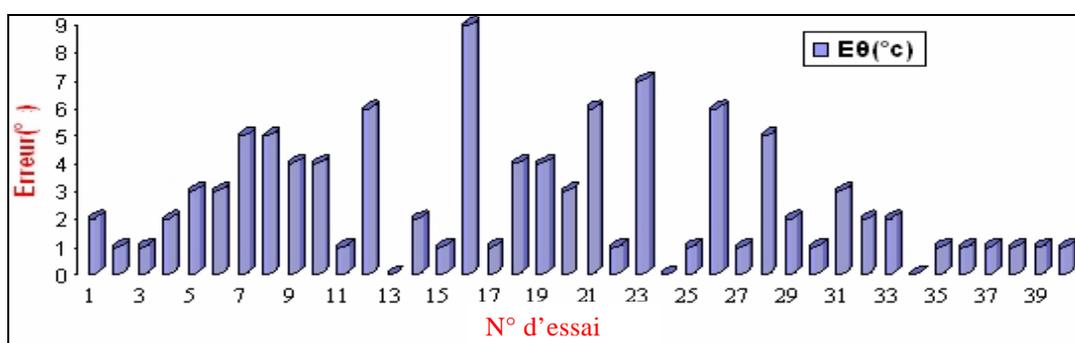


Figure 4.22 : « Erreur en orientation » pour le 2^{ème} exemple en utilisant les systèmes flous (100x70)

4.3.3 Exemple3

Dans le troisième exemple, nous avons essayé de rendre le plan de l'environnement plus grand et plus compliqué car la taille de la pièce joue un rôle très important dans la précision des résultats et le temps d'exécution nécessaire. la grille globale correspondante à cette pièce à une taille de (200x200) cellules avec une résolution de 5 cm par cellule. Nous montrons dans la figure 4.24 un exemple d'exécution.

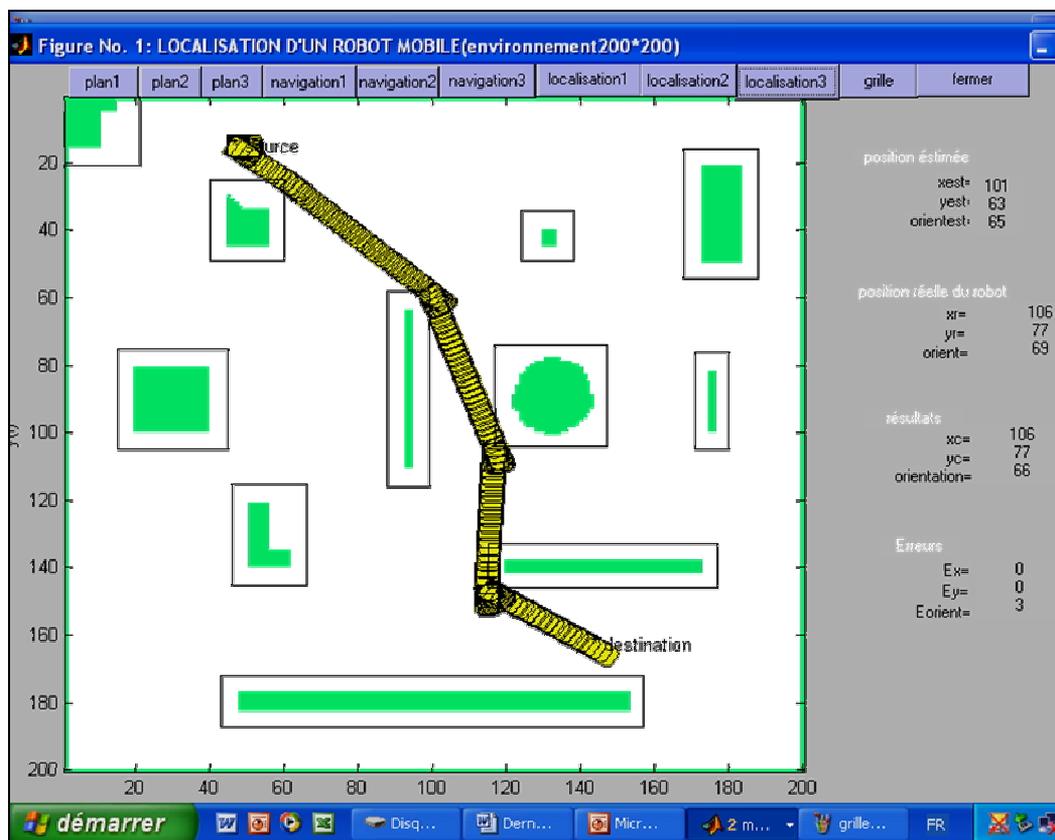


Figure 4.23 : Environnement de 200x200 cellules

La figure 4.25 représente deux trajectoires, l'une prise réellement par le robot, la deuxième estimée dans le cas où on prend une vitesse constante, ces deux cas de figures considèrent ce dernier plan d'environnement (figure 4.24).

