

République Tunisienne  
Ministère de l'Enseignement Supérieur  
de la recherche scientifique  
Université de Sfax - École Nationale  
d'Ingénieurs de Sfax  
Thèse de DOCTORAT  
Automatique et Informatique Industrielle  
No d'ordre: 2012, 85/11



République Française  
Pôle Universités Centre Val de Loire  
Université D'Orléans  
Ecole Doctorale - Science et Technologies  
Laboratoire PRISME  
Thèse de DOCTORAT  
Automatique et Informatique Industrielle  
No d'ordre: 2012, 85/11

# THÈSE

*présentée en cotutelle aux :*

**École Nationale d'Ingénieurs de Sfax  
et Université d'Orléans**

*pour obtenir le grade de :*

**Docteur de l'Université d'Orléans et de l'Université de Sfax**

**Dans la discipline Génie Électrique**

Automatique et Informatique Industrielle

*par*

**Lobna AMOURI-JMAIEL**

---

---

## **CONTRIBUTION À LA COMMANDE ET AU PILOTAGE RÉACTIF DE ROBOTS MOBILES À ROUES**

---

---

*soutenue le 20 Février 2012, devant le jury composé de :*

|   |                    |
|---|--------------------|
| <b>MM.</b> Tarak DAMAK (Professeur à l'ENIS)                      | Président          |
| Naceur BENHADJ BRAIEK (Professeur à l'ESSTT)                      | Rapporteur         |
| Saïd ZEGHLOUL (Professeur à l'Université de Poitiers)             | Rapporteur         |
| Gilles MOURIOUX (Maître de Conférences à l'Université de Limoges) | Examineur          |
| Mohamed JALLOULI (Maître de Conférences à l'ISSIG)                | Encadrant          |
| Cyril NOVALES (Maître de Conférences à l'Université d'Orléans)    | Encadrant          |
| Nabil DERBEL (Professeur à l'ENIS)                                | Directeur de thèse |
| Gérard POISSON (Professeur à l'Université d'Orléans)              | Directeur de thèse |

## Remerciements

Les travaux présentés dans cette thèse en cotutelle sont le résultat de collaboration entre la Tunisie et la France. La première partie de la thèse a été effectuée au sein du Laboratoire de recherche «Control & Energy Management Laboratory (CEMLab)», dirigé par Monsieur Abderrezek Ouali, Professeur en Génie Électrique à l'École Nationale d'Ingénieurs de Sfax. La deuxième partie de la thèse a été effectuée au sein du Laboratoire «Pluridisciplinaire de Recherche en Ingénierie des Systèmes, Mécanique, Energétique (PRISME)», dirigé par Madame Christine Rousselle.

J'adresse mes vifs remerciements à Messieurs Nabil Derbel, Professeur en Génie Électrique à l'École Nationale d'Ingénieurs de Sfax, et Monsieur Gérard Poisson, Professeur à l'Université d'Orléans responsable de l'axe Robotique du laboratoire PRISME pour m'avoir encadrée et accueillie dans leurs laboratoires de recherche et pour toute l'aide et le soutien qu'ils m'ont apportés.

Je tiens à remercier Monsieur Mohamed Jallouli, Maître de Conférences à l'Institut Supérieur des Systèmes Industriels de Gabès, et Monsieur Cyril Novalès, Maître de Conférences à l'Université d'Orléans pour toute l'aide et le soutien qu'ils m'ont apportés et pour leur patience durant la période de thèse.

À Monsieur Tarak Dammak, Professeur à l'École Nationale d'Ingénieurs de Sfax, je voudrais exprimer ma vive gratitude pour l'honneur qu'il m'a fait en acceptant de présider le jury de ma thèse de doctorat.

Je tiens à remercier Monsieur Gilles Mourioux, Maître de Conférences à l'Université de Limoges, pour avoir accepté de participer au jury de ma thèse de doctorat. Je voudrais exprimer ma reconnaissance à Monsieur Naceur BenHadj Braïek, Professeur à l'École Supérieure des Sciences et Techniques de Tunis (ESSTT), et à Monsieur Saïd Zeghloul Professeur à l'Université de Poitiers, et coordonnateur du département «Génie Mécanique et Systèmes Complexes» de PPRIME, pour avoir accepté d'examiner mes travaux de recherche et d'en être les rapporteurs.

Enfin, que tous les collègues et amis du Département de Génie Électrique, du laboratoire CEMLab et du laboratoire PRISME, trouvent ici un témoignage de mon amitié pour la solidarité et le soutien moral dont ils ont toujours fait preuve.

*to ...*

*My parents to whom I owe everything*

*to ...*

*My husband for his endless love and support*

*My sons for their kindness*

*to ...*

*All my family*

# Table des matières

|  |             |
|--|-------------|
| <b>Table des figures</b>   | <b>viii</b> |
| <b>Liste des tableaux</b>  | <b>xiii</b> |
| <b>Introduction générale</b>                                     | <b>1</b>    |
| <b>I Cadre de l'étude</b>  | <b>5</b>    |
| I.1 Modélisation des plateformes robotiques . . . . .            | 5           |
| I.1.1 Plateforme unicycle . . . . .                              | 6           |
| I.1.1.1 Premier robot unicycle : Khepera II . . . . .            | 6           |
| I.1.1.2 Deuxième robot unicycle : fauteuil roulant . . . . .     | 8           |
| I.1.2 Plateforme omnidirectionnelle . . . . .                    | 10          |
| I.1.2.1 Modélisation cinématique . . . . .                       | 11          |
| I.1.2.2 Modèle cinématique inverse (MCI) . . . . .               | 13          |
| I.1.2.3 Modèle cinématique direct (MCD) . . . . .                | 14          |
| I.2 Localisation des robots mobiles . . . . .                    | 14          |
| I.2.1 Les techniques de localisation . . . . .                   | 15          |
| I.2.2 Différents types de capteurs en robotique mobile . . . . . | 16          |
| I.2.2.1 Les informations proprioceptives . . . . .               | 16          |
| I.2.2.2 Les informations extéroceptives . . . . .                | 17          |
| I.3 Les stratégies de navigation . . . . .                       | 18          |

---

|       |   |    |
|-------|---|----|
| I.3.1 | Les stratégies de navigation locale . . . . .               | 18 |
| I.3.2 | Les stratégies de navigation globale . . . . .              | 19 |
| I.3.3 | Choix de la stratégie de navigation . . . . .               | 21 |
| I.4   | Méthodes d'évitement d'obstacles . . . . .                  | 21 |
| I.4.1 | Méthode de la fenêtre dynamique . . . . .                   | 22 |
| I.4.2 | Méthode du flux optique . . . . .                           | 22 |
| I.4.3 | Méthode de la Zone de Déformation Virtuelle (ZDV) . . . . . | 23 |
| I.4.4 | Choix de la méthode d'évitement d'obstacles . . . . .       | 23 |
| I.5   | Architectures de contrôle . . . . .                         | 24 |
| I.5.1 | Contrôleurs hiérarchiques . . . . .                         | 24 |
| I.5.2 | Contrôleurs réactifs . . . . .                              | 25 |
| I.5.3 | Contrôleurs hybrides . . . . .                              | 25 |
| I.5.4 | Contrôleurs génériques . . . . .                            | 26 |
| I.5.5 | Choix de l'architecture de contrôle . . . . .               | 26 |
| I.6   | Conclusion . . . . .  | 27 |

## **II Contribution à la commande floue de deux types de robots mobiles en utilisant la localisation relative** **28**

|        |   |    |
|--------|---|----|
| II.1   | Introduction . . . . .  | 28 |
| II.2   | Synthèse du contrôleur flou pour un robot unicycle : Khepera II . . . . .       | 29 |
| II.2.1 | Choix des fonctions d'appartenance des entrées floues . . . . .                 | 31 |
| II.2.2 | Définition de la base des règles du contrôleur flou . . . . .                   | 34 |
| II.2.3 | Sorties du contrôleur flou . . . . .  | 34 |
| II.2.4 | Optimisation par la méthode du gradient . . . . .                               | 35 |
| II.2.5 | Résultats de simulation . . . . .   | 36 |
| II.2.6 | Résultats Expérimentaux . . . . .   | 37 |
| II.3   | Synthèse du contrôleur flou pour un robot unicycle : fauteuil roulant . . . . . | 38 |

---

|  |  |           |
|--|--|-----------|
| II.3.1   | Fonctions d'appartenance des entrées du contrôleur flou . . . . .        | 39        |
| II.3.2   | Définition de la base des règles du contrôleur flou . . . . .            | 40        |
| II.3.3   | Sorties du contrôleur flou . . . . .                                     | 40        |
| II.3.4   | Résultats expérimentaux . . . . .  | 41        |
| II.4   | Synthèse du contrôleur flou pour un robot omnidirectionnel : ROMNI . . . | 42        |
| II.4.1   | Fonctions d'appartenance des entrées du contrôleur flou . . . . .        | 43        |
| II.4.2   | Définition de la base des règles du contrôleur flou . . . . .            | 44        |
| II.4.3   | Sorties du contrôleur flou . . . . .                                     | 45        |
| II.4.4   | Résultats de la simulation sur le ROMNI . . . . .                        | 45        |
| II.5   | Conclusion . . . . .   | 48        |
| <b>III Contribution à la localisation absolue et la commande floue d'un robot mobile</b> |  | <b>49</b> |
| III.1  | Introduction . . . . .   | 49        |
| III.2  | Localisation absolue par une caméra . . . . .                            | 51        |
| III.2.1  | La morphologie mathématique . . . . .                                    | 51        |
| III.2.2  | Acquisition d'une image de référence . . . . .                           | 52        |
| III.2.2.1  | Niveau de gris . . . . .   | 53        |
| III.2.2.2  | Le filtrage : la technique Sobel . . . . .                               | 53        |
| III.2.2.3  | Le seuillage de l'image . . . . .  | 54        |
| III.2.2.4  | L'opérateur morphologique : érosion . . . . .                            | 55        |
| III.2.2.5  | La transformation du cercle de Hough (TCH) . . . . .                     | 56        |
| III.2.3  | Méthode de détermination de l'orientation du robot mobile . . . . .      | 57        |
| III.2.4  | Méthode de passage des coordonnées image aux coordonnées réelles         | 58        |
| III.2.5  | Méthode d'optimisation du temps de traitement des images . . . . .       | 58        |
| III.3  | Résultats Expérimentaux . . . . .  | 61        |
| III.3.1  | Première expérience . . . . .  | 62        |

---

|           |  |           |
|-----------|--|-----------|
| III.3.2   | Deuxième expérience . . . . .  | 63        |
| III.3.3   | Troisième expérience . . . . .   | 65        |
| III.4     | Interface de commande et de supervision d'un robot mobile avec Webcam .      | 66        |
| III.5     | Conclusion . . . . .   | 67        |
| <b>IV</b> | <b>Navigation référencée multi-capteurs en environnement encombré</b>        | <b>69</b> |
| IV.1      | Introduction . . . . .   | 69        |
| IV.2      | Synthèse d'un évitement d'obstacles par ZDV pour le fauteuil . . . . .       | 70        |
| IV.2.1    | Zone de sécurité non déformée . . . . .                                      | 71        |
| IV.2.2    | Zone de sécurité déformée . . . . .  | 72        |
| IV.2.3    | Déformation . . . . .  | 73        |
| IV.2.4    | Fonctions Jacobiennes . . . . .  | 73        |
| IV.3      | Synthèse d'un évitement d'obstacles par ZDV pour le ROMNI . . . . .          | 75        |
| IV.3.1    | Zone de sécurité circulaire du ROMNI . . . . .                               | 75        |
| IV.3.2    | Fonctions Jacobiennes . . . . .  | 78        |
| IV.4      | Synthèse de l'architecture de contrôle . . . . .                             | 78        |
| IV.4.1    | Architecture du système de contrôle . . . . .                                | 79        |
| IV.4.2    | Le problème de commutation : solution hystérésis . . . . .                   | 81        |
| IV.4.3    | Problème de commutation : solution floue . . . . .                           | 83        |
| IV.5      | Résultats de simulation et expérimentaux du robot unicycle : le fauteuil . . | 86        |
| IV.5.1    | Expérience 1 : trajectoire complexe . . . . .                                | 87        |
| IV.5.2    | Expérience 2 : situation de minimum local (coin) . . . . .                   | 88        |
| IV.5.3    | Expérience 3 : interaction homme-machine . . . . .                           | 89        |
| IV.6      | Résultats de simulation sur le robot omnidirectionnel : ROMNI . . . . .      | 91        |
| IV.6.1    | Expérience 1 : situation couloir . . . . .                                   | 92        |
| IV.6.2    | Expérience 2 : obstacles de formes différentes . . . . .                     | 93        |
| IV.7      | Conclusion . . . . .   | 94        |

---

|   |            |
|---|------------|
| <b>Conclusions et Perspectives</b>                      | <b>96</b>  |
| <b>Bibliographie</b>                                    | <b>98</b>  |
| <b>Annexes</b>  | <b>104</b> |
| 1 Spécifications du capteur laser . . . . .             | 104        |
| 2 Protocole de communication du capteur laser . . . . . | 104        |
| 3 Programmes d'acquisition du capteur laser . . . . .   | 107        |

# Table des figures

|      |   |    |
|------|---|----|
| I.1  | Schéma du robot mobile Khepera II . . . . .   | 6  |
| I.2  | Schéma cinématique du robot mobile à deux roues motrices indépendantes  | 7  |
| I.3  | Configuration courante du fauteuil roulant. La partie logicielle sur l'ordinateur portable utilise les mesures du module CB405 et le capteur laser pour contrôler les mouvements du fauteuil . . . . .                            | 9  |
| I.4  | Diagramme décrivant les méthodes d'interaction de l'utilisateur : à travers le joystick ou l'ordinateur portable. Toutes les commandes du joystick sont traitées avec le module «CB-405» avant d'être envoyées aux moteurs. . . . | 9  |
| I.5  | Représentation du robot omnidirectionnel ROMNI (Photo PRISME) . . .   | 10 |
| I.6  | Représentation d'un essieu (Thèse Mouriaux 2006[3]) . . . . .   | 11 |
| I.7  | Schéma cinématique du robot omnidirectionnel ROMNI (Thèse Mouriaux [3]). . . . .  | 11 |
| I.8  | Repère absolu, repère robot et repère essieu (vue 2D de dessus). . . . .  | 12 |
| I.9  | GPS de type Navilock-NL-464US – GPS – USB – 2 – 0 – Stick (Photo de la société Navilock). . . . .   | 16 |
| I.10 | Télémètre Laser de type URG-04X (Photo de la société Hokuyo Automatic Co., LTD). . . . .  | 18 |
| I.11 | Action associée à un lieu (Trullier [7]) . . . . .  | 19 |
| I.12 | Navigation topologique (Trullier [7]) . . . . .   | 20 |
| I.13 | Navigation métrique (Trullier [7]) . . . . .  | 20 |
| I.14 | Détection du flux optique (Projet FP6-IST-045248 [21]) . . . . .  | 23 |
| I.15 | Détection d'un obstacle en mouvement (Projet FP6-IST-045248 [21]) . . .   | 23 |
| I.16 | Architecture hiérarchique (Arkin [25]). . . . .   | 24 |

---

|       |   |    |
|-------|---|----|
| I.17  | Architecture réactive (Arkin [25]) . . . . .  | 25 |
| I.18  | Architecture hybride (Arkin [25]) . . . . .   | 26 |
| II.1  | Configuration du robot par rapport à un point objectif. . . . .   | 30 |
| II.2  | Schéma de commande d'un robot mobile Khepera II par un contrôleur flou optimisé. . . . .  | 31 |
| II.3  | Fonction d'appartenance de type triangulaire : $\text{triang}(a,b,c)$ . . . . .   | 32 |
| II.4  | Fonction d'appartenance de type gaussiennes : $\text{gauss}(x,250,200)$ . . . . .   | 32 |
| II.5  | Fonctions d'appartenances triangulaires de la distance $d$ . . . . .  | 32 |
| II.6  | Fonctions d'appartenances gaussiennes de la distance $d$ . . . . .  | 32 |
| II.7  | Fonctions d'appartenances gaussiennes de l'angle $\varphi$ . . . . .  | 33 |
| II.8  | Fonctions d'appartenances triangulaires de l'angle $\varphi$ . . . . .  | 33 |
| II.9  | Trajectoire avec les fonctions gaussiennes (trait continu), trajectoire avec les fonctions triangulaires (trait discontinu) . . . . . | 33 |
| II.10 | Variables de sorties des vitesses du robot Khepera II. . . . .  | 35 |
| II.11 | Différentes trajectoires du robot mobile pour le même point origine (0,0) avec différentes orientations. . . . .                      | 36 |
| II.12 | Vitesse droite et gauche pour une trajectoire partant de (0,0) à (0,300). . . . .   | 36 |
| II.13 | Evolution de la distance à la cible en fonction du numéro d'itération . . . . .   | 37 |
| II.14 | Environnement expérimental du robot mobile Khepera II. . . . .  | 38 |
| II.15 | Trajectoires expérimentales obtenues. En vert et noir : avec optimisation ; en bleu : sans optimisation. . . . .                      | 38 |
| II.16 | Les fonctions d'appartenance de la distance (d). . . . .  | 39 |
| II.17 | Fonctions d'appartenance de l'angle $\varphi$ du fauteuil. . . . .  | 39 |
| II.18 | Variables de sorties ( $V_g$ et $V_d$ ) du fauteuil roulant. . . . .  | 40 |
| II.19 | Trajectoire optimisée parcourue par le fauteuil roulant (la trajectoire est celle du centre d'essieu) . . . . .                       | 41 |
| II.20 | Vitesses réelle et théorique du fauteuil roulant . . . . .  | 42 |
| II.21 | Configuration du ROMNI par rapport à un point objectif. . . . .   | 42 |
| II.22 | Fonctions d'appartenance de type gaussiennes de la distance d. . . . .  | 43 |

---

|  |    |
|--|----|
| II.23 Fonctions d'appartenance de l'angle $\varphi$ . . . . .                                    | 44 |
| II.24 Variables de sorties ( $\varphi_1$ , $\varphi_2$ et $\varphi_3$ ) du ROMNI. . . . .        | 45 |
| II.25 Position du robot avec $\varphi = 0$ . . . . .   | 46 |
| II.26 Variation des vitesses pour $\varphi = 0$ . . . . .  | 46 |
| II.27 Position du robot avec $\varphi = 90^\circ$ . . . . .                                      | 47 |
| II.28 Variation des vitesses pour $\varphi = 90^\circ$ . . . . .                                 | 47 |
| II.29 Position du robot avec $\varphi = 180^\circ$ . . . . .                                     | 47 |
| II.30 Variation des vitesses pour $\varphi = 180^\circ$ . . . . .                                | 47 |
|  |    |
| III.1 Image de référence contenant les 4 étiquettes avant traitement. . . . .                    | 53 |
| III.2 Image de référence après binarisation. . . . .   | 53 |
| III.3 Image de référence en entrée. . . . .  | 54 |
| III.4 Image de référence après détection de bords avec le filtre «Sobel». . . . .                | 54 |
| III.5 Image de référence en entrée. . . . .  | 55 |
| III.6 Image de référence après seuillage. . . . .  | 55 |
| III.7 Image de référence après érosion . . . . .   | 55 |
| III.8 Image de référence après l'application de la TCH . . . . .                                 | 56 |
| III.9 Une image qui localise le robot après application du TCH . . . . .                         | 57 |
| III.10 Positionnement d'une étiquette blanche sur l'axe des abscisses . . . . .                  | 57 |
| III.11 Image réelle du robot avec l'étiquette blanche. . . . .                                   | 58 |
| III.12 Image binaire montrant les centres de gravité du robot et de l'étiquette. . . . .         | 58 |
| III.13 Traitement des trois premières images . . . . .   | 59 |
| III.14 Organigramme d'optimisation du temps de calcul pour le traitement des<br>images . . . . . | 60 |
| III.15 Principe de l'algorithme de localisation absolue par la caméra . . . . .                  | 61 |
| III.16 Environnement expérimental. . . . .   | 61 |
| III.17 Séquence de 17 images au cours du déplacement du robot . . . . .                          | 62 |
| III.18 Différentes positions images du robot au cours de son déplacement. . . . .                | 63 |

---

|   |    |
|---|----|
| III.19 Comparaison des coordonnées du point d'arrivée par les 2 méthodes de localisation . . . . .  | 63 |
| III.20 Localisation du Robot (courte trajectoire). . . . .  | 64 |
| III.21 Vitesses droite et gauche pour la trajectoire de la figure III.20. . . . .   | 64 |
| III.22 Localisation du Robot (longue trajectoire). . . . .  | 64 |
| III.23 Vitesse des roues droite et gauche. . . . .  | 64 |
| III.24 Localisation du Robot (trajectoire rampe). . . . .   | 64 |
| III.25 Vitesses de commande pour la trajectoire de la figure III.24. . . . .  | 64 |
| III.26 Relevé de 4 positions du robot avec 3 méthodes de mesures . . . . .  | 65 |
| III.27 Correction de la trajectoire du robot par le système de localisation utilisant la caméra . . . . .   | 66 |
| III.28 Positions du robot estimées à partir des deux systèmes (odométrique et caméra) avec un recalage du point d'arrivée . . . . .   | 67 |
| IV.1 Principe général de la méthode ZDV . . . . .   | 70 |
| IV.2 Evitement d'obstacles réflexes [23] . . . . .  | 71 |
| IV.3 Zone de sécurité sous forme d'ellipse non déformée [23]. . . . .   | 72 |
| IV.4 Zone de sécurité sous forme d'ellipse déformée. . . . .  | 73 |
| IV.5 Illustration du principe de la ZDV appliquée pour le ROMNI. . . . .  | 76 |
| IV.6 Zone de sécurité circulaire non déformée du ROMNI. . . . .   | 77 |
| IV.7 L'architecture de contrôle générique du fauteuil roulant . . . . .   | 80 |
| IV.8 Diagramme par Hystérésis appliqué au robot . . . . .   | 81 |
| IV.9 Trajectoire avec un changement de la direction du robot et des obstacles dynamiques . . . . .  | 82 |
| IV.10 Cette figure montre les quatre mini-zone de sécurité : $I_g$ , $I_d$ , $I_{gc}$ et $I_{dc}$ . Les règles sont conçues pour des scénarios multiples afin d'aider l'utilisateur . . . . . | 83 |
| IV.11 Les fonctions d'appartenances de l'intrusion de devant (droite et gauche). Ceci implique les déformations avec un angle $\pm 60^\circ$ par rapport à l'orientation . . . . .            | 84 |
| IV.12 Les fonctions d'appartenances de l'intrusion de côté (droite et gauche) . . . . .   | 85 |
| IV.13 Variables de sortie du commutateur flou . . . . .   | 85 |

---

|       |   |     |
|-------|---|-----|
| IV.14 | Environnement de travail du fauteuil . . . . .  | 87  |
| IV.15 | Trajectoire complexe parcourue par le fauteuil avec $\theta = 0$ . . . . .                    | 87  |
| IV.16 | Vitesses linéaires des roues droite et gauche . . . . .                                       | 88  |
| IV.17 | Trajectoire parcourue par le fauteuil dans une situation de minimum local<br>(coin) . . . . . | 89  |
| IV.18 | Interface utilisateur 2D. L'obstacle choisi est un couloir. . . . .                           | 89  |
| IV.19 | Interface 2D au cours de la navigation du fauteuil. . . . .                                   | 89  |
| IV.20 | Suivi de trajectoire : une série de points objectifs avec un minimum local.                   | 90  |
| IV.21 | Suivi de trajectoire avec des obstacles de formes différentes. . . . .                        | 91  |
| IV.22 | Trajectoire parcourue par le ROMNI avec un angle initial $\theta = 0$ . . . . .               | 92  |
| IV.23 | Vitesses linéaires du ROMNI pour $\theta = 0$ . . . . .                                       | 93  |
| IV.24 | Trajectoire parcourue par le ROMNI avec un angle initial $\theta = 180^\circ$ . . . .         | 94  |
| 1     | Intervalle de balayage du capteur laser Hukoyo . . . . .                                      | 105 |
| 2     | Requêtes émises et reçues . . . . .   | 106 |
| 3     | Requête émise par l'hôte . . . . .  | 106 |
| 4     | Requête de réponse du capteur . . . . .   | 106 |
| 5     | Procédure de conversion des données en distance . . . . .                                     | 107 |
| 6     | Les distances acquises du Hukoyo en fonction de l'angle . . . . .                             | 107 |

# Liste des tableaux

|       |   |    |
|-------|---|----|
| II.1  | Tableau d'inférence linguistique pour rejoindre l'objectif : cas du Khepera II.   | 34 |
| II.2  | Tableau des inférences linguistiques pour rejoindre l'objectif : cas du fauteuil. | 40 |
| II.3  | Tableau des inférences linguistiques pour rejoindre l'objectif : cas du ROMNI.    | 45 |
| III.1 | Evaluation des performances des méthodes de localisation . . . . .                | 65 |
| IV.1  | Tableau des inférences linguistiques du commutateur flou. . . . .                 | 86 |
| IV.2  | Paramètres de contrôle du fauteuil roulant . . . . .                              | 86 |
| IV.3  | Paramètres de contrôle du ROMNI . . . . .   | 92 |

# Introduction générale

Le terme robot a été largement étudié et utilisé dans la littérature mais le sens visé par cette expression tourne en général autour de celui-ci :

*Un robot est une machine équipée de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a.*

La manière dont un robot gère l'enchaînement de ce cycle Perception/Décision/Action est définie par son architecture de contrôle, qui peut éventuellement faire appel à un modèle interne de l'environnement lui permettant alors de planifier ses actions à long terme.

## Mise en contexte

Avec les progrès constants dans ce domaine, les robots tendent à accomplir des tâches de plus en plus complexes avec une intervention moindre de l'homme pour les guider. Ils deviennent ainsi plus autonomes, et ils interagissent de mieux en mieux avec leur environnement pour accomplir la mission qui leur a été assignée, le robot devient alors une "machine intelligente". Au cours des dernières décennies, la robotique a connu un développement très marqué. Les bras manipulateurs se sont largement développés dans les milieux manufacturiers et poursuivent leurs développements dans d'autres milieux tels que la chirurgie. De la même façon, la robotique mobile s'est aussi largement développée. Cela a commencé par une automatisation des chariots guidés, en passant par les plates formes mobiles pour arriver aux robots humanoïdes. A la différence des robots manipulateurs industriels qui évoluent dans des environnements protégés et très structurés tels que des chaînes de montage, les robots mobiles sont appelés à intervenir dans des environnements en perpétuelle évolution, en particulier en raison de la présence humaine. Le robot doit

donc disposer d'un niveau d'autonomie suffisant pour remplir sa mission. Les chercheurs progressent dans cette direction et dotent les robots mobiles d'un niveau d'autonomie de plus en plus grand.

La problématique de la navigation et du pilotage des robots, s'ouvre aujourd'hui vers de nombreux autres champs d'application, tels que : les véhicules, les drones aériens, les engins sous-marins, la chirurgie, les robots d'assistance, etc. Dans tous les cas, il s'agit de faire évoluer des systèmes de façon sûre dans des milieux imparfaitement connus en contrôlant les interactions entre le robot et son environnement. En premier lieu, lors de la planification, cela se fera par l'acquisition, la modélisation et la manipulation des connaissances sur l'environnement et sur l'objectif à réaliser. Ensuite, durant l'exécution, il s'agit d'exploiter les données perceptuelles pour adapter au mieux le comportement du système aux conditions de la mission qu'il doit réaliser.

## Problématique

Ce mémoire traite le thème de la recherche sur la navigation des robots mobiles à roues avec comme champ d'application la robotique d'assistance aux handicapés. Dans nos travaux, nous considérons deux types de robots mobiles. Le premier type concerne les robots uni-cycle et le deuxième type est un robot omnidirectionnel. Nous nous plaçons plus précisément dans le cadre de la navigation des robots mobiles (commande, localisation et pilotage).

Des travaux de recherche ont entamé la partie commande en utilisant différentes méthodes y compris la commande neuronale, optimale, floue. Ce travail de mémoire consiste à proposer un algorithme de commande soit pour suivre une trajectoire soit pour atteindre un point objectif qui est généralement un point fixe. La méthode que nous avons retenue est la commande floue qui permet de donner plus d'autonomie à un robot mobile pour arriver à destination. En revanche, pour une forte variation entre l'angle de départ du robot et l'angle du point d'arrivée, la précision de la commande se dégrade fortement. Ainsi, une étude bibliographique sur les techniques de localisation (relative et absolue) était nécessaire. Des travaux de recherche ont entamé cette phase avec différentes techniques qui vont de la simple fusion de données capteurs avec Filtre de Kalman à l'adaptation floue. C'est dans ce contexte de la fusion de données capteurs que se situent nos travaux de recherche en s'intéressant à la localisation absolue avec Webcam. On a recours à cette technique de localisation parce que la plupart des robots mobiles sont équipés d'un système de localisation relative utilisant des encodeurs qui est indispensable pour déterminer la position du robot mobile en temps réel. Mais on doit avoir recours à un deuxième

---

système qui permet de combler les lacunes du premier. Parmi les lacunes en question on peut rappeler, la connaissance du point initial, un recalage de temps en temps... On a montré que l'utilisation d'une caméra de type webcam qui est un dispositif bon marché comparé avec d'autres capteurs donnés peut déterminer la position absolue du robot. Le principal inconvénient apporté par l'utilisation de cette technologie est l'importance du temps de calcul. On a pu diminuer ce temps de calcul en n'effectuant le traitement que sur la partie de l'image qui contient l'image du robot. En définitive, c'est les deux systèmes de localisation relative et absolue qu'il est judicieux de mettre en œuvre, car ils sont complémentaires.

La dernière problématique se rapportant à la navigation des robots mobiles est le pilotage. En effet plusieurs travaux de recherche ont entamé, dans ce cadre, le pilotage des robots mobiles en proposant différentes techniques d'évitement d'obstacles réactives. Parmi ces dernières, se situe essentiellement les travaux de Zapata [22] qui a proposé une méthode qui montre le comportement réactif des manipulateurs mobiles en se basant sur la technique de zone à déformation virtuelle. En effet, les actions réflexes des robots mobiles peuvent être définies comme l'habilité à réagir quand un événement soudain se produit, et ceci en se déplaçant dans un environnement inconnu et dynamique. L'idée principale est de définir l'interaction robot/environnement comme une Zone de Déformation Virtuelle (ZDV) qui entoure le véhicule. C'est dans cette partie que nous allons nous proposer de la résoudre pour les deux plates formes mobile unicycle et omnidirectionnelle.

L'architecture de contrôle adoptée pour combiner la partie commande et la partie évitement d'obstacle est une architecture générique qui assure plus d'autonomie au robot. Cette architecture nous a permis d'aboutir à des résultats de navigation satisfaisants tout en utilisant une interface 2D permettant un suivi en temps réel du véhicule ainsi qu'une collaboration utilisateur-vehicule.

## Organisation du mémoire

Ce mémoire comporte quatre chapitres structurés de la manière suivante. Dans le premier chapitre nous présentons le cadre de l'étude et nous rappelons les modèles mathématiques associés aux robots étudiés durant les travaux de thèse ainsi que les méthodes de commandes, d'évitements d'obstacles et de contrôle utilisées.

La partie commande, basée sur deux méthodes, est présentée dans le deuxième et le troisième chapitre. La première méthode, sujet du deuxième chapitre, est un contrôleur flou qui permet de commander un robot mobile à roues d'une position initiale vers une position désirée, tout en respectant les contraintes cinématiques. Cette commande a été

développée et appliquée (simulation et expérimentation) pour les différents robots cités ci-dessus.

Cette approche floue s'avère limitée dans certains cas, notamment qu'elle se base uniquement sur des sources d'informations proprioceptives. Nous avons cherché à étendre les résultats obtenus en développant une localisation absolue par des données caméra assurant une correction hors ligne des données relatives. Cette deuxième commande a été développée et implémentée sur le robot Khepera II et fait l'objet du troisième chapitre.

Dans le chapitre quatre, nous détaillons le problème de la navigation d'un robot mobile dans un environnement parsemé d'obstacles. Un contrôleur basé sur la théorie de zones de déformations virtuelles (ZDV) a été développé. L'architecture de contrôle adoptée est de type générique associée à une commutation floue. Nous avons ainsi conçu les trois étapes nécessaires pour assurer l'autonomie d'un robot mobile : perception/décision/action. Nous avons exploité ces résultats dans le contexte applicatif de la navigation de deux types de plate-forme mobiles. Une interface graphique sous Matlab a été développée pour faciliter les opérations lors de la simulation et de la validation expérimentale.

Finalement nous concluons sur les travaux réalisés et présentons nos perspectives sur les futurs travaux.

# Chapitre I

## Cadre de l'étude

**Résumé** – Dans ce chapitre nous présentons les modèles mathématiques des deux types de robot mobiles utilisés dans ces travaux de thèse. Deux robots mobiles de types unicycle qui sont : le robot Khepera II et un fauteuil roulant pour handicapés. Le second véhicule est un robot omnidirectionnel ROMNI. Ensuite, nous détaillons les différents types de localisation notamment la localisation relative qui se base sur les sources d'informations proprioceptives et la localisation absolue qui utilise les sources d'informations extéroceptives. Puis, nous présentons quelques méthodes d'évitement d'obstacles et de navigation que le robot pourra utiliser pour se déplacer tout en lui assurant plus d'autonomie. Enfin, nous présentons les principales architectures de contrôle utilisées en robotique. Cette étude bibliographique nous a permis à la fin de déduire quand aux méthodes de commande et de pilotage les mieux adaptées à nos robots mobiles.

**Mots Clés** – Modélisation, localisation, navigation, pilotage, sources d'informations.

### I.1 Modélisation des plateformes robotiques

Ces travaux de thèse entrent dans le cadre de collaboration entre deux unités de recherche. Ceci a influé positivement en terme de richesse des plateformes robotiques étudiées. Dans ce cadre nous avons étudié deux robots de type unicycle : le robot Khépéra et un fauteuil roulant pour handicapés de l'unité de recherche ICOS. Le laboratoire français Prisme nous a fourni une plateforme omnidirectionnelle nommée ROMNI.

### I.1.1 Plateforme unicycle

Dans la littérature il y a plusieurs types de robots mobiles à roues [1]. Les plus étudiés sont de type voiture, commandés par l'angle de braquage des roues. Dans cette thèse nous avons étudié deux plates formes de type unicycle : un robot Khepera II et un fauteuil roulant pour handicapés.

#### I.1.1.1 Premier robot unicycle : Khepera II

Le robot Khepera II est un mini-robot qui a été conçu et développé au Laboratoire de Micro Informatique (LAMI) de l'Ecole Polytechnique Fédérale de Lausanne. Ce robot est composé de divers modules qui s'imbriquent les uns sur les autres. Au départ, le Khepera II fut développé comme outil de recherche pour vérifier la validité de divers algorithmes dans un contexte réel, par exemple : algorithmes de contrôle de trajectoires, algorithmes d'évitement d'obstacles et algorithmes de prétraitement des informations fournies par des capteurs. Les robots Khepera II se composent tous de deux modules de base : le module comportant la partie motrice et la partie d'alimentation d'une part, le module de contrôle possédant le microcontrôleur et la mémoire d'autre part. Sur ces deux modules de base, peuvent s'ajouter d'autres modules appelés tourelles, tels que : une pince pour manipuler des objets, une caméra unidimensionnelle, une tourelle permettant le développement d'applications spécifiques ou une tourelle radio. La figure I.1 montre le module de base du robot Khepera II.



FIG. I.1 – Schéma du robot mobile Khepera II

Le Khepera II possède (dans sa version de base) peu de capteurs, mais quand même suffisamment pour faire des applications intéressantes. Le Khepera II dispose de huit capteurs de distance répartis tout autour du robot qui permettent de détecter des obstacles, de deux moteurs indépendants permettant au robot d'avancer.

Etant donné que nos expérimentations seront faites sur un robot mobile à deux roues motrices indépendantes dont le mouvement est obtenu en agissant sur la vitesse de chaque roue comme le montre le modèle cinématique I.1, un ensemble d'équations montre qu'il s'agit d'un système d'équations multivariées. Les paramètres  $V_d$ ,  $V_g$  et  $\omega$  sont respectivement la vitesse linéaire de la roue droite, la vitesse linéaire de la roue gauche et la vitesse angulaire du robot mobile. Les paramètres  $x$ ,  $y$  et  $\theta$  sont respectivement l'abscisse, l'ordonnée et l'orientation du robot dans le repère défini selon la figure I.2.  $L$  est la distance entre les deux roues motrice du robot.

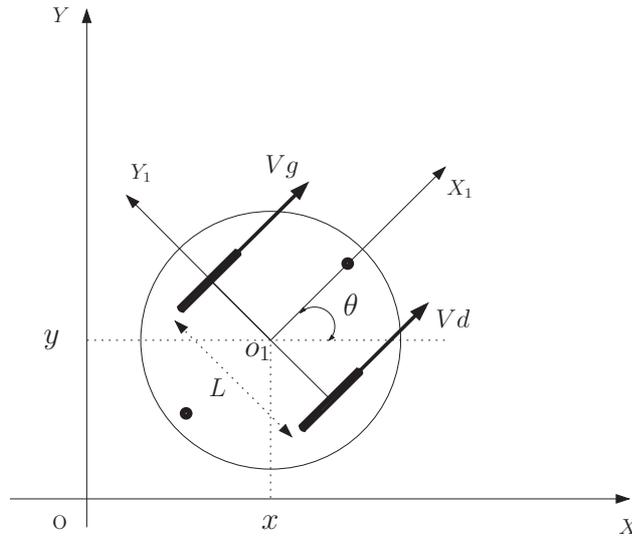


FIG. I.2 – Schéma cinématique du robot mobile à deux roues motrices indépendantes

$$\begin{cases} \frac{dx}{dt} = \frac{V_d + V_g}{2} \cos\theta \\ \frac{dy}{dt} = \frac{V_d + V_g}{2} \sin\theta \\ \frac{d\theta}{dt} = \omega = \frac{V_d - V_g}{L} \end{cases} \quad (\text{I.1})$$

Pour la simulation on a retenu le modèle échantillonné défini par le système d'équations (I.2) où  $T$  est la période d'échantillonnage.

$$\begin{cases} x_k = x_{k-1} + T \frac{V_{d_k} + V_{g_k}}{2} \cos\theta_k \\ y_k = y_{k-1} + T \frac{V_{d_k} + V_{g_k}}{2} \sin\theta_k \\ \theta_k = \theta_{k-1} + T \frac{V_{d_k} - V_{g_k}}{L} \end{cases} \quad (\text{I.2})$$

Pour les expériences, le robot mobile Khepera II est muni d'un système odométrique qui permet de déterminer la position et l'orientation du robot dans un repère de référence. Pour ce système de mesure, la position du robot à un instant  $k$  est définie en fonction de sa position à l'instant  $k - 1$ ,  $(x_{k-1}, y_{k-1}, \theta_{k-1})$ , et du déplacement des roues,  $(lg_k, ld_k)$  entre  $k$  et  $k - 1$ . On peut noter ici l'importance de la connaissance du point initial. Ce modèle d'évolution du robot s'écrit :

$$\begin{cases} x_k &= x_{k-1} + l_k \cos(\theta_k + \frac{\Delta\theta_k}{2}) \\ y_k &= y_{k-1} + l_k \sin(\theta_k + \frac{\Delta\theta_k}{2}) \\ \theta_k &= \theta_{k-1} + \Delta\theta_k \end{cases} \quad (\text{I.3})$$

avec :

$$\begin{cases} l_k &= \frac{ld_k + lg_k}{2} \\ \Delta\theta_k &= \frac{ld_k - lg_k}{L} \end{cases} \quad (\text{I.4})$$

### I.1.1.2 Deuxième robot unicycle : fauteuil roulant

La deuxième plateforme unicycle étudiée est un fauteuil roulant pour handicapés monté dans l'unité de recherche ICOS (I.3). Ce robot est doté de deux roues folles et deux roues motrices indépendantes ce qui fournit une base mobile avec deux degrés de mobilité [2].