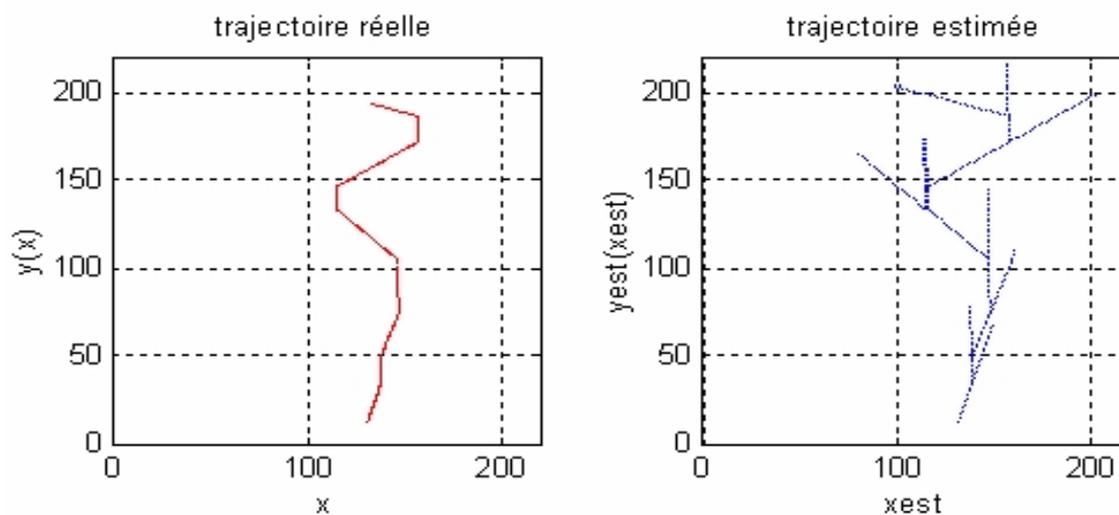


Figure 4.24 : Exemple de trajectoire réelle parcourue par le robot et la trajectoire estimée correspondante (environnement 200x200 cellules)

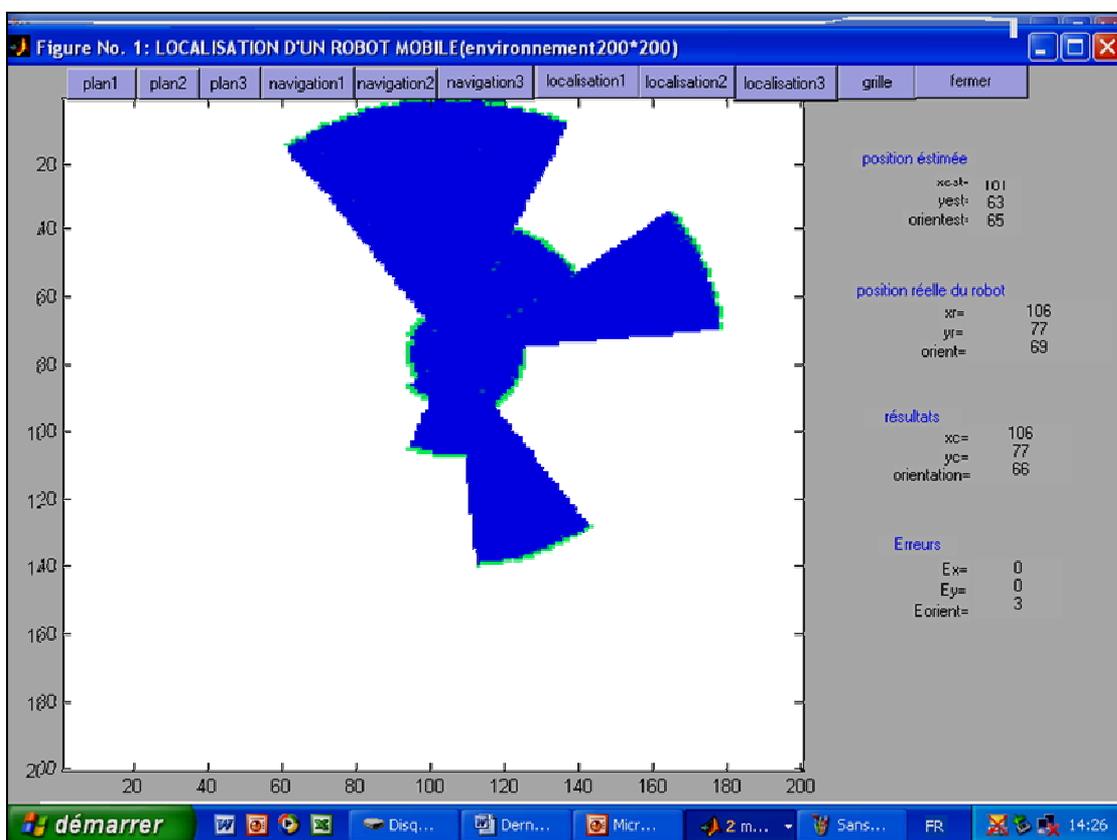


Figure 4.25 : la grille locale construite à l'arrêt du Robot (environnement 200x200cellules)

Les résultats obtenus dans cet environnement sont illustrés par les figures suivantes. Ce dernier montre clairement que ces erreurs sont largement acceptables pour un fauteuil électrique vu son application.

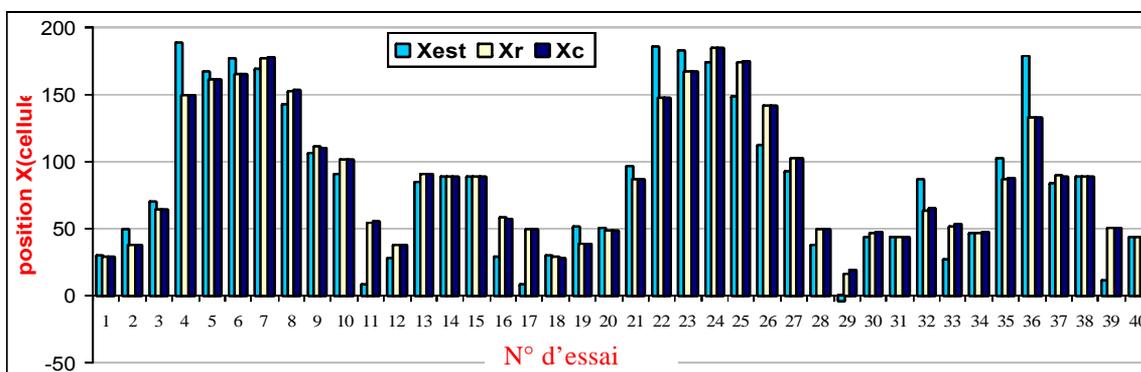


Figure 4.26 : Les résultats de la « position en X » pour le 3^{ème} exemple en utilisant les systèmes flous (200x200)