Les principales caractéristiques de cette plateforme sont les suivantes :

- La distance entre les deux roues motrices est  $L = 750$  mm
- Le rayon de la roue est  $r = 160$  mm
- La masse est 70 Kg

Cette plateforme est aussi équipée d'un système de localisation basé sur deux encodeurs de type «Easy Roller ENC300CPR» placés sur les roues motrices du fauteuil et permettant de déterminer la position du robot par rapport à son repère global en utilisant le même modèle cinématique que celui décrit dans le paragraphe précédent. Ces encodeurs procurent en sortie 300 impulsions/tour, ces signaux de sortie permettent de calculer le déplacement linéaire et angulaire de chaque roue.

Durant les premières expériences, nous avons utilisé un ordinateur portable placé sur le système robotique interfacé avec le joystick et l'unité de contrôle des moteurs en utilisant un système bloc nommé «CB-405» comme le montre la figure I.3.

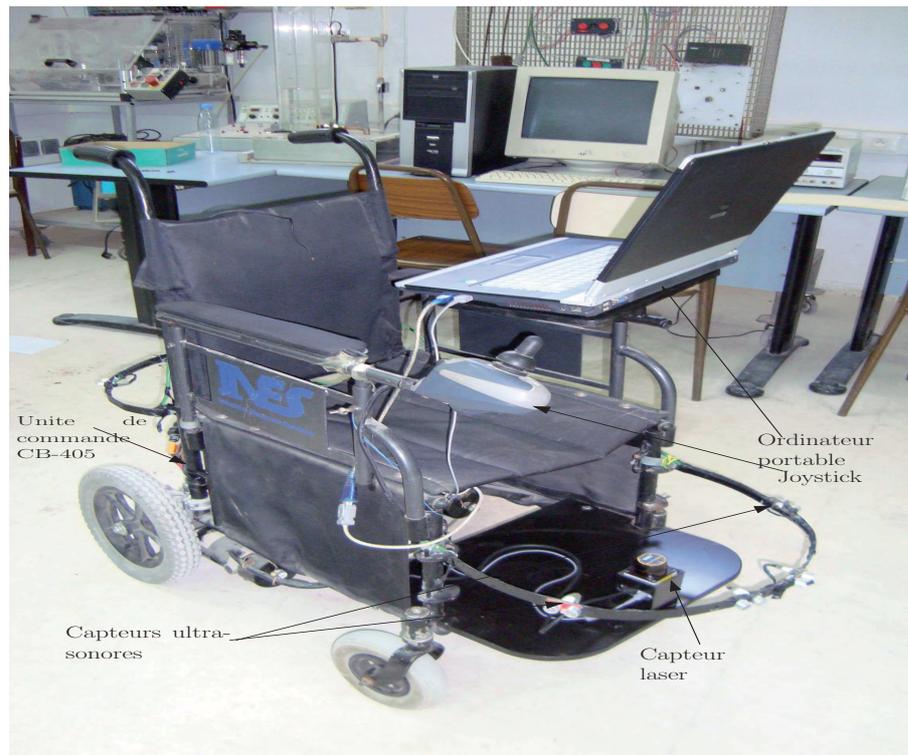


FIG. I.3 – Configuration courante du fauteuil roulant. La partie logicielle sur l'ordinateur portable utilise les mesures du module CB405 et le capteur laser pour contrôler les mouvements du fauteuil

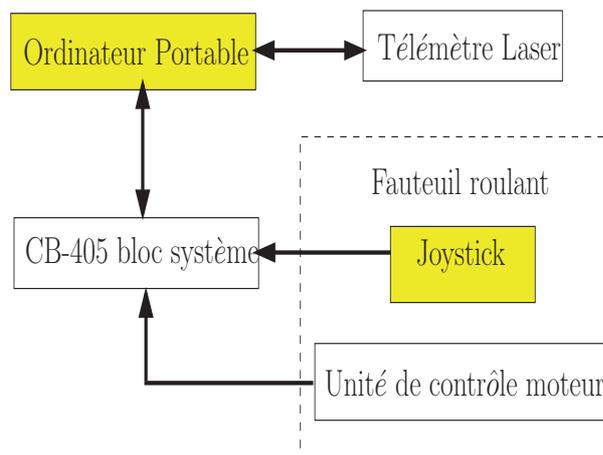


FIG. I.4 – Diagramme décrivant les méthodes d'interaction de l'utilisateur : à travers le joystick ou l'ordinateur portable. Toutes les commandes du joystick sont traitées avec le module «CB-405» avant d'être envoyées aux moteurs.

Cette architecture de communication nous permet d'intercepter les signaux du joystick et les détruire (si nécessaire), ceci avant de les envoyer à l'unité de commande des moteurs du fauteuil (I.4). Nous avons développé aussi un système de localisation se basant sur les encodeurs. Ce système est fonctionnel dans des environnements internes et externes. Dans un deuxième temps l'ordinateur portable a été remplacé par un PC embarqué de type «PC-104» permettant d'assurer plus d'autonomie et de coopération à l'application robotique envisagée pour assister des personnes handicapées.

### I.1.2 Plateforme omnidirectionnelle

Le ROMNI est un robot omnidirectionnel réalisé (fabrication et intégration mécatronique complète) dans le Laboratoire «Pluridisciplinaire de Recherche en Ingénierie des Systèmes, Mécanique, Energétique (PRISME)». Cette plateforme est constituée de trois essieux et d'un plateau circulaire en alliage d'aluminium de diamètre de 600 mm et d'épaisseur maximale de 15 mm. Sur ce plateau, sont fixés trois moteurs d'articulation, des cartes PC et des cartes de pilotage des moteurs, des batteries pour l'autonomie énergétique [3]. La figure I.5 montre une vue de ce robot.

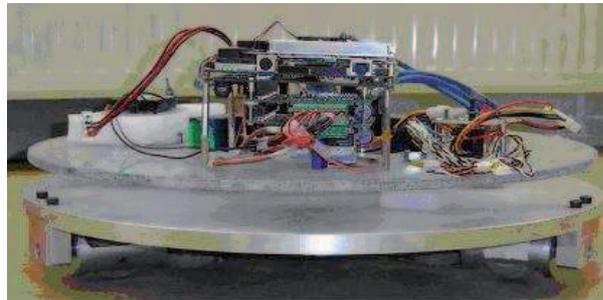


FIG. I.5 – Représentation du robot omnidirectionnel ROMNI (Photo PRISME)

La particularité du robot ROMNI réside essentiellement au niveau de ses essieux. Sur la figure I.6 est présenté un essieu formé par deux sphères d'axes de rotation perpendiculaires l'un par rapport à l'autre. Chacune des deux sphères, est tour à tour en contact avec le sol et fait office de roue motrice. La mobilité interne que constitue la rotation libre des sphères, permet un déplacement en translation de l'essieu (et de la structure qui le porte) dans sa direction longitudinale, sans glissement des roues sur le sol et sans perturber la fonction motrice qui agit dans une direction perpendiculaire à cet essieu.

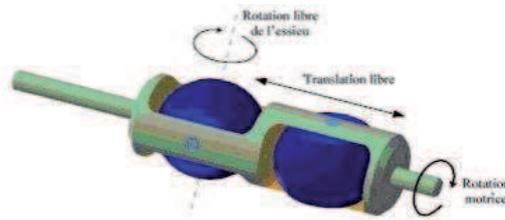


FIG. I.6 – Représentation d'un essieu (Thèse Mourioux 2006[3])

### I.1.2.1 Modélisation cinématique

Dans le but de déterminer le modèle cinématique de la plateforme omnidirectionnelle, nous allons commencer par calculer la vitesse du point  $O_i$  de l'essieu par rapport au repère absolu ( $R_O$ ) [4]. La figure I.7 décrit une représentation cinématique d'un essieu dans le repère absolu.

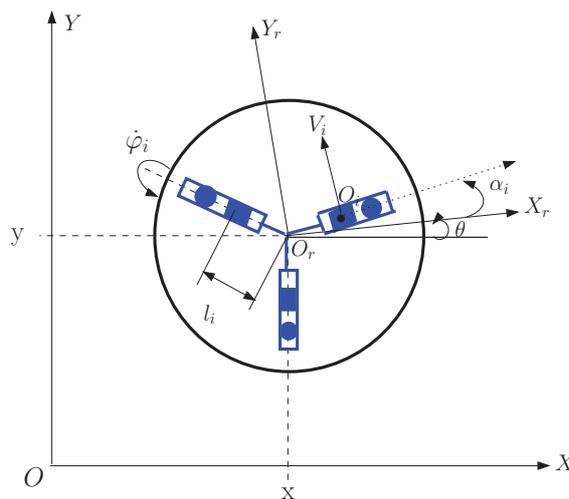


FIG. I.7 – Schéma cinématique du robot omnidirectionnel ROMNI (Thèse Mourioux [3]).

Les indices  $i$  ( $i$  variant de 1 à 3) caractérisent respectivement les essieux de 1 à 3 :

$\vec{u}_i$  : vecteur de direction longitudinale de l'essieu,

$\vec{v}_i$  : vecteur directement perpendiculaire à  $\vec{u}_i$ ,

$O_i$  : centre de la sphère en contact avec le sol,

$r = 30\text{mm}$  : rayon de la sphère,

$\theta = (\overrightarrow{O_r X}, \overrightarrow{O_r X_r})$  : angle que fait le vecteur  $\overrightarrow{O_r X}$  avec le vecteur  $\overrightarrow{O_r X_r}$ , il caractérise l'orientation absolue du robot.

Une étude plus détaillée du mécanisme du Romni permet d'aboutir à la représentation cinématique de la figure I.8 :

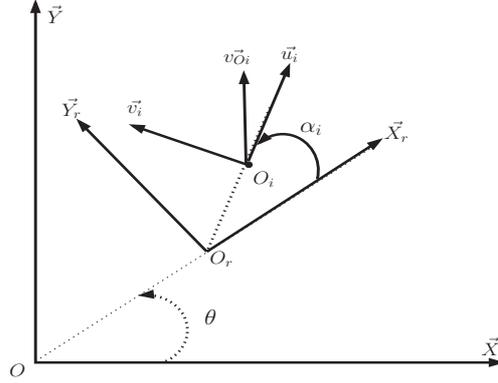


FIG. I.8 – Repère absolu, repère robot et repère essieu (vue 2D de dessus).

avec :

- $(O, \vec{X}, \vec{Y})$  : repère absolu du système
- $(O_r, \vec{X}_r, \vec{Y}_r)$  : repère de centre  $O_r$  (lié à la plateforme)
- $\dot{\varphi}_i$  : vitesse de rotation propre de l'essieu,
- $\vec{V}_{O_i}$  : vecteur vitesse du point (égal à  $\vec{V}_{(O_i \in \text{Essieu}/R_O)}$  )
- $\alpha_i$  : angle  $(\overrightarrow{O_r X_r}, \overrightarrow{O_r u_i})$  avec

$$\begin{cases} \alpha_1 = 0 \\ \alpha_2 = \frac{2\pi}{3} \\ \alpha_3 = \frac{4\pi}{3} \end{cases}$$

Afin de déterminer le modèle cinématique de notre plateforme, nous calculons la vitesse du point  $O_i$  de l'essieu par rapport au repère absolu ( $R_O$ )(Thèse Mourieux [3]) :

$$\vec{V}_{(O_i \in \text{Essieu}/R_O)} = \vec{V}_{(O_r \in \text{Essieu}/R_O)} + \vec{\Omega}_{\text{Essieu}/R_O} \wedge \overrightarrow{O_r O_i} \quad (\text{I.5})$$

Avec une roue classique et l'hypothèse de roulement sans glissement, on obtient que  $\vec{V}_{(O_i \in \text{Essieu}/R_O)}$  est colinéaire à  $\vec{V}_i$  ou encore  $\vec{V}_{(O_i \in \text{Essieu}/R_O)} = -r\dot{\varphi}_i \vec{V}_i$

Dans notre cas (figure I.8), la vitesse dans la direction  $\vec{u}_i$  est libre et non commandée par le mouvement de l'essieu ( $\vec{V}_{(O_i \in \text{Essieu}/R_O)} \cdot \vec{u}_i$  est quelconque). Ceci se traduit donc pour chacun des trois essieux, par la contrainte cinématique suivante (Thèse Mourieux [3]) :

$$\vec{V}_{(O_i \in \text{Essieu}/R_O)} \cdot \vec{V}_i = -r\dot{\varphi}_i \quad \forall i \in 1, 2, 3 \quad (\text{I.6})$$

Le mouvement de la plateforme est un mouvement plan sur plan ; une roue (sphère) de chaque essieu est à tout moment en contact avec le sol, il y a toujours trois sphères en

contact, la plateforme conserve ainsi son assiette. Considérons  $\dot{P} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}$  les paramètres cinématiques de ce mouvement au centre  $O_r$  de la plateforme. Si pour chaque essieu, on note  $l_i$ , la distance  $O_r O_i$  (distance variable selon la sphère en contact avec le sol), on obtient (Thèse Mourioux [3]) :

$$\begin{aligned} \vec{V}_{(O_i \in \text{Essieu}/R_O)} &= \vec{V}_{(O_r \in \text{Essieu}/R_O)} + \vec{\Omega}_{\text{Essieu}/R_O} \wedge \overrightarrow{O_r O_i} \\ &= V_{xr} \vec{X}_r + V_{yr} \vec{Y}_r + \Omega l_i \vec{V}_i \end{aligned} \quad (\text{I.7})$$

$V_{xr}$  et  $V_{yr}$  sont les composantes de  $\vec{V}_{(O_r \in \text{Essieu}/R_O)}$  suivant  $\vec{X}_r$  et  $\vec{Y}_r$ .

Cette relation peut s'écrire pour chacun des 3 centres  $O_1$ ,  $O_2$  et  $O_3$  des sphères en contact où les angles  $(\overrightarrow{O_r X_r}, \overrightarrow{O_r u_i})$  ont respectivement les valeurs suivantes :  $\alpha_1 = 0$ ,  $\alpha_2 = 2\Pi/3$  et  $\alpha_3 = 4\Pi/3$ .

La combinaison des équations décrites ci-dessous donne (Thèse Mourioux [3]) :

$$\begin{cases} -\dot{x} \sin(\theta + \alpha_1) + \dot{y} \cos(\theta + \alpha_1) + \dot{\theta} l_1 &= -r\dot{\varphi}_1 \\ -\dot{x} \sin(\theta + \alpha_2) + \dot{y} \cos(\theta + \alpha_2) + \dot{\theta} l_2 &= -r\dot{\varphi}_2 \\ -\dot{x} \sin(\theta + \alpha_3) + \dot{y} \cos(\theta + \alpha_3) + \dot{\theta} l_3 &= -r\dot{\varphi}_3 \end{cases} \quad (\text{I.8})$$

Le saut de roues entraîne une discontinuité théorique des composantes  $\dot{P} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}$  et on peut constater que la distance  $l_i$  n'a d'effet que pour un mouvement comprenant une rotation, une approximation peut être envisagée en considérant une valeur médiane :

$$l_i = \frac{l_{imax} + l_{imin}}{2}$$

avec :  $l_{imax} = 200\text{mm}$  ;  $l_{imin} = 130\text{mm}$ .

### I.1.2.2 Modèle cinématique inverse (MCI)

Le système d'équations (I.8) peut s'écrire sous forme matricielle (I.9), ce qui constitue le Modèle Cinématique Inverse (MCI) du robot (Thèse Mourioux [3]) :

$$\dot{\phi} = A\dot{P} \quad (\text{I.9})$$

Avec :

$$\dot{\phi} = \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix}, \dot{P} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}, A = -\frac{1}{r} \begin{bmatrix} -\sin(\theta + \alpha_1) & \cos(\theta + \alpha_1) & l_1 \\ -\sin(\theta + \alpha_2) & \cos(\theta + \alpha_2) & l_2 \\ -\sin(\theta + \alpha_3) & \cos(\theta + \alpha_3) & l_3 \end{bmatrix}$$

L'existence de ce modèle et le fait que la matrice A soit de rang plein sans configuration singulière montre que, pour toute expression du vecteur d'entrée  $\dot{P}$ , on est capable de déterminer les variables articulaires : on prouve de ce fait l'omnidirectionnalité du robot.

### I.1.2.3 Modèle cinématique direct (MCD)

L'existence du Modèle Cinématique Inverse (MCI) (I.9) montre que pour toute expression du vecteur d'entrée  $\dot{P}$ , on est capable de déterminer les variables articulaires  $\dot{\phi}$ . Le calcul du déterminant de la matrice de transfert A, selon les valeurs des rayons  $l_i$ , conduit à quatre expressions différentes, toutes de valeurs non nulles. Cette matrice est donc inversible ; on obtient alors le Modèle Cinématique Direct (MCD)(Thèse Mouriaux [3]) :

$$\dot{P} = A^{-1}\dot{\phi} \quad (\text{I.10})$$

avec :

$$A^{-1} = H \begin{bmatrix} l_2 C_3 - l_3 C_2 & l_3 C_1 - l_1 C_3 & l_1 C_2 - l_2 C_1 \\ l_2 S_3 - l_3 S_2 & l_3 S_1 - l_1 S_3 & l_1 S_2 - l_2 S_1 \\ \sin(\alpha_2 - \alpha_3) & \sin(\alpha_1 - \alpha_3) & \sin(\alpha_1 - \alpha_2) \end{bmatrix}, C_i = \cos(\theta + \alpha_i), S_i = \sin(\theta + \alpha_i)$$

$$\text{et } H = \frac{-r}{l_1 \sin(\alpha_2 - \alpha_3) - l_2 \sin(\alpha_1 - \alpha_3) + l_3 \sin(\alpha_1 - \alpha_2)}$$

Dans cette partie, nous avons présenté les modèles cinématiques directs (MCD) de trois mobiles à roues : Khepera II, fauteuil et ROMNI. Ces modèles vont nous permettre d'estimer la position et l'orientation du robot durant sa navigation. Pour cela, différentes techniques de localisation pourront être utilisées. Dans la partie suivante nous allons présenter quelques techniques de localisation utilisées en robotique mobile.

## I.2 Localisation des robots mobiles

Les robots mobiles utilisent diverses techniques pour se localiser. Ces techniques présentent des avantages et des inconvénients : elles peuvent fournir soit la position relative,

soit la position absolue du mobile. Certaines nécessitent une instrumentation plus ou moins importante de l'environnement. La précision et le champ d'application de ces techniques sont variables. Souvent, la localisation est exécutée en combinant plusieurs de ces techniques.

### I.2.1 Les techniques de localisation

Parmi les techniques utilisées en robotique mobile, nous citons :

- **Filoguidage** : Un fil enterré dans lequel circule un courant électrique qui matérialise la route à suivre. Des capteurs inductifs embarqués sur le véhicule permettent de détecter le fil. Il ne reste plus alors au véhicule qu'à le suivre en utilisant une technique d'asservissement classique.
- **Guidage sur plots magnétiques** : Le principe est similaire à celui du filoguidage. Des plots magnétiques enterrés et espacés de quelques mètres remplacent le fil électrique. Le véhicule utilise ces plots pour se relocaliser [5]. La navigation entre deux plots se fait en se basant sur la localisation proprioceptive (typiquement l'odométrie). Cette technique a moins d'impact sur l'environnement que le filoguidage. La navigation est moins précise mais elle peut se faire à des vitesses plus élevées.
- **GPS** (Global Positioning System) : C'est un système de localisation absolue qui se base sur la mesure du temps de vol des signaux en provenance de différents satellites). La précision du GPS standard est de l'ordre de plusieurs mètres. Cette valeur est insuffisante pour la navigation. Par contre, le GPS différentiel (utilisation d'une station au sol qui sert de référence locale) et surtout le Real-Time Kinematic (RTK) GPS (analyse du temps de vol mais aussi de la phase des signaux satellites) permettent d'obtenir des précisions de l'ordre du centimètre qui sont suffisantes pour la navigation. Le GPS semble la solution de localisation idéale pour les robots d'extérieurs. Ce système a malheureusement des limites, notamment en milieu urbain ou la présence de bâtiments élevés peut empêcher de détecter suffisamment de satellites pour une bonne localisation. La figure I.9 montre un exemple de GPS pouvant être directement relié à un ordinateur via la liaison USB. Ce capteur est implémenté sur le fauteuil roulant pour handicapés.



FIG. I.9 – GPS de type Navilock-NL-464US – GPS – USB – 2 – 0 – Stick (Photo de la société Navilock).

Pour se localiser et détecter les obstacles, le robot doit au moins être équipé par plusieurs types de capteurs.