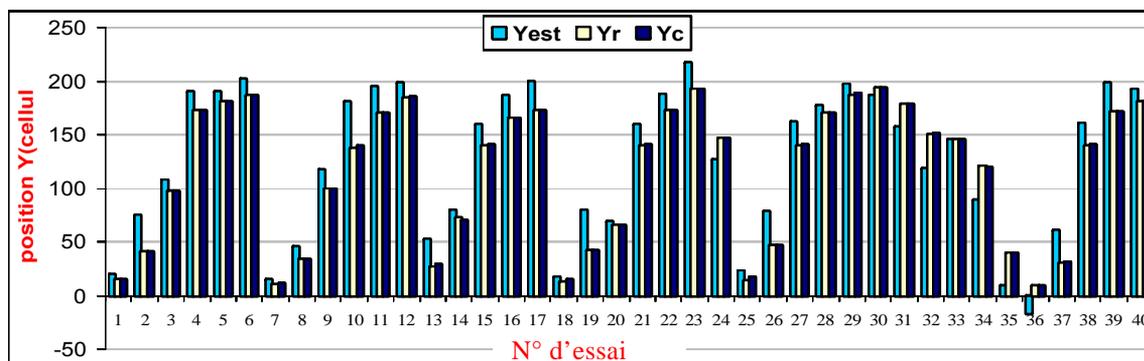


Figure 4.27 : Les résultats de la « position en Y » pour le 3^{ème} exemple en utilisant les systèmes flous (200x200)

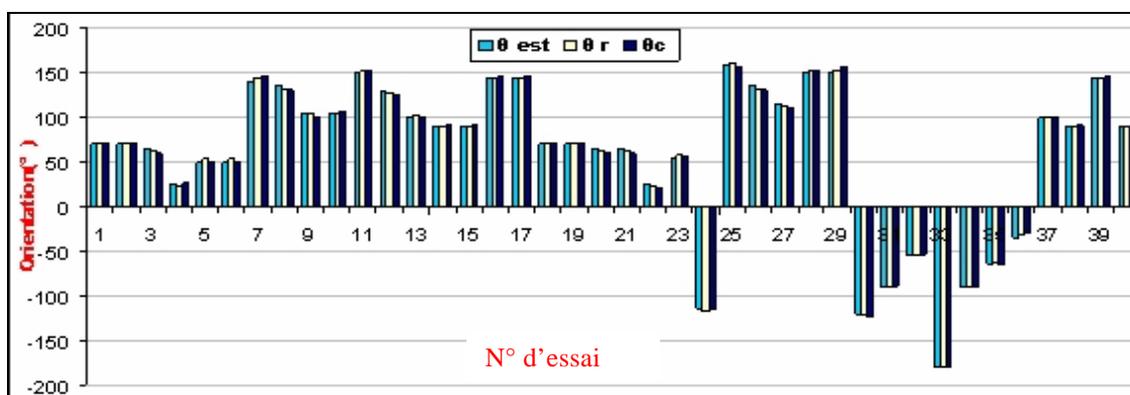


Figure 4.28 : Les résultats de « l'orientation » pour le 3^{ème} exemple en utilisant les systèmes flous (200x200)

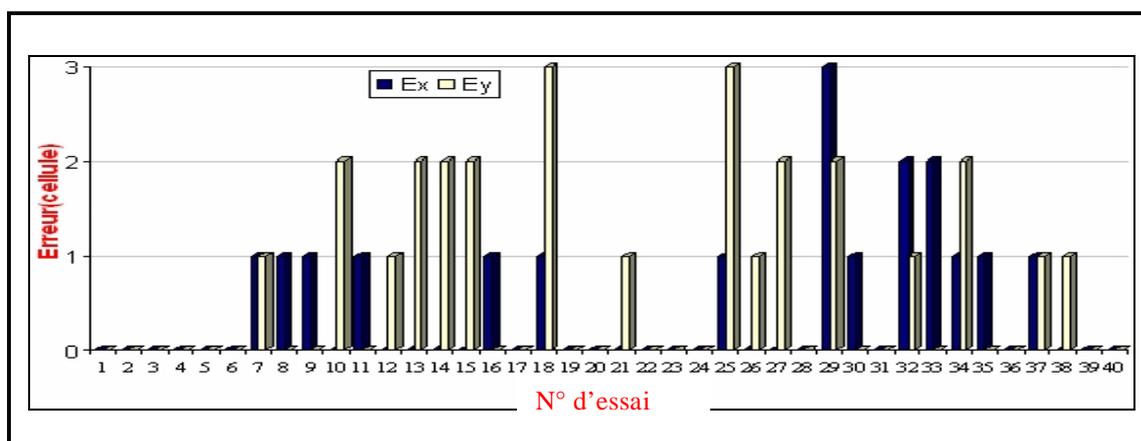


Figure 4.29 : « Erreur en position » pour le 3^{ème} exemple en utilisant les systèmes flous (200x200)

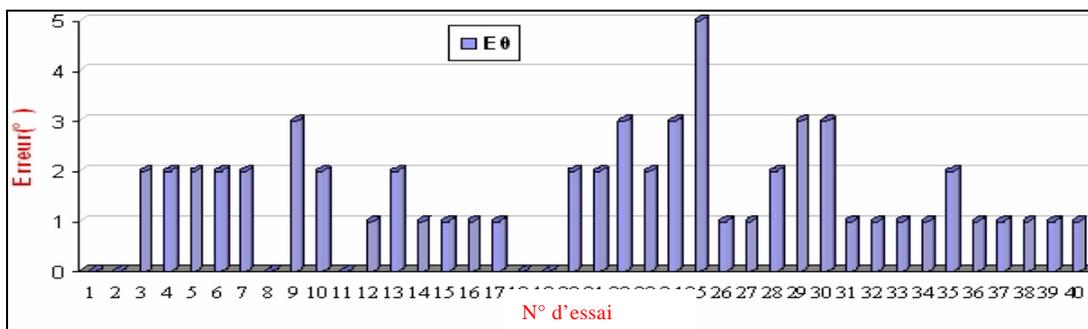


Figure 4.30 : « Erreur en orientation » pour le 3^{ème} exemple en utilisant les systèmes flous (200x200)

4.4 Conclusion

Dans ce chapitre nous avons expliqué comment nous avons réduit le temps de calcul par une procédure de recherche hiérarchisée qui consiste à réaliser la localisation sur plusieurs niveaux de grilles à différentes résolutions. Nous avons également présenté les résultats de simulation.

Les corrections que nous avons portées sur le localisateur par mise en correspondance des grilles d'occupation développé par Alain Courcelle dont [16], [17] et [28] se résument dans l'intégration de deux systèmes flous.

Les systèmes flous que nous avons développé offrent plus de flexibilité à la zone de recherche et la position estimée afin d'avoir la capacité d'être variable et mieux adapté à la taille de l'environnement.

Enfin, on peut dire que les résultats obtenus sont largement acceptables pour toutes les applications qui peuvent être envisagés pour un fauteuil électrique.

CONCLUSION

L'objectif de ce travail était d'apporter une contribution à la localisation de robot mobile dans un environnement intérieur modélisé par les grilles d'occupation grâce à une ceinture de 12 capteurs à ultrasons. Pour cela, l'approche consistait à optimiser les performances de chaque étape composant le processus de localisation dans ce type d'approche. Au terme de notre travail nous pouvons tirer les conclusions suivantes :

- Sur le mode de représentation des mesures et de l'environnement ; nous avons opté pour les grilles d'occupation qui permettent de répondre aux caractéristiques de la méthode de localisation utilisée qui sont : l'utilisation dans tout type d'environnement intérieur, réalisation quelque soit l'emplacement du robot et l'emploi des mesures de capteurs à ultrasons.
- Sur la méthode de localisation : l'objectif fixé dans cette partie était de développer un algorithme de recherche de la bonne mise en correspondance entre deux grilles : la première représentant l'environnement de travail, la seconde décrivant l'espace du robot perçu par les capteurs.
La meilleure correspondance est évaluée par la translation et la rotation d'une grille par rapport à l'autre au voisinage d'une position estimée du robot mobile. Cette dernière est déduite du bloc de navigation.
- Sur le calcul de la position et de l'orientation : le temps écoulé dans la recherche de la position du robot dans la totalité de la grille étant trop long. Ce qui nous mis dans l'obligation de limiter la zone de recherche. A cet effet nous avons développé un système flou générant une zone de recherche qui s'auto adapte avec le déplacement

Les algorithmes ayant été développées sous Matlab, les résultats en terme de temps de calcul n'ont pu être présentés. L'efficacité de la méthode est vérifiée dans nos

essaies de simulation ou nous trouvons la position réelle du robot à partir des mesures des capteurs à ultrasons seulement.

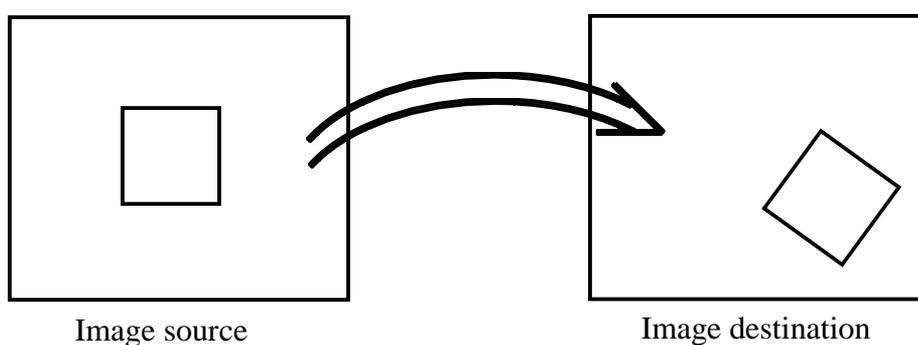
Comme perspectives de ce mémoire, d'un point de vue pratique, l'implantation des algorithmes développés sur un robot réel et avec un langage temps réel permettrait d'évaluer de façon plus précise les performances en termes de temps d'exécution.

Ensuite, concernant l'aspect théorique du travail, la diminution de la taille des grilles locales et globales, offre une réduction importante du temps de calculs, ainsi qu'une amélioration des résultats.

APPENDIX

APPENDIX 1 : Transformation en coordonnées homogènes

Les coordonnées homogènes fournissent une notation uniforme pour les transformations. Par exemple, les transformations dans un plan sont décrites par une matrice homogène 3 x 3.

1. Translation

$$\begin{aligned} X' &= X + a \\ Y' &= Y + b \end{aligned}$$

$$\begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & a \\ 0 & 0 & b \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

En notation tensorielle :

$$P^B = T_A^B P^A \quad \text{Pour } A, B = 1, 2, 3.$$

Donc T_A^B est une transformation du repère A vers le repère B.
Les indices permettent de noter les repères.