## I.2.2 Différents types de capteurs en robotique mobile

La localisation repose sur deux types généraux d'informations différentes : les informations proprioceptives qui mènent à une localisation relative et les informations extéroceptives assurant une localisation absolue.

### I.2.2.1 Les informations proprioceptives

Ce sont des informations internes au robot qui le renseignent, dans le cas de la navigation, sur son déplacement dans l'espace. Ces informations peuvent provenir de la mesure de la rotation de ses roues ou de la mesure de l'accélération grâce à une centrale inertielle. Un processus d'intégration permet alors, en accumulant ces informations au cours du temps, d'estimer la position relative de deux points par lesquels le robot est passé. Cependant, la qualité de cette information se dégrade continuellement au cours du temps, à cause de l'intégration temporelle des mesures effectuées par les capteurs internes. En effet, chaque capteur produit une mesure bruitée de la vitesse ou de l'accélération du robot (glissement des roues). Les encodeurs forment l'exemple de capteur proprioceptif le plus répandu. Il se fixe soit sur les roues motrices soit directement sur l'arbre du moteur associé à la roue. Il permet de délivrer des impulsions déterminant le déplacement latéral et angulaire de la roue comme le montre le système suivant I.11 :

$$\Delta S_{g,d} = \frac{\Pi D_n}{nC_e} N_{g,d} \quad (\text{I.11})$$

avec :

$D_n$  (mm) : diamètre nominal de la roue

$C_e$  (impulsion/mm) : résolution des encodeurs

$n$  : rapport de réduction du moteur

$N$  : incrémentation des impulsions

Les indices  $g$  et  $d$  designent respectivement la roue gauche et la roue droite.

### I.2.2.2 Les informations extéroceptives

Les informations extéroceptives ou encore les perceptions sont des informations caractéristiques d'une position que le robot peut acquérir dans son environnement. Ces informations peuvent être de nature très variée. Par exemple, un robot peut mesurer la distance des obstacles avec des capteurs infrarouges ou utiliser une caméra. En effet, les informations proprioceptives fournissent des informations sur le déplacement du robot, alors que les informations extéroceptives fournissent des informations directement sur la position du robot dans l'environnement.

- **Détecteurs de choc** : Ce sont de simples capteurs mécaniques. Ils sont en général intégrés dans les pare-chocs du robot et provoquent l'arrêt de celui-ci en cas de contact direct avec un obstacle. En terme de sécurité, ils ne peuvent remplir leur rôle que pour de faibles vitesses.
- **Télémètres** : Il existe différents types de télémètres (ultra-sonore, Infra-rouge, laser, radar) qui fonctionnent tous avec le même principe. On calcule la distance à un obstacle en mesurant le temps de vol d'une onde émise (ultra-sonore, lumineuse ou radio selon le type de télémètre).
  - *Télémètre ultra-sonore* : Ils sont adaptés pour la détection d'obstacles proches (de un à quelques mètres). L'information fournie est peu précise et sujette à un certain nombre de problèmes bien connus (absorption, réflexion spéculaire, réverbération, etc.). Ils peuvent suffire pour certaines applications et ont l'avantage d'être peu coûteux.
  - *Télémètre laser* : Ils sont basés sur un faisceau laser rotatif permettant de balayer un plan horizontal et de détecter les obstacles situés dans ce plan. La portée de ces capteurs peut atteindre 50 mètres avec une très bonne précision. Cependant, les capteurs lasers sont sensibles aux conditions atmosphériques (pluie, brouillard, etc). Ce sont à l'heure actuelle les capteurs les plus utilisés. Mais leur principal inconvénient est leur prix élevé.

L'exemple de la figure I.10 montre un capteur laser de type URG-04X qui peut balayer 270° avec une portée de 4 mètres.



FIG. I.10 – Télémètre Laser de type URG-04X (Photo de la société Hokuyo Automatic Co., LTD).

A ce niveau, nous avons présenté les modèles mathématiques de nos robots ainsi que quelques types de localisation utilisés en robotique mobile. Ces deux parties sont nécessaires à la commande d'un robot mobile à roues. La partie suivante sera donc consacrée à présenter les principales stratégies de navigation utilisées.

## I.3 Les stratégies de navigation

Les stratégies de navigation permettant à un robot mobile de se déplacer pour rejoindre un but sont extrêmement diverses [6]. Afin de situer le type de navigation adopté dans ce présent rapport dans son contexte général, nous nous basons sur une classification établie par Trullier *et al.* [7], [8] et [9], laquelle présente l'avantage de distinguer les stratégies de navigation sans modèles internes (stratégies locales) et les stratégies avec modèle interne (stratégies globales).

### I.3.1 Les stratégies de navigation locale

En se basant sur la classification de Trullier, la navigation locale comporte deux classes qui sont :

- **Approche d'un objet** : dans cette approche, le robot peut se diriger vers un objet visible depuis sa position courante et ceci en se basant en général sur la perception de l'objet. Cette stratégie utilise des actions réflexes, dans lesquelles chaque perception est directement associée à une action. C'est une stratégie locale, c'est-à-dire qu'elle est fonctionnelle uniquement dans la zone de l'environnement pour laquelle le but est visible.





Ainsi, elle mémorise les positions métriques relatives des différents lieux et ceci additionné à la possibilité de passer d'un lieu à un autre. Ces positions relatives permettent de calculer une trajectoire qui mène d'un lieu à un autre. Cette capacité est possible même si ce déplacement n'a pas été mémorisé sous forme d'un lien.

### I.3.3 Choix de la stratégie de navigation

A partir de la classification présentée, nous pouvons remarquer que les méthodes globales garantissent la sûreté et l'optimisation dans la navigation. Cependant les méthodes locales apportent une rapidité et une gestion de l'univers dynamique visible [13]. Les principales limitations des méthodes globales sont de deux types, la nécessité d'une modélisation préalable de l'environnement et la limitation sur le nombre de degrés de liberté du système considéré. En contre partie ces méthodes dites globales garantissent l'obtention d'une classe de solutions parmi lesquelles le choix d'une trajectoire particulière peut s'effectuer en fonction de contraintes sur la forme de la trajectoire ou d'un critère d'optimisation [14].

Cette étude de quelques stratégies de navigation nous a permis de distinguer la méthode à appliquer pour les trois robots mobiles déjà présentés. Nous étions donc orientés vers l'utilisation d'une navigation globale de type topologique qui se base sur les modèles internes de l'environnement. Cette stratégie utilisera la théorie des sous-ensembles flous pour permettre au robot de rejoindre un point objectif.

Le choix de la stratégie de navigation dans un environnement non encombré étant établi. Il nous reste donc à considérer le cas où le robot est contraint à différents obstacles dans son environnement. Une étude sur les méthodes d'évitement d'obstacles fera donc le sujet de la partie suivante de cette thèse.

## I.4 Méthodes d'évitement d'obstacles

L'évitement d'obstacles est un comportement quasiment essentiel pour tous les robots mobiles. Il lui permet de naviguer en présence d'obstacles tout en gérant les écarts entre le modèle interne du robot et le monde réel.

L'efficacité des méthodes que nous présenterons dans ce rapport a été prouvée dans différents travaux de recherche. Cependant, cette efficacité est conditionnée par l'existence d'une perception correcte de l'environnement (nature du capteur : laser, sonar, caméra). Pour remédier à ce problème, l'utilisation d'une représentation locale (de l'environnement

proche du robot et centrée sur le robot) a prouvé son impact sur le filtrage des signaux bruités de certain capteurs. Parmi ces méthodes il y a l'évitement d'obstacles par : la fenêtre dynamique, le flux optique et la zone de déformation virtuelle.

### I.4.1 Méthode de la fenêtre dynamique

La méthode de la fenêtre dynamique proposée initialement par Fox *et al.*[15], et étendue par Brock et Khatib[16], repose sur l'utilisation de la perception locale de l'environnement. Cette perception procure des informations permettant de générer un couple de vitesse de translation et de vitesse de rotation. Ce couple doit être calculé de manière à répondre à certaines contraintes dont celles d'éviter les obstacles. La méthode de la fenêtre dynamique permet de sélectionner le couple qui répond le mieux à ces contraintes. Une fenêtre qui fait apparaître les vitesses accessibles à partir des vitesses courantes du robot, est générée à chaque pas de temps. Au sein de cette fenêtre il faut choisir les couples de vitesses assurant une navigation sans collision.

En 2005 Rimon et Koditschek ont proposé un traitement théorique qui a prouvé la convergence des propriétés de la méthode de la fenêtre dynamique en utilisant le théorème de Lyapunov [17].

### I.4.2 Méthode du flux optique

La méthode d'évitement d'obstacle par flux optique s'inspire de la façon dont certains insectes (mouches et abeilles) utilisent le flux optique (par le biais de la vision et des capteurs gyroscopiques) pour déterminer leurs distances aux objets environnants [18], [19], [20]. Cette stratégie bio-inspirée consiste à rester à égale distance des obstacles latéraux par l'analyse du flux optique induit par le mouvement. Ainsi, l'insecte est capable de stabiliser sa trajectoire, maintenir son altitude ainsi que d'éviter des obstacles.

Cette méthode est transposable aux micro-drones. Le principe d'application consiste à calculer la différence entre le flot gauche et le flot droit du capteur de vision placé sur le micro-drone. A cette différence de flots, des fonctions d'attraction/répulsion sont définies. Elles permettent d'orienter le drone, d'estimer sa vitesse, et également d'assurer sa «survie» par évitement réflexe des obstacles. Les figures I.14 et I.15 montrent respectivement une détection de flux optique (présenté en trait rouge) et une détection d'un obstacle en mouvement par flux optique.



FIG. I.14 – Détection du flux optique (Projet FP6-IST-045248 [21])



FIG. I.15 – Détection d'un obstacle en mouvement (Projet FP6-IST-045248 [21])

### I.4.3 Méthode de la Zone de Déformation Virtuelle (ZDV)

La méthode d'évitement d'obstacles par Zone de Déformation Virtuelle (ZDV) a été proposée initialement par MM. Zapata [22] pour le cas des manipulateurs mobiles. Cette méthode repose sur la théorie d'évitement réflexe. Cette théorie permet de se rapprocher au mieux d'un comportement humain, car le contrôle du mouvement d'un robot mobile dans un environnement complexe nécessite une intelligence et une autonomie.

L'idée principale est de définir l'interaction robot/environnement comme une ZDV qui contourne le véhicule. La déformation de la zone de sécurité est due à l'intrusion d'une information provenant d'un capteur de proximité. Le calcul de cette déformation permet de piloter les réactions du robot qui dépendent des vitesses : linéaire et angulaire. En résumé, la zone de sécurité perturbée par un obstacle peut être reformée en agissant sur les vitesses du robot. Ensuite, cette méthode a été étendue dans les travaux [23] et [24] de Lapierre *et al.* Ces chercheurs ont combiné la partie pilotage basée sur ZDV avec la partie navigation (suivi de trajectoire). Cette méthode a été prouvée en utilisant le théorème de Lyapunov.

### I.4.4 Choix de la méthode d'évitement d'obstacles

L'étude bibliographique établie sur les différentes méthodes d'évitement d'obstacles, nous a permis d'opter pour le choix de la stratégie d'évitement par ZDV. Ce choix s'explique par le fait que cette méthode est réactive et qu'elle génère des actions réflexes rapides. Ceci permet d'assurer une autonomie importante au robot.

A ce niveau, nous sommes arrivés à présenter quatre parties essentielles pour l'autonomie d'un robot mobile : modélisation, localisation, commande et évitement d'obstacles. Ainsi, il nous reste à étudier les architectures de contrôle qui assurent le bon passage entre ces quatre parties tout en respectant les exigences du robot.

## I.5 Architectures de contrôle

Un robot est un système complexe qui doit satisfaire à des exigences variées et parfois contradictoires. Un exemple typique pour un robot mobile est l'action qui doit être faite pour atteindre un but et la prise en compte d'éléments imprévus, tels que les obstacles. Nous pouvons reprendre une traduction de la définition de Ronald Arkin [25] de l'art de concevoir de telles architectures :

*L'architecture robotisée est la discipline consacrée à la conception de robots fortement spécifiques et individuels et ceci à partir d'une collection de composantes logicielles communes.*

Ces architectures peuvent être classées en trois grandes catégories que nous détaillerons par la suite : les contrôleurs hiérarchiques, les contrôleurs réactifs et les contrôleurs hybrides.

### I.5.1 Contrôleurs hiérarchiques

Ces architectures fonctionnent en se basant sur une modélisation de l'environnement, planification des actions au sein de cette représentation, puis exécution (cf. figure I.16).

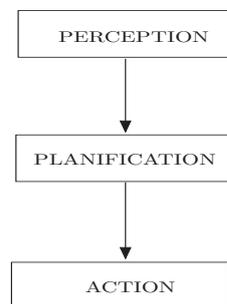


FIG. I.16 – Architecture hiérarchique (Arkin [25]).

Ces architectures n'assurent pas un contrôle important lors de l'exécution des actions. En effet, une fois l'action est choisie, elle est exécutée. Dans une telle situation, on suppose

que le modèle du monde est correct et qu'il n'y a pas un retour de la perception sur l'exécution de l'action. Les écarts entre les modèles internes et l'état réel de l'environnement ne peuvent être considérées que via le cycle de perception/modélisation/planification suivant. Ce retard peut engendrer de graves problèmes comme celui des collisions.

### I.5.2 Contrôleurs réactifs

Brooks [26] a proposé une solution aux inconvénients de l'architecture hiérarchique. Cette solution se présente sous la forme d'une architecture réactive. Dans cette architecture, un ensemble de comportements réactifs, qui fonctionnent simultanément, contrôle le robot sans utiliser de modèle interne. Ces architectures sont généralement basées sur plusieurs comportements : déplacement vers un but, évitement d'obstacles, déplacement aléatoire, suivi de mur, etc. Afin de guider le robot, le contrôleur doit choisir instantanément un comportement parmi les autres à activer. La solution proposée par Brooks utilise une hiérarchie des comportements. Cette hiérarchie se déclenche selon un ordre de priorité en fonction des perceptions du robot. (cf. figure I.17).

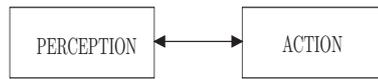


FIG. I.17 – Architecture réactive (Arkin [25])

### I.5.3 Contrôleurs hybrides

Actuellement, les contrôleurs utilisés adoptent une solution intermédiaire entre l'architecture hiérarchique et celle réactive. Cette solution s'appelle une architecture hybride [27]. Elle se compose de deux niveaux.

- Le premier niveau est chargé des tâches de navigation de haut niveau, telles que la localisation, la cartographie et la planification
- Le second niveau, étant réactif, est chargé d'exécuter les commandes et de gérer les éléments non modélisés de l'environnement (obstacles inconnus).
- Le niveau bas de l'architecture hybride se présente sous la forme de boucles sensori-motrices qui relient les actions aux perceptions. Cette liaison doit être réalisée avec une phase de décision très courte. Ceci permettra d'assurer la réactivité (cf. figure I.18).

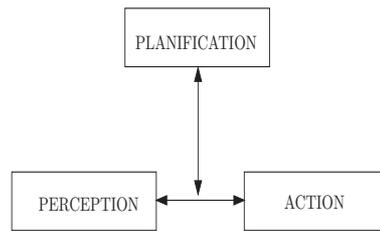


FIG. I.18 – Architecture hybride (Arkin [25])

#### I.5.4 Contrôleurs génériques

Le contrôleur générique a été conçu et développé en 2006 au sein du laboratoire PRISME par Mouriaux [28]. L'idée principale, à travers ce contrôleur, est d'assurer l'autonomie et la sécurité durant l'exécution d'une ou plusieurs tâches. Le système de contrôle générique d'un robot mobile doit être capable de : déterminer les intentions de l'utilisateur, générer une trajectoire optimale sans risque, et quand c'est nécessaire (présence d'obstacles) ajuster les signaux de commande afin d'atteindre l'objectif sans risque.

Cependant, le niveau d'autonomie dépend des capacités du robot (temps de réponse, angle de braquage maximal), et des tâches assignées. Une étape principale de ce développement est de concevoir une architecture de contrôle qui intègre à la fois la commande et l'évitement d'obstacle nécessaires pour l'autonomie du robot. Pour cela cette méthode a été basée sur le concept de niveaux.

En effet, cette architecture est composée principalement de deux parties qui sont : la partie de décision et la partie de perception. Ces deux parties du même niveau sont directement connectées : dans chaque niveau des modules sont définis, chaque module correspond à une fonctionnalité, qui permet une coupe fonctionnelle.

#### I.5.5 Choix de l'architecture de contrôle

A partir de l'étude établie ci-dessus sur les modèles mathématiques des robots considérés dans ces travaux de thèse, ainsi que les différents types de navigation et des architectures de contrôle, nous pouvons déduire quand à la méthode de navigation à appliquer à nos systèmes robotiques.

Les plates formes étudiés sont dotées d'un capteur interne qui est l'encodeur nous permettant d'assurer une localisation relative de ces systèmes. D'ailleurs, nous allons appliquer une méthode de navigation topologique permettant de rejoindre un point objectif avec une trajectoire optimisée.

---

L'architecture de contrôle retenue est hybride et elle est de type générique. La commutation entre la partie localisation/commande et la partie évitement d'obstacles est assurée à travers un contrôleur flou.

## I.6 Conclusion

Dans ce chapitre nous avons présenté différentes plateformes mobiles étudiées durant la préparation des travaux de thèse ainsi que leurs modèles mathématiques. Nous avons présenté deux robots de type unicycle : le Khepera II et le fauteuil roulant, et un autre de type omnidirectionnel : le ROMNI. Ensuite nous avons présenté différentes techniques de localisation (relatives et absolues) pouvant être adoptées sur ces plateformes ainsi que différentes sources d'informations à savoir celles proprioceptives et extéroceptives. Nous avons fini par présenter différentes stratégies de navigation, d'évitement d'obstacles et de contrôle existant dans la littérature. A partir de cette classification nous avons choisi d'opter pour les méthodes suivantes :

- une navigation topologique qui se base sur la théorie des ensembles flous.
- une technique d'évitement d'obstacles comportementale nommée Zone de déformation virtuelle (ZDV).
- une architecture de contrôle hybride permettant une commutation continue entre les parties commande et évitement d'obstacles.

Dans le chapitre qui suit nous allons commencer par aborder la partie commande des deux types de plates formes mobiles à roues.

# Chapitre II

## Contribution à la commande floue de deux types de robots mobiles en utilisant la localisation relative

**Résumé** – Dans ce chapitre nous étudions la conception d’un contrôleur à base de la logique floue et ceci dans le but de commander les deux types de robots mobiles à roues qui sont déjà présentés dans le chapitre I. Le contrôleur assure une navigation topologique qui se base sur un modèle interne de l’environnement qui respecte les caractéristiques cinématiques du robot. Son objectif principal est donc d’aboutir à un déplacement autonome du robot mobile d’une position initiale vers une autre position désirée. Le contrôleur flou permet de déterminer les consignes de vitesses à appliquer aux moteurs afin d’assurer l’arrivée du robot au but. Une validation des résultats théoriques est réalisée par des expériences menées sur un mini robot mobile (Khepera II), ensuite sur un robot de taille plus importante (fauteuil roulant pour handicapés) et enfin sur un robot omnidirectionnel. Une interface graphique est développée pour faciliter les simulations et les expériences sur le robot mobile.

**Mots Clés** – Contrôleur flou, Robot mobile, Khepera II, fauteuil, ROMNI, Navigation.

### II.1 Introduction

La conception de robots mobiles et en particulier de véhicules autonomes à roues est un domaine de recherche qui est encore ouvert. Il présente des enjeux industriels

considérables. En effet, ces véhicules sont de plus en plus utilisés dans l'industrie comme moyens de transport ou d'inspection. Ils sont utilisés dans le domaine civil pour venir en aide à certaines personnes handicapés [29]. Ils sont aussi particulièrement adaptés à des interventions en environnement hostile (exemples : nucléaire, sous marin , espace) [30]. En robotique l'utilisation de la théorie des sous ensembles flous est très répandue [31], [32], [33], [?]. Le domaine d'application qui nous concerne est la navigation d'un robot mobile à roue. L'objectif est de donner plus d'autonomie à un robot mobile pour arriver à destination.