2. Rotation

Repère main droite, rotation sens trigonométrique

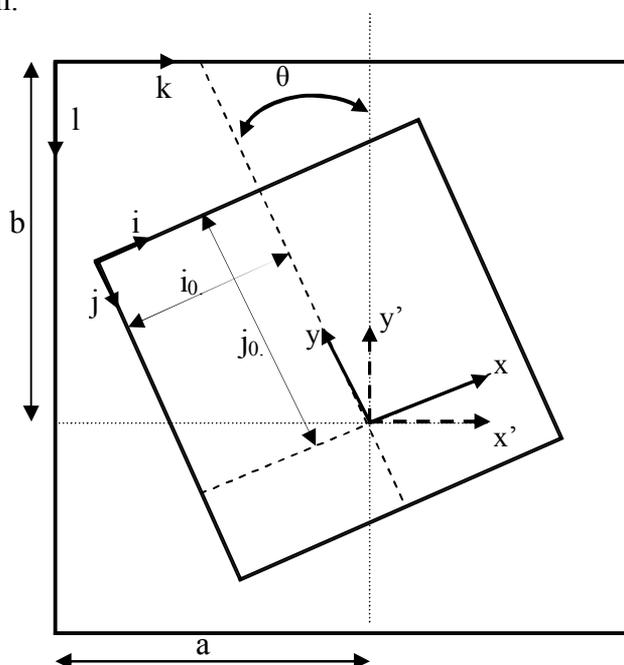
$$\begin{aligned} X' &= \cos(\theta)X - \sin(\theta)Y \\ Y' &= \sin(\theta)X + \cos(\theta)Y \end{aligned}$$

$$\begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

3. Transformations d'échelle, rotation et translation

Une transformation de similitude d'une image est définie par une rotation, une translation et un changement d'échelle.

Soit un changement d'échelle $(i - i_0)$ et $(j - j_0)$ suivi d'une rotation d'angle θ dans le plan de l'image source, suivi d'une translation a, b s'exprimes dans le repère de la destination.



Par le passage du changement d'échelle : $x = i - i_0$
 $y = j - j_0$

En faisant les trois transformation on aura :

$$\begin{aligned} k &= a + (i - i_0) \cdot \cos(\theta) + (j - j_0) \cdot \sin(\theta) \\ l &= b - (i - i_0) \cdot \sin(\theta) + (j - j_0) \cdot \cos(\theta) \end{aligned}$$

Ou bien :

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} (i - i_0) \cos(\theta) & (j - j_0) \sin(\theta) & a \\ -(i - i_0) \sin(\theta) & (j - j_0) \cos(\theta) & b \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

APPENDIX 2 : Formalisme de logique floue

1. La théorie des sous-ensembles floues

Dans un univers de discours U (ensemble de référence qui contient tous les éléments x qui appartiennent à un contexte spécifié), un sous-ensemble E de ce référentiel U est caractérisé par une fonction caractéristique μ_E de U dans $\{0, 1\}$, qui

est définie comme :
$$\mu_E = \begin{cases} 1 & \text{si } x \in E \\ 0 & \text{si } x \notin E \end{cases}$$

En fait, un sous-ensemble flou F est formellement défini par sa fonction d'appartenance μ_F , qui prend ses valeurs dans l'intervalle des nombres réels $[0, 1]$. Cette fonction d'appartenance indique le degré d'appartenance de l'élément x à F . Donc, l'ensemble flou F est défini comme suit : $F = \{(x, \mu_F(x)), x \in U\}$. De même, on utilise la

notation suivante :
$$F = \mu_F(x_1)/x_1 + \mu_F(x_2)/x_2 + \dots + \mu_F(x_n)/x_n = \sum_{i=1}^n \mu_F(x_i)/x_i$$

Où : $\mu_F(x_i)$: est le degré d'appartenance de l'élément x_i dans F , les signes / et + représentent la séparation et l'union respectivement. Pour un ensemble flou continu F dans l'ensemble de référence U , on utilise la notation suivante :
$$F = \int_U \mu_F(x)/x$$

où : la fonction d'appartenance est exprimée sous une forme analytique.

2. Différentes formes de fonctions d'appartenance

Chaque ensemble flou peut être représenté par sa fonction d'appartenance. Si l'ensemble de référence est discret, les fonctions d'appartenance sont des valeurs discrètes de l'intervalle $[0, 1]$. Si l'univers du discours est un ensemble continu, nous pouvons représenter ces valeurs analytiquement, comme fonctions d'appartenance.

En général, la forme des fonctions d'appartenance dépend de l'application, et peut être monotone (croissante ou décroissante), triangulaire, trapézoïdale ou en forme de cloche.

3. Opérations sur les ensembles flous

- Egalité : Deux sous-ensembles flous A et B de U sont égaux si leurs fonctions d'appartenance prennent la même valeur pour tout élément de U:

$$\forall x \in U, \mu_A(x) = \mu_B(x)$$

- Inclusion : Etant donné deux sous ensembles flous A et B de U, on dit que A est inclus dans B et on note: $A \subseteq B$, si leurs fonctions d'appartenance sont telles que:

$$\forall x \in U, \mu_A(x) \leq \mu_B(x)$$

- Intersection : L'intersection de deux sous ensembles flous A et B de U est le sous ensemble flou, $A \cap B$ tel que: $\forall x \in U, \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$

La théorie des ensembles flous donne une très large gamme d'opérateurs différents qui peuvent être utilisés pour l'intersection. Ils s'appellent T-normes. Les opérateurs les plus importants pour l'intersection sont:

Opérateur min: $\min(\mu_A(x), \mu_B(x))$

Opérateur algébrique: $\mu_A(x) \cdot \mu_B(x)$

Produit borné: $\max(0, \mu_A(x) + \mu_B(x) - 1)$

Produit drastique: $\begin{cases} \mu_A(x) & \text{si } \mu_B(x) = 1 \\ \mu_B(x) & \text{si } \mu_A(x) = 1 \\ 1 & \text{si } \mu_B(x), \mu_A(x) < 1 \end{cases}$

Produit d'Einstein: $\frac{\mu_A(x) \cdot \mu_B(x)}{2 - [\mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)]}$

Produit d'Hamasher: $\frac{\mu_A(x) \cdot \mu_B(x)}{\mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)}$

- Union : L'union de deux sous-ensembles flous A et B de U est le sous-ensemble flou $A \cup B$, tel que: $\forall x \in U, \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$

Pour l'union de deux sous-ensembles flous, nous utilisons les opérateurs T-conormes.

Dans la littérature, on rencontre le plus souvent:

Opérateur max: $\max(\mu_A(x), \mu_B(x))$

Somme algébrique: $\mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$

Somme bornée: $\min(1, \mu_A(x) + \mu_B(x))$

Somme drastique: $\begin{cases} \mu_A(x) & \text{si } \mu_B(x) = 0 \\ \mu_B(x) & \text{si } \mu_A(x) = 0 \\ 1 & \text{si } \mu_B(x), \mu_A(x) > 0 \end{cases}$

Somme d'Einstein: $\frac{\mu_A(x) + \mu_B(x)}{1 + \mu_A(x) \cdot \mu_B(x)}$

Produit d'Hamasher: $\frac{\mu_A(x) + \mu_B(x) - 2 \cdot \mu_A(x) \cdot \mu_B(x)}{1 - \mu_A(x) \cdot \mu_B(x)}$

Somme disjonctive: $\max(\min(\mu_A(x), 1 - \mu_B(x)), \min(1 - \mu_A(x), \mu_B(x)))$

- Complément : Le complément d'un sous ensemble flou A de U est le sous ensemble flou A de U de fonction d'appartenance: $\forall x \in U, \mu_{\bar{A}}(x) = 1 - \mu_A(x)$

4. Propriétés des opérations sur les ensembles flous

Si l'on suppose que A, B et C sont trois ensembles flous dans un ensemble de référence U, les propriétés valables dans la théorie des ensembles flous sont les suivants:

Transitivité: si $A \subseteq B \subseteq C$ alors $A \subseteq C$

Commutativité: $A \cap B = B \cap A$

$$A \cup B = B \cup A$$

Associativité: $A \cap (B \cap C) = (A \cap B) \cap C$

$$A \cup (B \cup C) = (A \cup B) \cup C$$

Distributivité: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Idempotence: $A \cup A = A$

$$A \cap A = A$$

Absorption: $A \cap (A \cup B) = A$

$$A \cup (A \cap B) = A$$

Identité: $A \cap \phi = \phi$

$$A \cap U = A$$

$$A \cup \phi = A$$

$$A \cup U = U$$

Lois de Morgan:

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$A \cap \overline{A} = \phi$$

$$A \cup \overline{A} = U$$

5. Relations floues

Une relation floue R entre deux ensembles de référence U et V est un ensemble flou de $U \times V$, de fonction d'appartenance $\mu_R (\mu_R : U \times V \rightarrow [0, 1])$. Donc la relation R est définie comme suit: $R = \{(u, v), \mu_R(u, v)\}, u \in U \text{ et } v \in V\}$

Si $U = V$, une relation floue définie sur les deux univers U et V est une relation binaire floue définie sur U.

5.1 Inverse d'une relation floue

L'inverse de la relation floue R entre U et V est la relation R^{-1} entre V et U définis par :

$$\forall v \in V, \forall u \in U, \mu_{R^{-1}}(v, u) = \mu_R(u, v)$$

5.2. Opérations sur les relations floues

▪ Intersection

L'intersection de deux relations floues R_1 et R_2 sur $U \times V$ est la relation floue $R_1 \cap R_2$ tel que:

$$\forall (u,v) \in U \times V: \mu_{R_1 \cap R_2}(u,v) = \min(\mu_{R_1}(u,v), \mu_{R_2}(u,v))$$

▪ L'union

L'union de deux relations floues R_1 et R_2 sur $U \times V$ est la relation floue $R_1 \cup R_2$ tel que:

$$\forall (u,v) \in U \times V: \mu_{R_1 \cup R_2}(u,v) = \max(\mu_{R_1}(u,v), \mu_{R_2}(u,v))$$

▪ Composition

La composition de deux relations floues R sur $U \times V$ et S sur $U \times W$ est une relation floue $R \circ S$ sur $U \times W$ de fonction d'appartenance définie par:

$$\forall (u,w) \in U \times W: \mu_{R \circ S}(u,v) = \max_v (\min(\mu_R(u,v), \mu_S(v,w)))$$

En particulier, si A est une relation unaire floue sur U et R est une relation binaire floue sur $U \times V$, on peut définir de la même façon la relation floue $A \circ R$ par:

$$\forall u \in V: \mu_{A \circ R}(v) = \max_u (\min(\mu_A(u), \mu_R(u,v)))$$

5.3. Propriétés des relations floues

Soit R une relation binaire floue sur U , cette relation est:

Associative $\forall (u,v) \in U \times U: \mu_{R \circ (R \circ R)}(u,v) = \mu_{(R \circ R) \circ R}(u,v)$

Symétrique: si elle vérifie $\forall (u,v) \in U \times U: \mu_{R \circ (R \circ R)}(u,v) = \mu_{(R \circ R) \circ R}(u,v)$

Antisymétrique: si elle vérifie $\forall (u,v) \in U \times U: (\mu_R(u,v) > 0 \text{ et } \mu_R(v,u) > 0) \Rightarrow (u=v)$

Réflexive: si elle vérifie $\forall u \in U, \mu_R(u,u) = 1$

Antiréflexive: si elle vérifie $\forall u \in U, \mu_R(u,u) = 0$

Transitive: si elle vérifie : $\forall (u,w) \in U \times U: \mu_R(u,w) \geq \max_v (\min(\mu_R(u,v), \mu_R(v,w)))$

6. Raisonnement en logique floue

Le raisonnement en environnement flou est un raisonnement approximatif du fait que l'on admet que des énoncés soient partiellement vrais et qu'on puisse raisonner plus souplement sur des connaissances imprécises et/ou incertaines qu'on ne pourrait le faire en logique classique.