Par ailleurs, la navigation autonome d'un robot mobile consiste à trouver une trajectoire permettant d'aller d'une position initiale à une position finale désirée tout en évitant d'éventuels obstacles. Les robots mobiles, connaissant leur position, se déplacent d'un point à un autre avec diverses contraintes. Dans certains cas, seule l'atteinte d'un point final est importante, sans se préoccuper du trajet. Dans d'autres cas le trajet parcouru est important tout comme sa dépendance du temps. Les techniques les plus employées pour la navigation des robots mobiles utilisent un ou plusieurs contrôleurs flous [34], [35], [36], neuro-flous [37] etc. On peut trouver par exemple un contrôleur qui permet au robot d'atteindre son objectif, un autre qui lui permet d'éviter un obstacle s'il se trouve sur son chemin. La majeure partie de ces contrôleurs est seulement validée par des simulations ou quelques fois sur des robots mobiles. Dans ce chapitre, un contrôleur basé sur la logique floue est synthétisé en simulation en utilisant le modèle cinématique. L'optimisation de ces contrôleurs est rarement étudiée. L'apport de notre étude est l'optimisation d'un contrôleur flou par une méthode classique basée sur l'algorithme du gradient. Enfin l'implémentation de ce contrôleur sur différents types de robots mobiles a été réalisée pour vérifier et comparer les résultats trouvés en simulation.

Une interface de supervision et de commande a été développée pour simuler et expérimenter l'algorithme de commande.

## II.2 Synthèse du contrôleur flou pour un robot unicycle : Khepera II

Le premier robot mobile étudié est le Khepera II : c'est une plateforme unicycle à roues différentielles dont le modèle cinématique est décrit dans le premier chapitre. Le robot est appelé à atteindre un point objectif à partir d'une position donnée comme le montre la figure II.1. La distance  $d$  et l'angle  $\varphi$  sont exprimés par les relations suivantes :

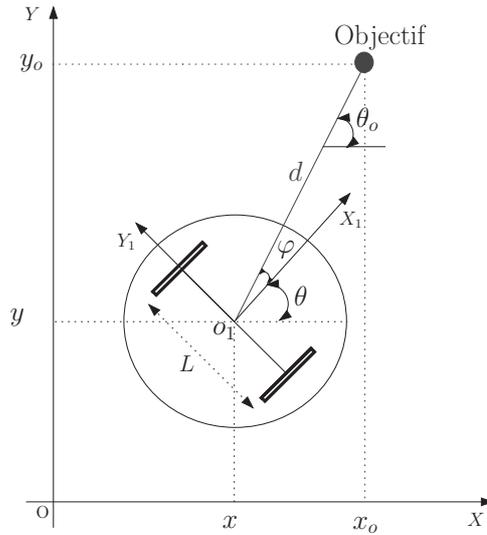


FIG. II.1 – Configuration du robot par rapport à un point objectif.

$$d = \sqrt{(x_o - x)^2 + (y_o - y)^2} \quad (\text{II.1})$$

$$\varphi = \theta_o - \theta \quad (\text{II.2})$$

avec :

$$\theta_o = \arctan \frac{(y_o - y)}{(x_o - x)} \quad (\text{II.3})$$

où :

- $d$  est la distance entre la position du robot et le point objectif
- $\varphi$  est la différence entre l'angle  $\theta_o$  de l'objectif et l'angle  $\theta$  du robot.

De point de vue automatique notre système de commande possède deux entrées et trois sorties : les entrées sont  $x_o$  et  $y_o$  qui sont les coordonnées d'un point objectif et les trois sorties  $x$ ,  $y$  et  $\theta$  qui sont les coordonnées du robot mobile définies dans un repère de référence. Le schéma de commande est donné par la figure II.2.

Le contrôleur flou a deux entrées : la distance  $d$  et l'angle  $\varphi$ . Les sorties sont les vitesses  $V_d$  et  $V_g$  à appliquer respectivement sur les arbres des deux moteurs droite et gauche. Les grandeurs d'entrées et de sorties du contrôleur flou sont déjà définies. On va étudier les différents constituants d'un contrôleur flou à savoir le choix des fonctions d'appartenance, la partition floue des variables d'entrées, l'élaboration d'une base de règles et le choix d'une méthode de défuzzification.

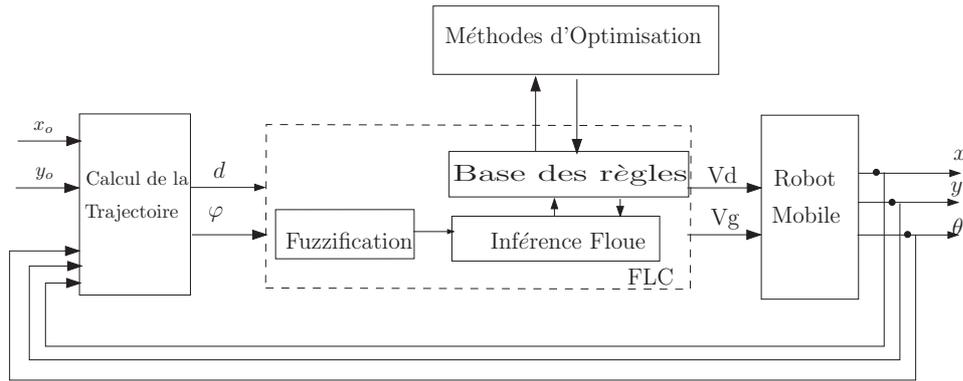


FIG. II.2 – Schéma de commande d'un robot mobile Khepera II par un contrôleur flou optimisé.

### II.2.1 Choix des fonctions d'appartenance des entrées floues

Il y a plusieurs fonctions d'appartenance (triangulaires, gaussiennes, trapézoïdales...) qu'on peut attribuer aux deux entrées ( $d$  et  $\varphi$ ) du contrôleur flou. En s'appuyant sur la littérature on trouve que les fonctions les plus utilisées sont les fonctions triangulaires ou gaussiennes. Les fonctions d'appartenance de type triangulaire (figure II.3) sont caractérisées par trois paramètres  $a$ ,  $b$  et  $c$  dont la fonction est définie par la l'équation II.4.

$$\text{Triangle}(x, a, b, c) = \max \left[ \min \left( \frac{x - a}{b - a}, \frac{c - x}{c - b} \right), 0 \right] \quad (\text{II.4})$$

Les fonctions d'appartenances de type gaussienne (figure II.4) sont caractérisées par deux paramètres  $a$  et  $\sigma$  avec  $a$  le centre et  $\sigma$  la largeur de la gaussienne. L'équation de la fonction gaussienne est définie par l'équation II.5.

$$\text{Gauss}(x, a, \sigma) = \exp \left( - \left( \frac{x - a}{\sigma} \right)^2 \right) \quad (\text{II.5})$$

Pour faire le choix entre ces deux types de fonctions on a mené des simulations sous Matlab. Cette étape consiste à spécifier le domaine de variation des variables d'entrées ( $d$  et  $\varphi$ ) : l'univers de discours, que l'on divise en intervalles (sous ensembles flous ou valeurs linguistiques). Cette répartition, qui consiste à fixer le nombre de ces valeurs et les distribuer sur le domaine, est en faite basée sur la connaissance du système et selon la précision désirée.

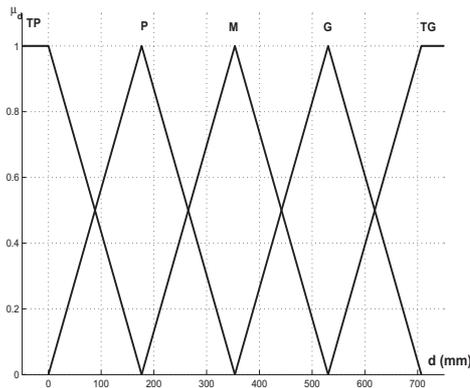


FIG. II.3 – Fonction d'appartenance de type triangulaire :  $\text{triang}(a,b,c)$ .

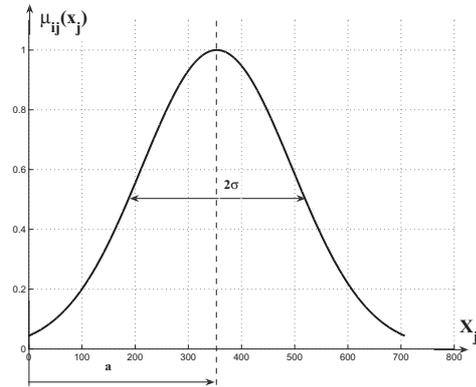


FIG. II.4 – Fonction d'appartenance de type gaussiennes :  $\text{gauss}(x,250,200)$ .

- La variable  $d$  varie de zéro à 700, car l'espace de navigation du robot Khepera II est un carré de 500 mm de côté. On lui a associé cinq sous-ensembles : TP : Très Petite, P : Petite, M : Moyenne, G : Grande, TG : Très Grande. Les fonctions d'appartenance objet de l'étude sont de type triangulaires (figure II.5) ou gaussiennes (figure II.6).

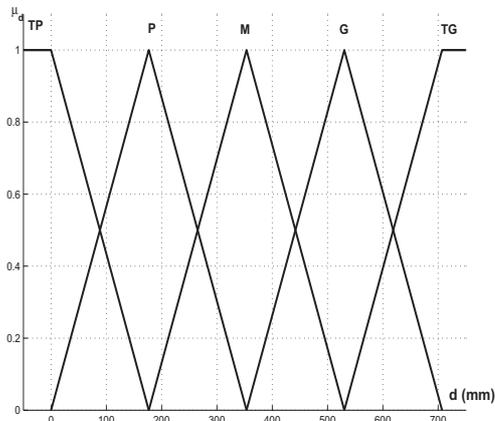


FIG. II.5 – Fonctions d'appartenances triangulaires de la distance  $d$ .

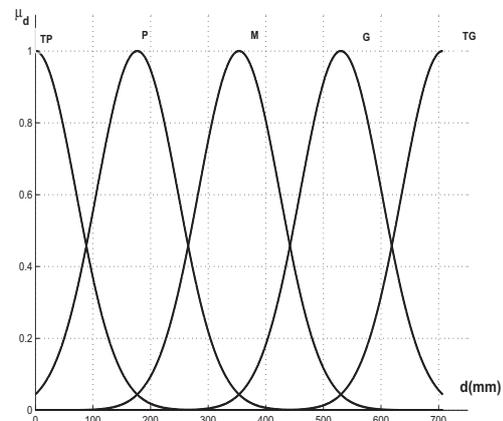
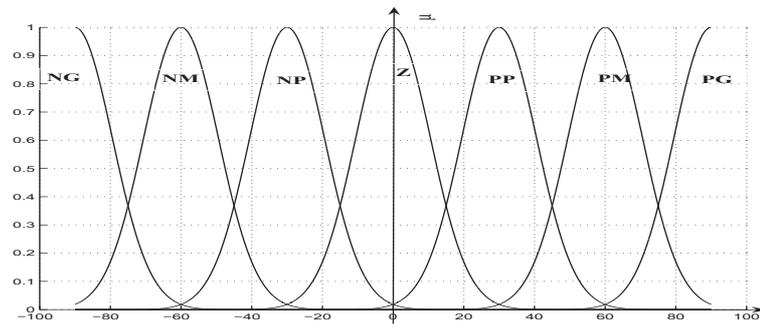
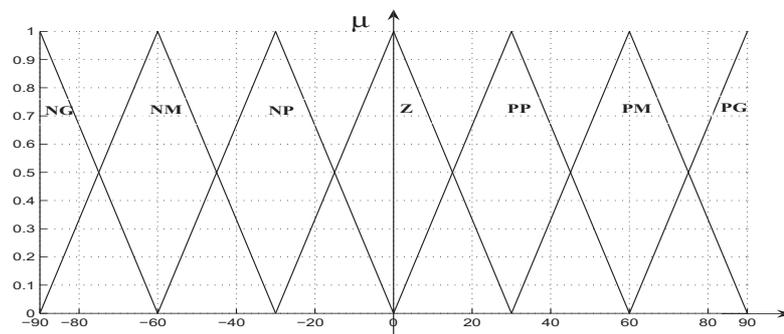


FIG. II.6 – Fonctions d'appartenances gaussiennes de la distance  $d$ .

- La variable  $\varphi$  varie de  $-90^\circ$  à  $+90^\circ$ . On lui a associé sept sous-ensembles flous : NG : Négatif Grand, NM : Négatif Moyen, NP : Négatif Petit, Z : Zéro, PP : Positif Petit, PM : Positif Moyen, PG : Positif Grand. La figure II.7 donne ces fonctions d'appartenance dans le cas des fonctions d'appartenance de type gaussiennes et la figure II.8 donne ces fonctions dans le cas des fonctions d'appartenance de type triangulaire.

FIG. II.7 – Fonctions d'appartenances gaussiennes de l'angle  $\varphi$ .FIG. II.8 – Fonctions d'appartenances triangulaires de l'angle  $\varphi$ .

Pour choisir le meilleur type des deux fonctions d'appartenance une simulation sous Matlab 7 a été effectuée. D'après la figure II.9, on remarque que les deux types de fonctions donnent des résultats très semblables. Mais les fonctions d'appartenance de type gaussiennes donnent des trajectoires légèrement meilleures que celles obtenues avec le type triangulaire, résultant une distance parcourue légèrement plus faible.

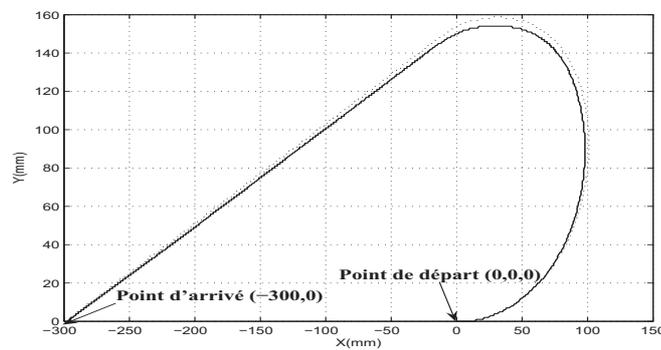


FIG. II.9 – Trajectoire avec les fonctions gaussiennes (trait continu), trajectoire avec les fonctions triangulaires (trait discontinu) .

## II.2.2 Définition de la base des règles du contrôleur flou

Cette étape concerne l'élaboration des règles, pour définir le comportement attendu du robot selon ses paramètres internes qui se présentent par les valeurs des encodeurs. Pour chacune des combinaisons des valeurs des variables d'entrées  $d$  et  $\varphi$ , une action sur les variables de sorties lui est associée. Ces règles sont de type :

$$\text{Si } (d \text{ est } A_i) \text{ et } (\varphi \text{ est } B_i) \text{ Alors } (V_d = y_i \text{ et } V_g = z_i)$$

Pour  $i = 1 \dots r$  avec  $r$  le nombre de règles.

L'ensemble des règles floues «situation/action», se trouve dans le tableau II.1. Ce tableau est construit à la suite de plusieurs simulations et expérimentations.

TAB. II.1 – Tableau d'inférence linguistique pour rejoindre l'objectif : cas du Khepera II.

| $V_g \& V_d$ |    | $\varphi$ |       |       |       |       |       |       |       |       |       |       |       |       |       |
|--------------|----|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|              |    | NG        |       | NM    |       | NP    |       | Z     |       | PP    |       | PM    |       | PG    |       |
|              |    | $V_g$     | $V_d$ | $V_g$ | $V_d$ | $V_g$ | $V_d$ | $V_g$ | $V_d$ | $V_g$ | $V_d$ | $V_g$ | $V_d$ | $V_g$ | $V_d$ |
| $d$          | TP | G         | P     | M     | P     | P     | Z     | Z     | Z     | Z     | P     | P     | M     | P     | G     |
|              | P  | TG        | M     | G     | P     | M     | P     | P     | P     | P     | M     | P     | G     | M     | TG    |
|              | M  | TG        | G     | TG    | M     | G     | P     | M     | M     | P     | G     | M     | TG    | G     | TG    |
|              | G  | TG        | G     | TG    | G     | TG    | M     | G     | G     | M     | TG    | G     | TG    | G     | TG    |
|              | TG | TG        | G     | TG    | G     | TG    | G     | TG    | TG    | G     | TG    | G     | TG    | G     | TG    |

## II.2.3 Sorties du contrôleur flou

Une fois la mise en place des fonctions d'appartenance et l'établissement des règles définissant le comportement du régulateur ont été effectués, on passe à la sélection d'une méthode de défuzzification. C'est cette étape qui permet de transformer les valeurs de commande du domaine flou vers le domaine réel (variables physiques). Ce choix est généralement conditionné par un compromis entre facilité d'implémentation et performance de calcul. Dans ce travail nous avons utilisé la méthode de Sugeno d'ordre 0 [34],[38]. Les vitesses sont données par les relations II.6 et II.7 ( $\alpha_i$  et  $\beta_i$  sont les degrés d'activation de la règle  $i$ ).

$$V_d = \frac{\sum_{i=1}^r \alpha_i y_i}{\sum_{i=1}^r \alpha_i} \quad (\text{II.6})$$

$$V_g = \frac{\sum_{i=1}^r \beta_i z_i}{\sum_{i=1}^r \beta_i} \quad (\text{II.7})$$

Après plusieurs tests en simulation et en expérimentations avec le robot mobile Khepera II, on a attribué des valeurs numériques aux variables linguistiques (voir figure II.10).

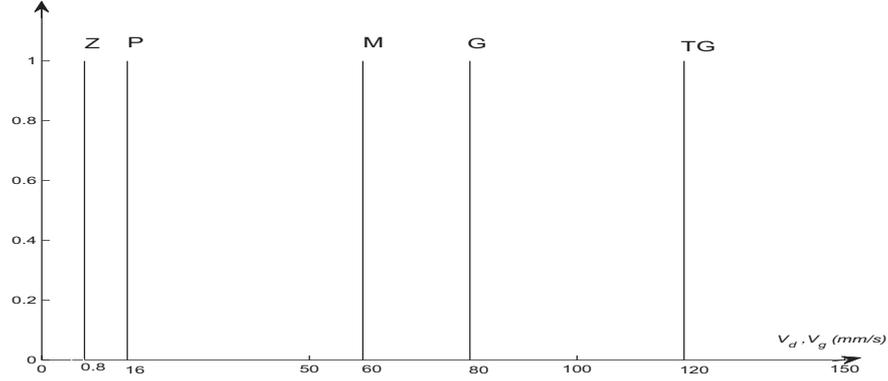


FIG. II.10 – Variables de sorties des vitesses du robot Khepera II.

## II.2.4 Optimisation par la méthode du gradient

Nous avons utilisé la méthode du gradient pour l'optimisation des valeurs du tableau d'inférence II.1, avec les valeurs des variables linguistiques de la figure II.10. Pour réduire la distance de la trajectoire parcourue, on a choisi de minimiser le critère  $J$  :

$$J = \frac{1}{2} \int (x_o - x)^2 + (y_o - y)^2 dt \quad (\text{II.8})$$

$J$  exprime la distance entre la position courante du robot et la position désirée. L'ajustement des conclusions de la table d'inférence est réalisée en utilisant les équations suivantes :

$$\begin{cases} y_{i(k)} = y_{i(k-1)} - \varepsilon \frac{\partial J}{\partial y_i} \\ z_{i(k)} = z_{i(k-1)} - \varepsilon \frac{\partial J}{\partial z_i} \end{cases} \quad (\text{II.9})$$

En calculant les dérivées partielles on a :

$$\begin{cases} \frac{\partial J}{\partial y_i} = T^2 \frac{V_d + V_g}{2L} \frac{\alpha_i}{\sum_{i=1}^r \alpha_i} [(x - x_o) \sin \alpha - (y - y_o) \cos \theta] \\ \frac{\partial J}{\partial z_i} = T^2 \frac{V_d + V_g}{2L} \frac{\beta_i}{\sum_{i=1}^r \beta_i} [-(x - x_o) \sin \alpha + (y - y_o) \cos \theta] \end{cases} \quad (\text{II.10})$$

### II.2.5 Résultats de simulation

Avant de tester les performances du contrôleur flou sur le robot Khepera II, une simulation sous Matlab a été effectuée. L'ajustement des conclusions par l'algorithme du gradient a été faite en simulation. La période d'échantillonnage  $T$  est égale à 0,1s. Alors que le pas d'itération  $\varepsilon$  est de 0,5. Dans ces simulations, plusieurs configurations de la cible et de l'angle d'orientation de départ du robot étaient considérées. Les meilleures performances sont obtenues dans le cas où l'orientation du robot par rapport à la cible est la plus grande. Plusieurs tests, pour différentes configurations de l'orientation initiale et de la position désirée ont été effectués. Dans ces simulations, on s'est rapproché le plus possible des conditions des expériences pour pouvoir comparer et analyser les résultats. Pour cela on a considéré que le robot part toujours du même point (origine) de coordonnées (0,0) mais avec différentes orientations. La figure II.11 donne le résultat de la simulation de plusieurs trajectoires du robot mobile partant d'un même point initial mais pour différents points d'arrivées.