Nous allons donc faire appel des éléments concernant le raisonnement en environnement flou.

7. Variable linguistique

Une variable linguistique représente un état dans le système à régler ou une variable de réglage dans un contrôleur flou. Sa valeur est définie dans des termes linguistiques qui peuvent être des mots ou des phrases d'un langage naturel ou artificiel. Chaque variable linguistique est caractérisée par un ensemble tel que: $\{x, T(x), U, G, M\}$

Où: x : est le nom de la variable,

$T(x)$: est l'ensemble des valeurs linguistiques que peut prendre x ,

U : est l'univers du discours associé avec la valeur de base,

G : est la règle syntaxique pour générer les valeurs linguistiques de x ,

M : est la règle sémantique pour associer un sens à chaque valeur linguistique.

8. Propositions floues

▪ Proposition floue élémentaire

Etant donné un ensemble L de variables linguistiques, une proposition floue élémentaire est définie à partir d'une variable linguistique $\{x, T(x), U, G, M\}$ de L par la qualification « x est A », où A est une caractérisation floue appartenant à $T(x)$ ou à $M(T(x))$.

▪ Proposition floue générale

Une proposition floue générale est obtenue à partir de propositions floues élémentaires en utilisant les opérations suivantes: - NON (négation)

- ET (conjonction)

- OU (disjonction)

- SI ALORS (implication).

Donc, une proposition floue générale peut prendre les formes suivantes:

- Si (x est A) alors (y est B)
- Si [(x est A) et (y est B)] alors (z est C)
- Si [(x est A) ou (y est B)] alors (z est C)
- Si [(x est A) et (y est non B)] alors (z est C)

9. Règle linguistique

L'idée principale des systèmes basés sur la logique floue est d'exprimer la connaissance humaine sous la forme de règles linguistiques de forme SI ALORS. ., chaque règle a deux parties:

- La partie antécédente (prémisse ou condition) : est la description de l'état du système, elle est exprimée par Si
- La partie conséquente (conclusion): exprime l'action que l'opérateur qui contrôle le système doit exécuter, elle est exprimée par Alors,. Chaque règle floue est basée sur l'implication floue.

Il y a plusieurs formes de règles Si Alors, la forme générale est :

Si (proposition floue1) alors (proposition floue 2)

10. Inférence floue

L'inférence floue est basée généralement sur l'implication floue de type modus ponens généralisé (GMP) qui a la forme suivante:

Prémisse1 : x est A'

Prémisse2 : Si x est A alors y est B

Conséquence : y est B'

Où x est y sont des variables linguistiques : A, A', B et B' sont les valeurs linguistiques définies dans de discours de x et y.

REFERENCE

1. G.C.Anousaki, K.J.Kyriakopoulos « Simultaneous Localization And Map Building For Mobile Robot Navigation », IEEE transactions on robotics and automation, 1999.
2. O.Ait-aider “localisation référencée modèle d’un robot mobile d’intérieur », thèse de doctorat 2002.
3. N.Bennila « Contribution Au Développement D’un Robot Mobile Semi Autonome Et Simulation D’un Planificateur Réactif De Trajectoires » thèse de magister, 1995.
4. M. betke and L. Gurvits, « Mobile robot localization using landmarks», IEEE Trans. on Robotics and Automation, Vol. 13, N°2, pp. 251-263, 1998.
5. J.Borenstein, Y.Koren « The Vector Field Histogram- Fast Obstacle- Avoidance For Mobile Robots», IEEE transactions on robotics and automation, vol 7, n°3, June1991, pp.278-288
6. J. Borenstein, H.R. Everett, L. Feng, D.Wehe, “Mobile Robot Positioning and Sensors and Techniques”, Journal of Robotic Systems, Special Issue on Mobile Robots. Vol. 14 No. 4, pp. 231 – 249, 1997.
7. B.C.Valério « Modelisation De L’environnement Par Grille Adaptative Et Recherche De Chemin Pour Robot Mobile », Thèse d’état, 1990.
8. Botton, S.Diquenne, F.Egels, Y.Even, M.Willis, Pascal, “GPS, Localisation Et Navigation”-Hermes, Paris, 1997.
9. R.Brown, B.Donald “Mobile Robot Self-Localization Without Explicit Landmarks”. Algorithmica,november15,1996 .

10. W.Burgard, D.Fox, D.Henning, T.Schmidt, "Estimating The Absolute Position Of A Mobile Robot Using Position Probability Grids" .In proc.of the fourteenth national conference on artificial intelligence, pages 896-901,1996 .
11. R. Chatila and J.P. Laumond, « Position referencing and consistent world modeling for mobile robots », Proc of the IEEE International Conference on Robotics and Automation (ICRA85), pp. 138-145, 1985.
12. F.Chavand "Perception De L'environnement En Robotique", Edition Hermes, Paris, 1998.
13. J.Crowley "Navigation Et Modélisation Pour Un Robot Mobile", rapport de recherche, LIFIA ,1985.
14. J.Crowley "Navigation For An Intelligent Mobile Robot" .IEEE journal of robotics and automation , vol RA1, n°1, March 1985 .
15. J.Crowley "World Modeling And Position Estimation For A Mobile Robot Using Ultrasonic Ranging" , LIFIA (IMG), 1989 .
16. A.Courcelle, O.Horn, « Ultrasonic Data Representation : Application To Mobile Robots Localisation », Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Victoria, B.C., Canada, 13- 16 Octobre 1998, pp 1559-1564.
17. A.Courcelle, "Localisation D'un Robot Mobile : Application À L'aide à La Mobilité Des Personnes Handicapées Moteur", Thèse de l'Université de METZ, 14 janvier 2000, 141 pages.
18. J.Courtney, A.Jain: "Mobile Robot Localization Via Classification Of Multisensor Maps", proceedings of the IEEE international conference on robotics and automation, 1994, pp 1672-1678.