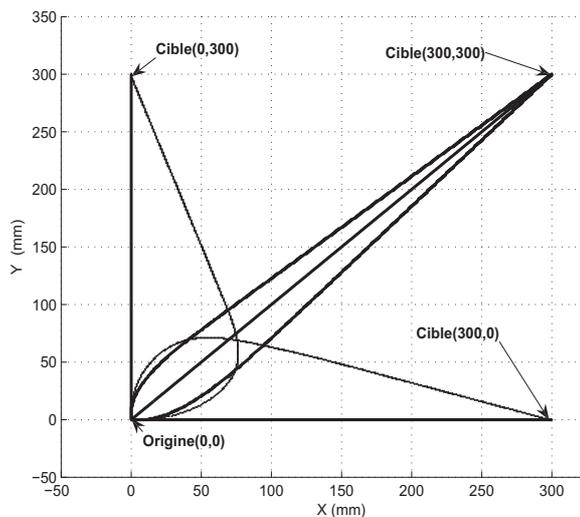


FIG. II.11 – Différentes trajectoires du robot mobile pour le même point origine (0,0) avec différentes orientations.

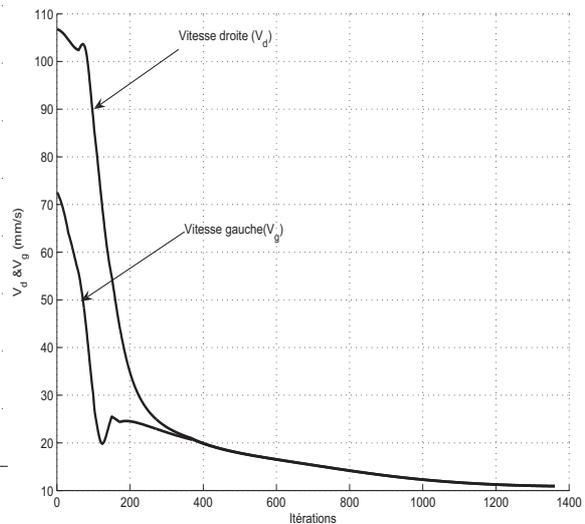


FIG. II.12 – Vitesse droite et gauche pour une trajectoire partant de (0,0) à (0,300).

On remarque que le robot n'est pas dirigé vers son objectif, il y a deux cas de figures :

- L'objectif se trouve devant et à gauche du robot. Le contrôleur doit fournir une valeur pour la vitesse droite plus importante que celle pour la roue gauche afin que le robot puisse s'orienter vers la cible comme le montre la figure II.12.

- L'objectif se trouve devant et à droite du robot. Le contrôleur doit fournir une valeur pour la vitesse gauche plus importante que celle pour la roue droite pour que le robot puisse s'orienter vers la cible.

Dans tous les cas, on remarque qu'une fois le robot se trouve orienté vers sa cible, les vitesses des roues deviennent égales jusqu'à l'arrivée à destination (voir figure II.12). A noter aussi que dans les cas où le robot n'est pas orienté vers la cible, la commande la plus simple consiste à faire tourner le robot autour de son centre de gravité pour le diriger vers la cible, puis lui appliquer deux vitesses identiques sur chaque roue pour le ramener au point objectif. Cette méthode ne nous convient pas car elle décompose la trajectoire en plusieurs mouvements et ne nous permet pas la commande du robot dans le cas où la cible est mobile. Dans la suite on va présenter les résultats expérimentaux de commande du robot Khépéra II avec le contrôleur flou. La figure II.13 représente l'évolution de la distance à la cible en fonction du nombre d'itérations pendant la phase d'optimisation. Ce résultat montre bien que la distance entre le point de départ et le point d'arrivée converge vers zéro.

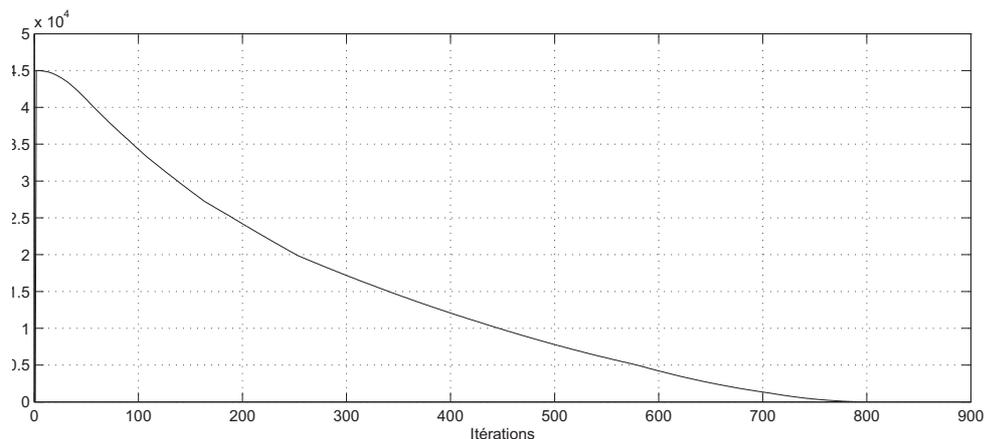


FIG. II.13 – Evolution de la distance à la cible en fonction du numéro d'itération

## II.2.6 Résultats Expérimentaux

Les expériences ont été réalisées avec le robot Khepera II [39]. La figure II.14 montre l'environnement expérimental du robot Khepera II. L'espace de navigation du robot est de dimension 600 mm  $\times$  600 mm. La seule précaution qu'il faut prendre lors de la commande du robot en temps réel c'est qu'avant d'envoyer une deuxième commande de vitesse, s'assurer que les vitesses des roues ont atteint la première commande. La figure II.15 montre un résultat expérimental obtenu pour un point objectif défini par les coordonnées

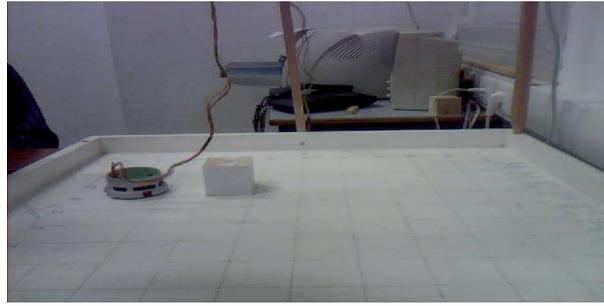


FIG. II.14 – Environnement expérimental du robot mobile Khepera II.

( $x_o = 0$  et  $y_o = 300$  mm). La position initiale du robot est définie par les coordonnées ( $x = 300$  mm et  $y = 300$  mm). La courbe bleue représente une trajectoire réelle du robot non optimisée. Les courbes vertes et noires représentent respectivement les résultats de simulation et les résultats expérimentaux obtenus en utilisant le contrôleur flou optimisé.

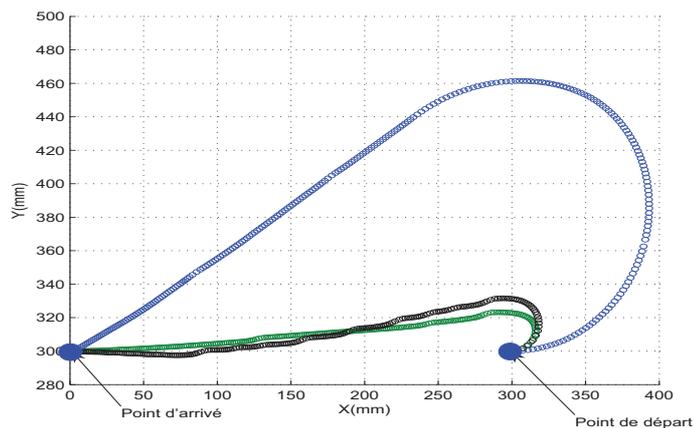


FIG. II.15 – Trajectoires expérimentales obtenues. En vert et noir : avec optimisation ; en bleu : sans optimisation.

## II.3 Synthèse du contrôleur flou pour un robot unicycle : fauteuil roulant

La deuxième plateforme mobile utilisée pour tester les performances du contrôleur flou décrit dans la section précédente est un fauteuil roulant pour handicapés. Ce robot est également une plateforme unicycle à roues différentielles, ainsi les mêmes variables de commande (entrées et sorties du contrôleur flou) appliquées sur le Khepera II seront

utilisées pour le fauteuil roulant. Dans cette partie, nous allons décrire les paramètres modifiés à savoir les fonctions d'appartenance et la base de règles floues.

### II.3.1 Fonctions d'appartenance des entrées du contrôleur flou

La partition floue des variables d'entrées consiste à spécifier les domaines de variations de ces variables ( $d$  et  $\varphi$ ).

- La variable  $d$  varie de zéro à 19 000 mm, car l'espace de navigation du fauteuil roulant est un laboratoire de dimension 14 m de côté. On lui a associé cinq sous-ensembles : TP : Très Petite, P : Petite, M : Moyenne, G : Grande, TG : Très Grande. Les fonctions d'appartenance objet de l'étude sont de type gaussiennes (figure II.16).

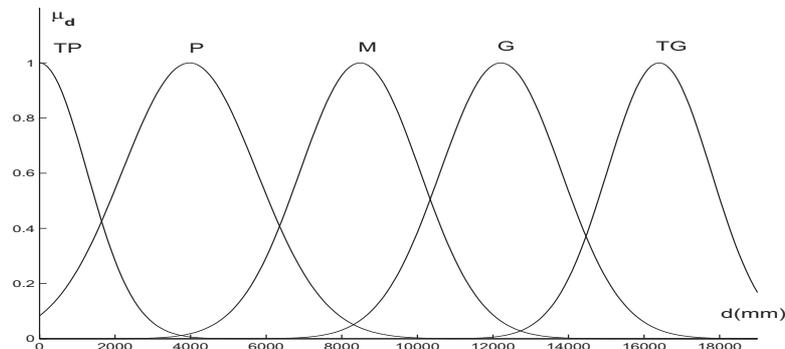


FIG. II.16 – Les fonctions d'appartenance de la distance ( $d$ ).

- La variable  $\varphi$  varie de  $-180^\circ$  à  $+180^\circ$ . On lui a associé sept sous-ensembles flous : NG : Négatif Grand, NM : Négatif Moyen, NP : Négatif Petit, Z : Zéro, PP : Positif Petit, PM : Positif Moyen, PG : Positif Grand. La figure II.17 donne ces fonctions d'appartenance.

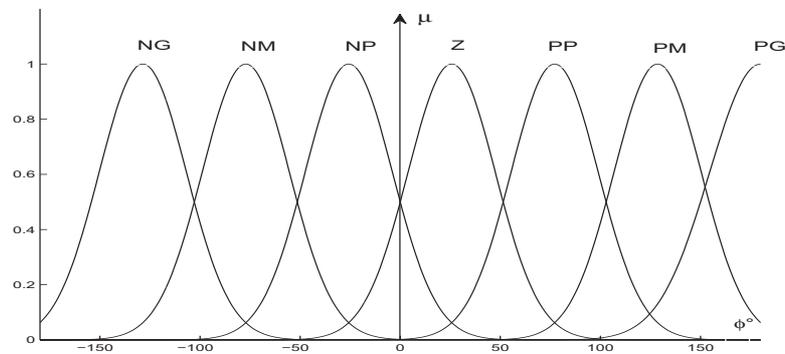


FIG. II.17 – Fonctions d'appartenance de l'angle  $\varphi$  du fauteuil.

### II.3.2 Définition de la base des règles du contrôleur flou

L'ensemble des règles floues générées en se basant sur une relation « situation/action », est résumé dans le tableau II.2. Ce tableau est construit à la suite de plusieurs simulations et expérimentations sur le fauteuil roulant pour handicapés.

TAB. II.2 – Tableau des inférences linguistiques pour rejoindre l'objectif : cas du fauteuil.

| $V_g \& V_d$ |    | $\varphi$ |       |       |       |       |       |       |       |       |       |       |       |       |       |
|--------------|----|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|              |    | NG        |       | NM    |       | NP    |       | Z     |       | PP    |       | PM    |       | PG    |       |
|              |    | $V_g$     | $V_d$ | $V_g$ | $V_d$ | $V_g$ | $V_d$ | $V_g$ | $V_d$ | $V_g$ | $V_d$ | $V_g$ | $V_d$ | $V_g$ | $V_d$ |
| $d$          | TP | G         | P     | M     | P     | P     | Z     | Z     | Z     | Z     | P     | P     | M     | P     | G     |
|              | P  | TG        | M     | G     | P     | M     | P     | P     | P     | M     | P     | G     | M     | TG    |       |
|              | M  | TG        | G     | TG    | M     | G     | P     | M     | M     | P     | G     | M     | TG    | G     | TG    |
|              | G  | TG        | G     | TG    | G     | TG    | M     | G     | G     | M     | TG    | G     | G     | G     | TG    |
|              | TG | TG        | G     | TG    | G     | TG    | TG    | G     | TG    | G     | TG    | G     | TG    | G     | TG    |

### II.3.3 Sorties du contrôleur flou

Après des tests de simulation avec le fauteuil roulant, on a attribué des valeurs numériques aux variables linguistiques de sortie qui sont les vitesses droites et gauches ( $V_d$  et  $V_g$ ) présentées dans la figure II.18. Les sous-ensembles flous utilisés sont en nombre de cinq : Z : Zéro, P : Petite, M : Moyenne, G : Grande, TG : Très Grande.

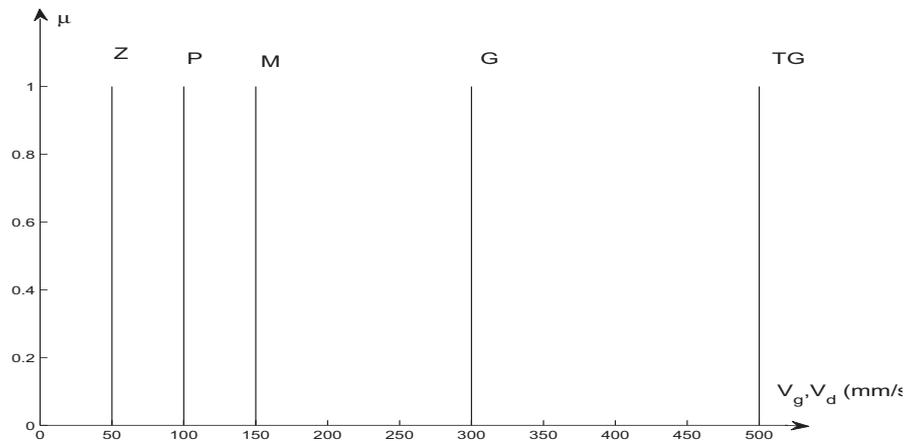


FIG. II.18 – Variables de sorties ( $V_g$  et  $V_d$ ) du fauteuil roulant.

Ensuite, nous avons appliqué la méthode d'optimisation avec le gradient afin de minimiser la distance parcourue par le robot. Le principe de cette méthode d'optimisation est le même que celui appliqué sur le robot Khepera II. Ce principe a été déjà décrit dans la section précédente de ce chapitre.

### II.3.4 Résultats expérimentaux

La figure II.19 montre un résultat expérimental obtenu en appliquant l'algorithme de commande floue optimisé. Dans cet exemple le fauteuil doit atteindre un point objectif défini par les coordonnées ( $x_o = 4\ 000$  mm et  $y_o = 0$  mm). Le point de départ du fauteuil est défini par les coordonnées ( $x = 0$ ,  $y = 0$  et  $\theta = \pi/2$ ). La courbe noire représente les résultats de simulation alors que la courbe rouge représente une trajectoire réelle du fauteuil. Dans cet exemple, on remarque une différence entre les courbes de simulation et expérimentales essentiellement quand le robot vise à changer sa direction (position :  $x = 300$  mm et  $y = 400$  mm). Ceci s'explique par le fait que le robot présente une inertie importante qui agit principalement quand il va tourner d'un angle supérieur à  $\pi/2$ . Mais malgré cette erreur, il arrive juste après 500 mm à rejoindre sa trajectoire de début. Ceci prouve l'efficacité de ce contrôleur même pour un robot de masse importante. A la fin, le robot atteint le point objectif avec une déflexion selon l'axe des Y de 50 mm qui est une erreur acceptable par rapport aux dimensions du fauteuil.

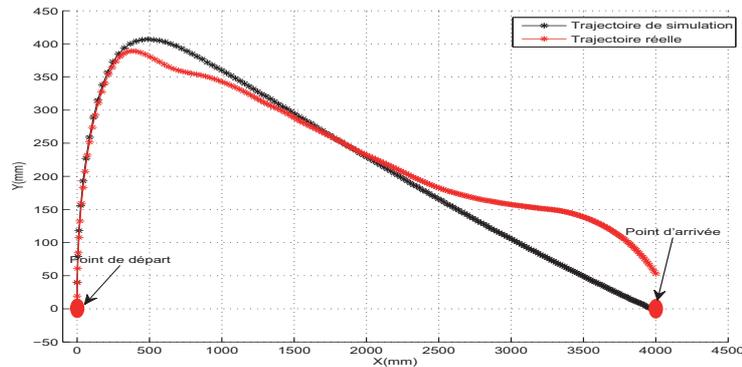


FIG. II.19 – Trajectoire optimisée parcourue par le fauteuil roulant (la trajectoire est celle du centre d'essieu)

La figure II.20 montre respectivement les vitesses théoriques et réelles du fauteuil roulant. Les courbes obtenues montrent un retard au début entre les deux vitesses. Ce retard est dû d'une part au temps de réponse du fauteuil qui est égal à 0,7 s. D'autre part,

le robot nécessite un certain temps pour atteindre la vitesse de début ce qui engendrera un retard sur toutes les positions obtenues.

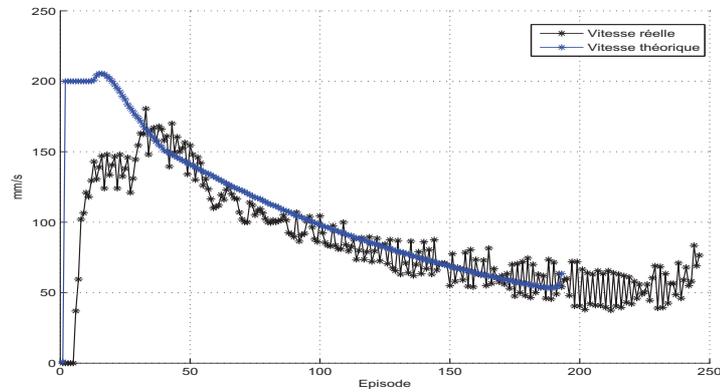


FIG. II.20 – Vitesses réelle et théorique du fauteuil roulant

## II.4 Synthèse du contrôleur flou pour un robot omnidirectionnel : ROMNI

La troisième plateforme mobile étudiée pour tester les performances du contrôleur flou est un robot omnidirectionnel : ROMNI (voir figure II.21). Ce robot présente des caractéristiques cinématiques différentes de celles d'un robot unicycle, ainsi les variables du système de commande seront modifiées.

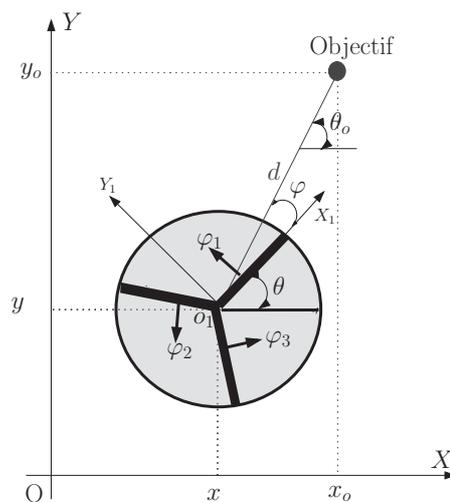


FIG. II.21 – Configuration du ROMNI par rapport à un point objectif.

Le contrôleur flou a deux entrées : la distance  $d$  et l'angle  $\varphi$ . Les sorties sont les vitesses des essieux  $\varphi_1$ ,  $\varphi_2$  et  $\varphi_3$  à appliquer respectivement sur les trois essieux du robot. Dans cette partie, nous allons décrire les paramètres modifiés à savoir les entrées et les sorties du contrôleur flou, les fonctions d'appartenances et la base de règles floues.

### II.4.1 Fonctions d'appartenance des entrées du contrôleur flou

La partition floue des variables d'entrées consiste à spécifier les domaines de variations de ces variables ( $d$  et  $\varphi$ ).

- La variable  $d$  varie de zéro à 19 000 mm, car l'espace de navigation du ROMNI est un laboratoire de dimension 14 m de côté. On lui a associé cinq sous-ensembles : TP : Très Petite, P : Petite, M : Moyenne, G : Grande, TG : Très Grande. Les fonctions d'appartenance objet de l'étude sont de type gaussiennes comme illustré dans la figure II.22.

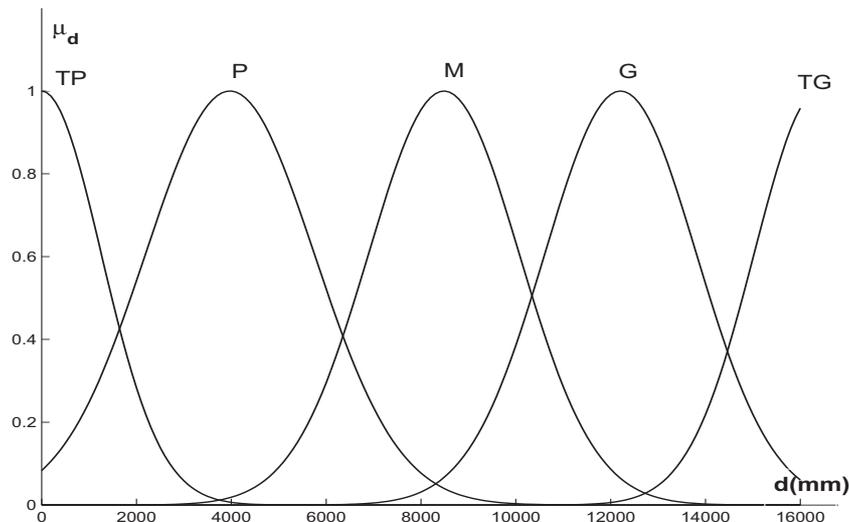
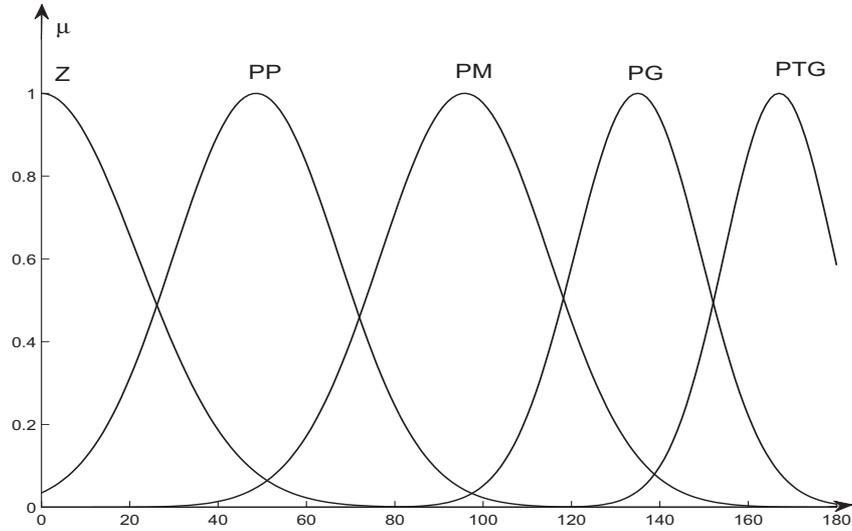


FIG. II.22 – Fonctions d'appartenance de type gaussiennes de la distance  $d$ .

- La variable  $\varphi$  varie de 0 à  $+180^\circ$ . On lui a associé cinq sous-ensembles flous : PTG : Positif Très Grand, PG : Positif Grand, PM : Positif Moyen, PP : Positif Petit, Z : Zéro. La figure II.23 montre ces fonctions d'appartenance de type gaussiennes.

FIG. II.23 – Fonctions d'appartenance de l'angle  $\varphi$ .

### II.4.2 Définition de la base des règles du contrôleur flou

Pour pouvoir déterminer la base de règles floues, nous avons considéré trois cas de figure pour la navigation du ROMNI (en se basant sur le modèle cinématique inverse (MCI)) (Thèse Mourieux 2006 [3]) :

- L'objectif se trouve devant le robot avec  $\varphi = 0$ . Ainsi, le robot doit se déplacer en translation selon  $\vec{X}$  avec une vitesse  $\dot{x}$ . Le contrôleur doit fournir les valeurs de vitesses suivantes :

$$\dot{\varphi}_1 = 0; \dot{\varphi}_2 = -\dot{\varphi}_3 = \dot{x} \times \frac{\sqrt{3}}{2 \times r} \quad (\text{II.11})$$

- L'objectif se trouve à droite du robot avec  $\varphi = 90^\circ$ . Ainsi, le robot doit se déplacer en translation selon  $\vec{Y}$  avec une vitesse  $\dot{y}$ . Le contrôleur doit fournir les valeurs de vitesses suivantes :

$$\dot{\varphi}_1 = -\frac{\dot{y}}{r}; \dot{\varphi}_2 = \dot{\varphi}_3 = \frac{\dot{y}}{2 \times r} \quad (\text{II.12})$$

- L'objectif se trouve à gauche du robot avec  $\varphi = 180^\circ$ . Ainsi, le robot doit se déplacer en translation selon  $\vec{x}$  avec une vitesse  $\dot{y}$ . Le contrôleur doit fournir les valeurs de vitesses suivantes :

$$\dot{\varphi}_1 = 0; -\dot{\varphi}_2 = \dot{\varphi}_3 = \dot{x} \times \frac{\sqrt{3}}{2 \times r} \quad (\text{II.13})$$

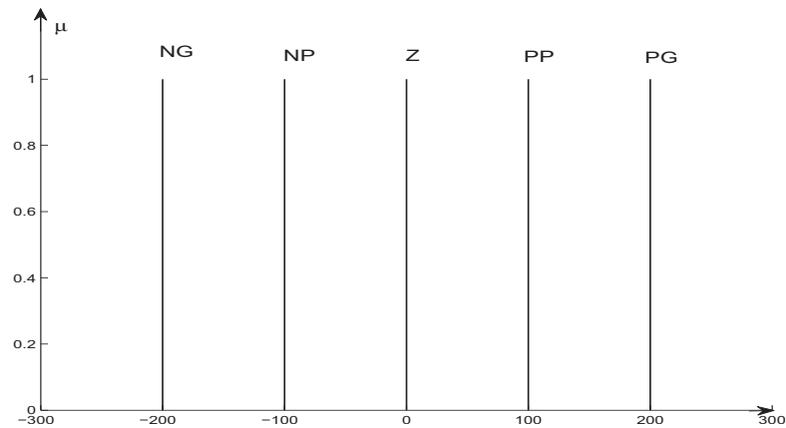
L'ensemble des règles floues «situation/action», se trouve dans le tableau II.3. Ce tableau est construit à la suite de plusieurs simulations.

TAB. II.3 – Tableau des inférences linguistiques pour rejoindre l’objectif : cas du ROMNI.

| $\varphi_1$ & $\varphi_2$ & $\varphi_3$ |    | $\varphi$   |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
|---|----|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|   |    | PTG         |             |             | PG          |             |             | PM          |             |             | PP          |             |             | Z           |             |             |
|   |    | $\varphi_1$ | $\varphi_2$ | $\varphi_3$ |
| $d$                                     | TP | Z           | NP          | PP          | NG          | Z           | PP          | NG          | PP          | PP          | NP          | PP          | Z           | Z           | PP          | NP          |
|   | P  | Z           | NP          | PP          | NG          | Z           | PP          | NG          | PP          | PP          | NP          | PP          | Z           | Z           | PP          | NP          |
|   | M  | Z           | NP          | PP          | NG          | Z           | PP          | NG          | PP          | PP          | NP          | PP          | Z           | Z           | PP          | NP          |
|   | G  | Z           | NG          | PG          | NG          | Z           | PG          | NG          | PG          | PG          | NG          | PG          | Z           | Z           | PG          | NG          |
|   | TG | Z           | NG          | PG          | NG          | Z           | PG          | NG          | PG          | PG          | NG          | PG          | Z           | Z           | PG          | NG          |

### II.4.3 Sorties du contrôleur flou

Après plusieurs tests en simulation avec le ROMNI, on a attribué des valeurs numériques aux variables linguistiques  $\varphi_1$ ,  $\varphi_2$  et  $\varphi_3$  comme présenté dans la figure II.25.

FIG. II.24 – Variables de sorties ( $\varphi_1$ ,  $\varphi_2$  et  $\varphi_3$ ) du ROMNI.

### II.4.4 Résultats de la simulation sur le ROMNI

Plusieurs simulations ont été effectuées afin de vérifier la validité de l’approche de commande sur le robot omnidirectionnel.

- **Expérience 1** : Dans le premier exemple, nous avons considéré un point initial défini par les coordonnées ( $x = 4\,000$  mm,  $y = 4\,000$  mm et  $\theta = 0$ ). Le point objectif

est défini par les coordonnées  $x_o = 8\,700$  mm et  $y_o = 4\,000$  mm. Ainsi, l'objectif se trouve devant le robot avec  $\varphi = 0$ .

Dans une telle situation, les vitesses générées par le contrôleur flou doivent vérifier l'équation II.11 lui permettant de se déplacer en translation selon  $\vec{X}$  avec une vitesse  $\dot{x}$ . Les figures II.25 et II.26 montrent respectivement la position et les vitesses du robot. Nous remarquons que les résultats de simulation vérifient bien la notion de déplacement latéral vu que  $\varphi_1 \simeq 0$  et  $\varphi_2 \simeq -\varphi_3$ .

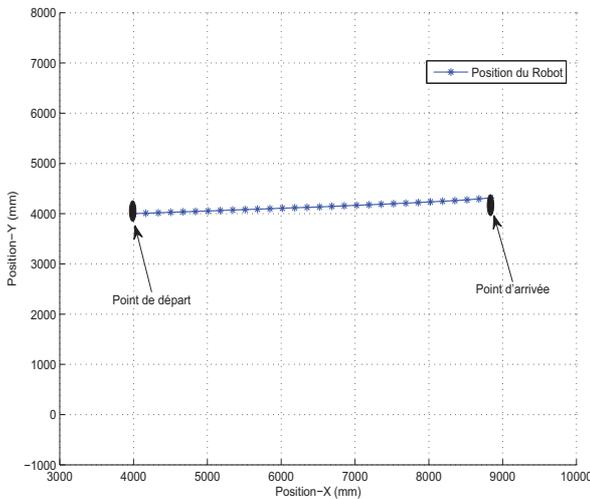


FIG. II.25 – Position du robot avec  $\varphi = 0$ .

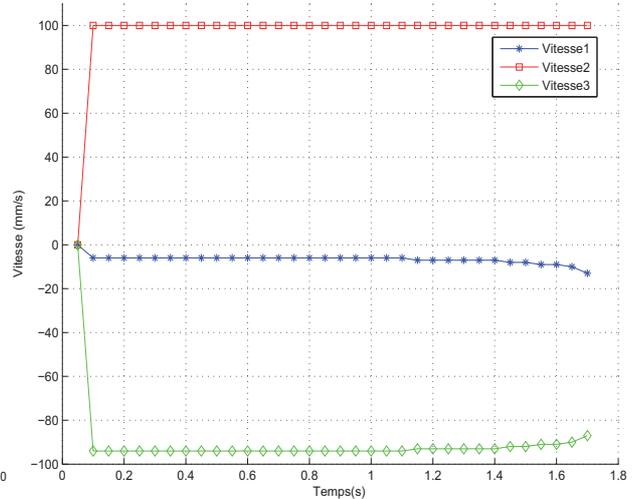


FIG. II.26 – Variation des vitesses pour  $\varphi = 0$ .

- **Expérience 2** : Dans cet exemple le point initial est défini par les coordonnées ( $x = 4\,000$  mm,  $y = 4\,000$  mm et  $\theta = 0$ ) alors que le point objectif est défini par les coordonnées  $x_o = 4\,000$  mm et  $y_o = 8\,000$  mm. En résultat, l'objectif se trouve à droite du robot avec  $\varphi = 90^\circ$ .

Ainsi, les vitesses à appliquer aux essieux doivent vérifier l'équation II.12 lui permettant de se déplacer en translation selon  $\vec{Y}$  avec une vitesse  $\dot{y}$ . Les figures II.27 et II.28 montrent respectivement la position et les vitesses du robot. Les vitesses obtenues en simulation sont :  $\varphi_1 \simeq -200$  mm/s et  $\varphi_2 \simeq \varphi_3 \simeq 100$  mm/s. Ces valeurs vérifient les valeurs théoriques déduites à partir de l'équation II.12.

- **Expérience 3** : Dans cet exemple le point initial est défini par les coordonnées ( $x = 13\,000$  mm,  $y = 2\,000$  mm et  $\theta = 0$ ) alors que le point objectif est défini par les coordonnées  $x_o = 1\,000$  mm et  $y_o = 2\,000$  mm. Nous remarquons que l'objectif se trouve à gauche du robot avec  $\varphi = 180^\circ$ . Ainsi, le robot doit se déplacer

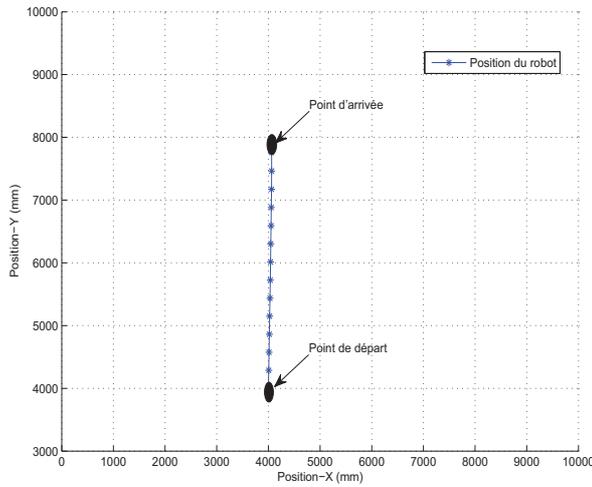


FIG. II.27 – Position du robot avec  $\varphi = 90^\circ$ .

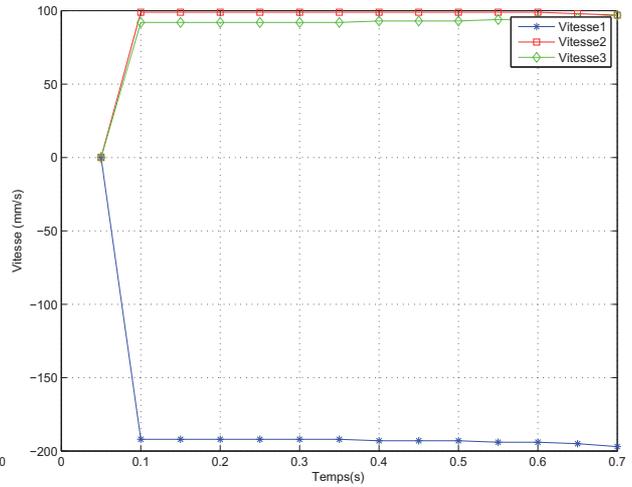


FIG. II.28 – Variation des vitesses pour  $\varphi = 90^\circ$ .

en translation selon  $\vec{x}$  avec une vitesse  $\dot{y}$  tout en respectant l'équation II.13. Les figures II.29 et II.30 montrent respectivement la position et les vitesses du robot. Les vitesses obtenues sont :  $\varphi_1 \simeq 0$  mm/s et  $\varphi_2 \simeq -100$  mm/s et  $\varphi_3 \simeq 100$  mm/s. Les vitesses générées permettent au robot d'atteindre le point objectif avec le moins de manoeuvres possibles.

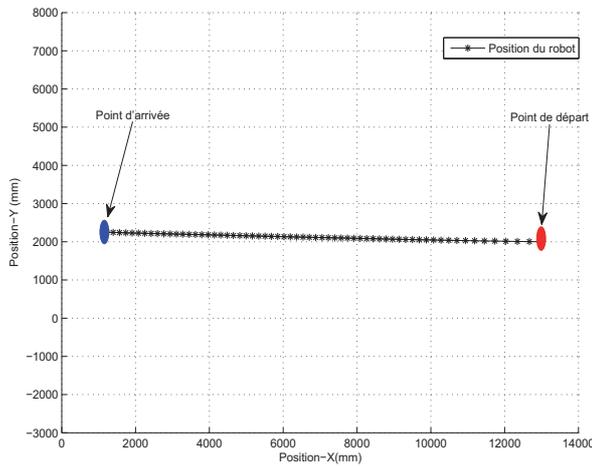


FIG. II.29 – Position du robot avec  $\varphi = 180^\circ$ .

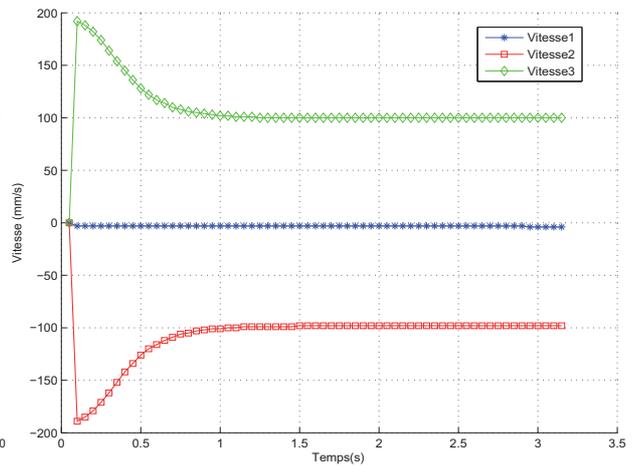


FIG. II.30 – Variation des vitesses pour  $\varphi = 180^\circ$ .

## II.5 Conclusion

Dans ce chapitre nous avons présenté une méthode de synthèse d'un contrôleur flou pour la commande de deux types de robots mobiles à roues : deux robots unicycles et un robot omnidirectionnel. Ce contrôleur est testé par une simulation sous Matlab. Une implémentation de ce contrôleur sur le mini robot mobile Khepera II et sur le fauteuil roulant a validé les résultats de la simulation.

En revanche, les résultats expérimentaux montrent l'influence de la masse du robot sur la robustesse de ce type de commande, essentiellement pour de longues trajectoires. Ainsi, l'application du contrôleur flou sur le robot Khepera (de masse faible) donne des erreurs de précision moins importantes que celles obtenues pour le cas du fauteuil roulant (de masse élevée). Par ailleurs, une comparaison des résultats de simulation obtenus pour les deux types de robots (unicycle et omnidirectionnel) montre aussi l'impact de la forme sur la rapidité du contrôleur proposé. En effet, les réponses de vitesses procurées pour le cas du ROMNI sont plus rapides que celles des deux robots unicycles, essentiellement pour des trajectoires qui présentent un changement de direction.

Ce contrôleur accompagné avec une localisation relative basée sur les mesures odométriques nous a permis d'aboutir à des résultats d'autonomie de commande satisfaisants. Cependant, la précision des mesures de localisation paraît entachée d'erreurs, essentiellement lorsque le robot doit parcourir de longues trajectoires.

Pour remédier à ce problème, nous avons choisi de corriger les mesures proprioceptives par des données extéroceptives provenant d'une Webcam placée sur un plan parallèle au plan de navigation du robot. Ce concept permettant d'assurer une certaine autonomie de décision, sera présenté en détails dans le chapitre suivant.