19. Cox “ Blanche –An Experimental In Guidance And Navigation Of An Autonomous Robot Vehicle” . IEEE transactions on robotics and automation vol 7,n°2 , April, 1991.
20. M.Drumheller, “Mobile Robot Localization Using Sonar”, IEEE Transactions on pattern analysis and machine intelligence, vol 9,n°2,march 1987,pp.325-332
21. Hans , Moravec et Elfes “High Resolution Maps From Wide Angle Sonar”. In proc.IEEE int. Conf . Robotics and automation, pages 116-121,1985
22. J. Gasós, A. Martín, “Mobile Robot Localization Using Fuzzy Maps”, Lecture Notes in Artificial Intelligence, A. Ralescu & T. Martin (Eds.), Springer Verlag, 1996, to appear.
23. Gutmann et al “AMOS: Comparison Of Scan Matching Approachs For Self-Localization In Indoor Environment” .In proceeding of the Euromicro workshop on advanced mobile robots, pages 61-67, IEEE , 1996 .
24. J.Gutmann, W. Burgard, D.Fox, K.Konolige “An Experimental Comparison Of Localization Methods” .In proceeding of the international conference on intelligent robots and systems IRO98), IEEE \RSJ ,1998 .
25. G. Hager and M. Mintz, « Sensor modeling and robust sensor data fusion », proc. Of the International Symposium on Robotics Research, pp.69-74, 1990.
26. P. Hoppenot, « Contribution de la robotique mobile à l'assistance aux personnes handicapées », Thèse de doctorat, Université d'Evry Val d'Essonne (EVE), 27 nov. 1997.
27. P.HOPPENOT. E.COLLE « Robotique D'assistance Aux Personnes Handicapées », Handicap'2000, Paris, pp. 11-16, juin, 2000.
28. O.Horn, A.Courcelle « Localisation Statique D'un Fauteuil Électrique Autonome », RAIRO-APII-JESA, Journal Européen des Systèmes Automatisés Volume 31 n°5/1997, pp 867-878.

29. P. Hoppenot, E. Colle, « Real-time localisation of a low-cost mobile robot with poor ultrasonic data », IFAC journal, Control Engineering practice, vol. 6, pp.925-934, 1998.
30. P.Jensfelt “Approches To Mobile Robot Localization In Indoor Environments” , Doctorat thesis, Department of signals, sensors and systems, Royal Institute of Technology,2001.
31. R. Kuc and M. Siegel, « Physically based simulation model for acoustic sensor robot navigation’ , IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol 9, N°6, 1987, pp. 766-778
32. J.C. Latombe “Robot Motion Planing”, Kluwer Academic Publishers, Boston, MA, 1991.
33. J.P.Laumond “La Robotique Mobile”, Edition Hermes Science Europe Ltd, Paris, 2001.
34. J.Leopoldo et al “Development And Experimental Validation Of An Adaptive Extended Kalman Filter For Localization Of Mobile Robots”. IEEE transactions on robotics and automation, vol 15, n°2, April 1999 .
35. J.Leonard, H.F.Durrant-Whyte, “Simultaneous Map Building Localization For An Autonomous Mobile Robot” ,Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on Published: 1991 , Page(s): 1442 -1447 vol.3.
36. J. Leonard , H. Durrant-White, « Mobile robot localization by tracking geometric beacons », IEEE Trans on Robotics and Automation, Vol 7, N° 3, pp. 376-382, 1991.
37. P. Moutarlier , R. Chatila, « Stochastic multisensory data fusion for mobile robot location and environmental modelling », Proc. Of the International Symposium on Robotics Research, pp. 85-94, 1990.

38. Neira et al “Fusing Range And Intensity Images For Mobile Robot Localization”. IEEE transactions on robotics and automation, vol 15, n°1, February 1999 .
39. D.Pagac, E.M.Nebot, H.Durrant-Whyte, “An Evidential Approach To Probabilistic”, proceedings of the IEEE International Conference on robotics and automation, April 1996, pp 745-750.
40. A. Pruski “Robotique Mobile:Planification De Trajectoire”.Edition Hermes, Paris, 1996.
41. P. Reignier, « Pilotage Réactif D’un Robot Mobile, Etude De Lien Entre La Perception Et L’action » thèse de doctorat, laboratoire d’informatique fondamentale et d’intelligence artificielle, Grenoble,1994
42. B.Schiele, J.L.Crowley “Comparison Of Position Estimation Techniques Using Occupancy Grids”, dans Proceeding of the IEEE International conference on robotics and automation (ICRA94), 1994.
43. S. Thrun and A. Bucken, « Integrating grid-based and topological maps for mobile robot navigation », In Proceedings of the Thirteenth National Conference on Artificial Intelligence, pp. 944-950, 1996
44. B.Yamauchi, “Mobile Robot Localization In Dynamic Environments Using Dead Reckoning And Evidence Grids”, proceedings of the IEEE international conference on robotics and automation, 1996, Minneapolis, Minnesota, April 1996, pp 1401-1406.