# Chapitre III

## Contribution à la localisation absolue et la commande floue d'un robot mobile

### Résumé -

Ce chapitre est consacré à l'étude d'une méthode de localisation absolue qui se base sur des données images. Un système de localisation absolue est indispensable, étant donné que le système de localisation relative, basé dans notre cas sur les encodeurs, ne permet pas à lui seul de localiser avec précision le robot mobile. Le système de localisation absolue va nous permettre de préciser automatiquement la position et l'orientation de départ du robot mobile et également d'avoir un système de recalage pour les mesures des encodeurs. Dans ce même contexte, il faut peut être rappeler que le système odométrique détermine les positions courantes du robot mobile à partir d'une position et d'une orientation initiales. La méthode ainsi développée ne permet pas de localiser en temps réel le robot mobile. Cela est résolu par l'application d'un algorithme qui ne traite que la partie utile dans une image. Cette partie correspond à la zone de déplacement du robot.

### Mots Clés -

Localisation absolue, Robot mobile, Khepera II, Navigation d'un robot, Webcam.

### III.1 Introduction

En robotique mobile, être capable de se localiser dans l'environnement dans lequel le robot va agir, est un préalable incontournable pour accéder à l'autonomie de décision. De

plus, la localisation est un problème fondamental de la robotique : en effet, il n'est pas possible de contrôler un robot sans estimation précise de sa position. Pour se localiser, le robot mobile doit être muni d'un certain nombre de capteurs.

Les capteurs utilisés dans la localisation des robots mobiles à roues peuvent être classés en deux types : les capteurs proprioceptifs (odomètres, gyroscopes...) qui permettent d'avoir une information sur l'état interne du robot, et les capteurs extéroceptifs (infrarouges, caméras, ultrasons, lasers, etc.) qui renseignent sur la situation du robot par rapport à son environnement. L'utilisation exclusive des capteurs proprioceptifs paraît extrêmement limitative [40]. Le problème posé par ce type de capteurs est que les erreurs sur la position du robot s'accroissent si on utilise simplement les mesures d'odométrie et la position devient rapidement erronée. En effet, de tels capteurs fournissent uniquement des informations sur l'état interne du système [41]. Ainsi, on ne peut pas espérer localiser précisément un robot en se basant uniquement sur ces mesures, du fait des problèmes de dérives et de précision des capteurs, et des incertitudes associées à la modélisation de l'ensemble robotique. Cependant ces mesures sont soumises à des imprécisions et des aléas : par exemple s'il y a un glissement de l'une ou des deux roues sur lesquels les capteurs sont placés, les informations sur l'état du robot seront erronées. En revanche, l'exploitation des données extéroceptives permet d'avoir une vision plus précise du véhicule dans son environnement, et d'avoir des indications sur les événements extérieurs. Un exemple de solution est la localisation sur amers qui est très utilisée en robotique mobile [42]. Un amer est un élément caractéristique de l'environnement qui peut être naturel ou artificiel. Si l'on dispose suffisamment d'informations sur les positions absolues de ces amers, on peut se localiser de façon absolue. Il existe déjà des systèmes commerciaux de localisation sur amers basés sur la combinaison laser-réflecteurs optiques (exemple : le système NAV 200 de Sick AG). Ce type de système est bien adapté à un usage intérieur. Il nécessite cependant une instrumentation de l'environnement. Les travaux de recherche actuels portent sur l'utilisation d'amers naturels de l'environnement et de la vision [41], [43].

Le système odométrique qui équipe la majeure partie des robots mobiles est en mesure de déterminer la position actuelle en ajoutant le déplacement mesuré grâce aux encodeurs optiques [44]. Ainsi ce système de mesure est tributaire de la position et de l'orientation initiales et intègre les erreurs de mesure ; ce qui nous a conduit à étudier un système de localisation absolue qui permet la détermination automatique de la position et de l'orientation de départ du robot. Ce système peut être aussi utilisé pour faire des points de recalage.

## III.2 Localisation absolue par une caméra

Pour pouvoir effectuer de longs trajets, le robot ne peut pas compter sur la seule intégration de son déplacement pour estimer sa position : il doit pouvoir se localiser dans un repère lié à son environnement. Cette tâche est réalisée par l'utilisation d'une caméra de type webcam placée au dessus de l'environnement de déplacement du robot mobile. L'image acquise se trouve dans un plan parallèle au plan de navigation du robot. Ces deux plans sont distants d'une distance ( $d$ ). On peut dire que le principe de cette localisation par caméra peut être décrit par quatre étapes [45], [46].

- L'acquisition d'une image de référence qui contient quatre amers.
- L'acquisition des images pendant le déplacement du robot.
- L'optimisation du temps de traitement par la sélection d'une zone réduite de l'image.
- La détermination de l'orientation du robot par placement d'une étiquette blanche sur le robot.

En revanche, ces étapes d'acquisition ont nécessité un algorithme de filtrage des données se basant sur les propriétés des filtres non linéaires et de la morphologie mathématique. Ce choix a été fait pour respecter les deux propriétés suivantes. D'une part, sur le plan mathématique, les opérateurs utilisées sont indépendants des types de paysages représentés dans les images. D'autre part, sur le plan physique, les opérateurs utilisés doivent permettre d'extraire des informations à toutes les échelles. Nous allons maintenant rappeler brièvement les principaux concepts de base de la morphologie mathématique indispensables à la compréhension de ce mémoire.

### III.2.1 La morphologie mathématique

De 1964 à 1968, G. Matheron et Jean Serra ont développé à l'Ecole des Mines de Paris, une nouvelle discipline : la Morphologie Mathématique [47], [48]. Cette nouvelle théorie porte sur la caractérisation mathématique des formes qui est un des volets indispensables du traitement d'image. Cette discipline a été initialement conçue pour des images binaires, ensuite elle a été étendue aux images en niveaux de gris. La morphologie mathématique traite les signaux en se basant sur une approche géométrique ce qui la ressemble à une méthode de traitement de signal 2D [49]. Ce type de traitement met en oeuvre certaines caractérisations d'images, notamment les formes et les contours plutôt que d'autres caractéristiques d'une image, comme les variations d'intensité par exemple. Dans le domaine de traitement d'images, la morphologie mathématique est principalement utilisée dans cinq volets :

- l'extraction d'attributs
- la reconnaissance et la détection d'objets ou de contours
- l'organisation spatiale ou volumique d'objets
- la compression
- la télédétection

Les opérations de morphologie mathématique travaillent sur le voisinage d'un point, appelé élément structurant. Celui-ci agit en fait comme une sonde qui se déplace sur l'image afin d'analyser les caractéristiques incluses dans le voisinage qu'il définit [49]. L'élément structurant est défini par un ensemble de paramètres que sont : la taille, la forme, la connexité et l'origine. L'origine de l'élément structurant correspond au pixel de l'image que l'on traite. Différentes opérations de morphologie mathématique utilisant l'élément structurant existent en littérature. Dans ce mémoire nous avons utilisé les opérations suivantes :

- La dilatation/érosion : L'érosion est l'opération duale de la dilatation. Ainsi, faire une dilatation sur un objet ou faire une érosion sur son complémentaire donne le même résultat sur une image. L'érosion est une opération croissante et anti-extensive. Les dilatations et les érosions sont invariantes par translation.
- Ouverture / Fermeture morphologiques : L'ouverture et la fermeture morphologiques sont les deux autres opérateurs de base. Ce sont en fait des compositions de dilatation et d'érosion. L'ouverture morphologique est une transformation composée d'une érosion puis d'une dilatation. La fermeture morphologique est une transformation composée d'une dilatation puis d'une érosion.

### III.2.2 Acquisition d'une image de référence

L'image de référence est une image du plan de navigation du robot contenant quatre amers qui seront utilisés par la suite pour localiser le robot. L'acquisition de cette image doit être précédée par les deux étapes suivantes :

- Placer quatre étiquettes de couleur noire au quatre coins de l'espace de déplacement du robot. Ces quatre étiquettes vont jouer le rôle des amers.
- Mettre au point l'image de référence en jouant sur l'emplacement des quatre étiquettes de telle façon qu'elles soient complètement visible dans cette image (tout en tenant compte de la dilatation de l'image).

Après l'acquisition de l'image de référence, l'étape suivante consiste à appliquer une série de traitements pour obtenir le centre de gravité des quatre étiquettes dans l'image de référence. Une fois que l'on a la position image du centre de gravité de chaque étiquette dans le plan image, on peut faire l'acquisition des images de l'espace de navigation du robot pour déterminer la position  $(x, y)$  du robot et de son orientation  $\theta$  dans le repère de référence. Les traitements appliqués sur l'image de référence sont les suivants : binarisation, filtrage, détection de contours [50] et seuillage.

### III.2.2.1 Niveau de gris

Le niveau de gris d'une image est simplement conçu pour avoir des couleurs qui sont toutes exprimées en gris. En effet, le "gris" est une couleur dans laquelle les composantes : rouge, vert et bleu ont toutes la même intensité dans l'espace RGB (Rouge, Vert et bleu). La méthode la plus simple qui permet la conversion de l'image en couleur à une autre en niveau de gris est de calculer la luminance d'un pixel en appliquant l'équation suivante :

$$Gris = 0.299 \times Rouge + 0.587 \times Vert + 0.114 * Bleu \quad (III.1)$$

Les figures III.1 et III.2 montrent respectivement l'image de référence avant et après détermination du niveau de gris sachant que la taille de cette image est de 640\*480 pixels.



FIG. III.1 – Image de référence contenant les 4 étiquettes avant traitement.



FIG. III.2 – Image de référence après binarisation.

### III.2.2.2 Le filtrage : la technique Sobel

Nous avons utilisé la technique de détection de bords «Sobel» afin de déterminer la magnitude en niveau de gris du gradient de chaque point noté  $I$  dans l'image de référence. Ce filtre permet ainsi de détecter les quatre étiquettes placées sur le robot. Nous avons

utilisé une paire de masque de convolution de taille  $3 \times 3$ , permettant respectivement d'estimer le gradient dans les directions  $X$  et  $Y$  [51]. Ces deux masques sont convolués avec les données images entrantes afin de mesurer les variations d'intensité tout au long des directions horizontales et verticales notées  $E_h$  et  $E_v$ . Ces deux mesures sont par la suite combinées pour estimer la direction et l'amplitude du bord. Le gradient de l'amplitude est estimé comme suit :

$$|Mag| = \sqrt{(E_h \times E_h) + (E_v \times E_v)} \quad (\text{III.2})$$

Les figures III.3 et III.4 montrent l'image en entrée et celle en sortie après l'application du filtrage « Sobel ».

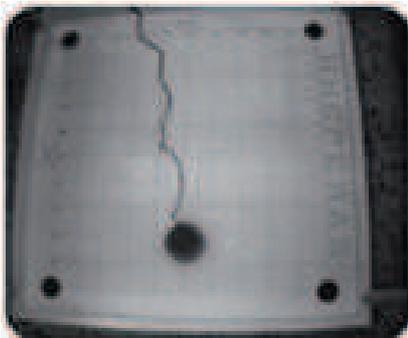


FIG. III.3 – Image de référence en entrée.

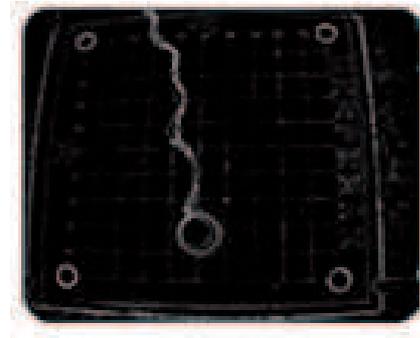


FIG. III.4 – Image de référence après détection de bords avec le filtre « Sobel ».

### III.2.2.3 Le seuillage de l'image

Le seuillage de l'image est une application indispensable à la plupart des applications d'analyse et de traitement d'images. Il consiste à extraire un objet à partir du plan image défini par un arrière plan. Parmi les méthodes de seuillage d'images nous citons le « seuillage à deux niveaux » qui sépare les pixels d'une image à deux régions (i.e. l'objet et l'arrière plan). Une région contient des pixels avec des valeurs de gris plus petites que la valeur du seuil. La seconde région contient des pixels avec des valeurs de gris plus larges que la valeur du seuil.

La combinaison entre la technique « Sobel » et le seuillage permet d'aboutir à un filtrage dont le principe se présente comme suit : soit  $f(i, j)$  le niveau de gris du point défini par les coordonnées pixels  $(i, j)$  et  $T$  est la valeur du seuil. L'application du seuillage

sur ce point permet d'obtenir un nouveau niveau de gris noté  $g(i, j)$  comme suit :

$$\begin{aligned} g(i, j) &= \text{Blanc si } f(i, j) \geq T \\ &= \text{Noir ailleurs} \end{aligned} \tag{III.3}$$

Les figures III.5 et III.6 montrent l'image en entrée et celle en sortie après l'utilisation du seuillage à deux niveaux.

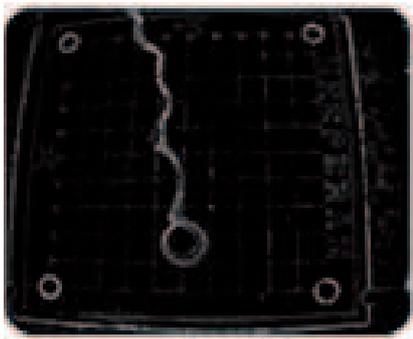


FIG. III.5 – Image de référence en entrée.

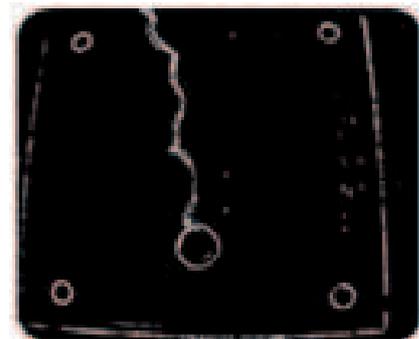


FIG. III.6 – Image de référence après seuillage.

#### III.2.2.4 L'opérateur morphologique : érosion

L'opérateur morphologique «érosion» a été appliqué sur l'image de navigation de la plateforme mobile uniquement pour isoler le robot des étiquettes. L'érosion consiste à considérer deux types de données en entrée. La première entrée est l'image à éroder. La seconde donnée est l'élément structurant. Cet élément détermine avec précision l'effet de l'érosion sur l'image en entrée.

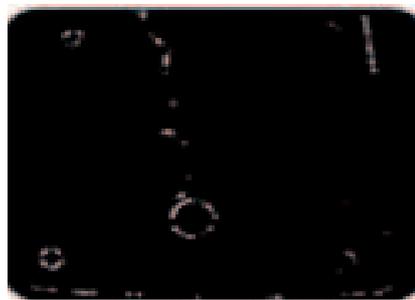


FIG. III.7 – Image de référence après érosion

La figure III.7 montre le résultat obtenu après application de l'érosion sur l'image de référence. L'élément structurant choisi dans cette partie est  $(0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0)$ . C'est un disque de rayon 5 et ceci dans le but de garder la forme circulaire du robot.

### III.2.2.5 La transformation du cercle de Hough (TCH)

Lors de la localisation des quatre étiquettes dans l'image de référence, un algorithme de détection de cercles doit être utilisé. Cet algorithme permettra en premier lieu d'extraire les coordonnées des centres des étiquettes. Puis, il assurera le calcul des coordonnées du robot dans chaque image acquise tout au long de sa navigation. L'algorithme adopté est la transformation de cercle de Hough. Le principe de cette méthode se présente comme suit :

- Détermination des bords
- Pour chaque point de bord, tracer un cercle centré sur ce point et de rayon  $r$ .
- Incrémenter les coordonnées du rayon jusqu'à obtenir un cercle de même périmètre que celui de l'espace des paramètres (qui est en fait de même dimension que celle de la matrice de l'accumulateur)
- Trouver un ou plusieurs maximums dans cette matrice ce qui correspond aux centres des cercles de l'image.

La figure III.8 montre le résultat obtenu en appliquant la transformation du cercle de Hough sur une image de référence. Nous remarquons que cette méthode nous a permis de bien localiser les étiquettes sur l'image entrante. La figure III.9 montre le résultat obtenu

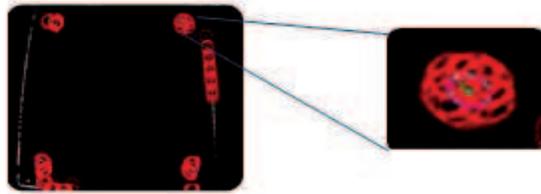


FIG. III.8 – Image de référence après l'application de la TCH

en implémentant cet algorithme sur une image de navigation du robot dans le but de le localiser au cours de son déplacement.

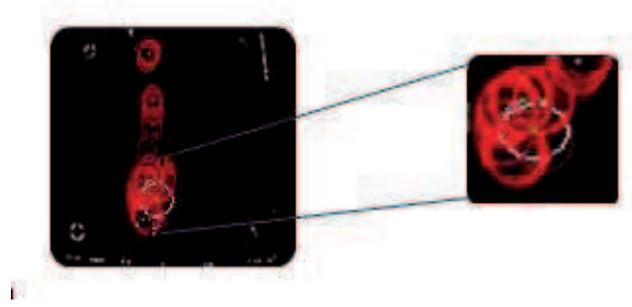


FIG. III.9 – Une image qui localise le robot après application du TCH

### III.2.3 Méthode de détermination de l'orientation du robot mobile

La méthode est basée sur la mesure de l'emplacement d'une étiquette blanche disposée sur la partie supérieure du robot. Celle-ci est collée sur l'axe des abscisses d'un repère attaché au robot comme le montre les figures III.10 et III.11.

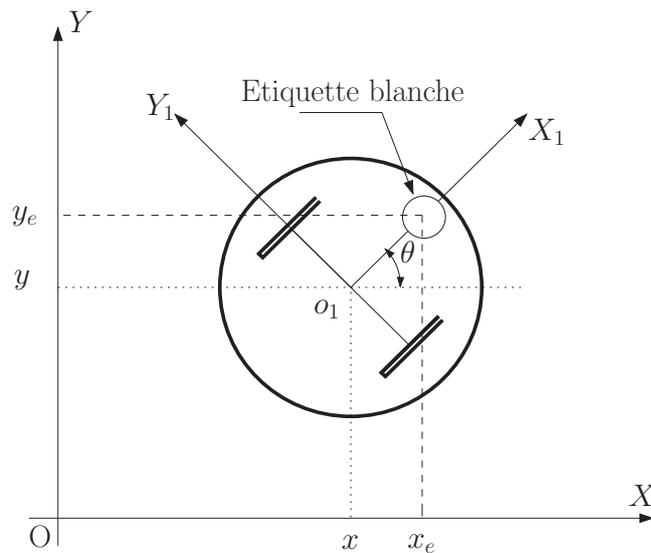


FIG. III.10 – Positionnement d'une étiquette blanche sur l'axe des abscisses

L'orientation est obtenue par l'application de l'équation suivante :

$$\theta = \arctg \frac{(y_e - y)}{(x_e - x)} \quad (\text{III.4})$$

avec  $(x, y)$  les coordonnées du robot et  $(x_e, y_e)$  les coordonnées de l'étiquette dans le repère de référence comme le montre la figure III.12.



FIG. III.11 – Image réelle du robot avec l'étiquette blanche.

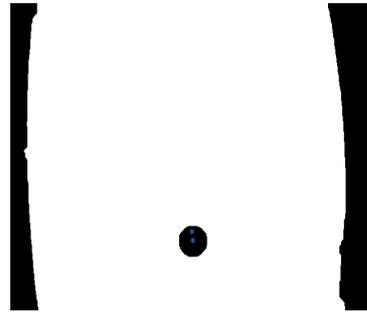


FIG. III.12 – Image binaire montrant les centres de gravité du robot et de l'étiquette.

Le système de localisation absolue par la caméra permet de déterminer ces coordonnées dans un repère lié à l'image de la caméra. Il faut trouver une méthode qui permet le passage des coordonnées images aux coordonnées réelles.

### III.2.4 Méthode de passage des coordonnées image aux coordonnées réelles

En se basant sur la connaissance des paramètres suivants :

- es positions réelles des étiquettes noires dans le repère de l'environnement et de leurs positions dans l'image de référence
- les coordonnées image des centres de gravité du robot notées  $X_R$  et  $Y_R$
- les coordonnées de l'étiquette dans le plan image

nous pouvons calculer les coordonnées réelles du robot ( $x$ ,  $y$  et  $\theta$ ) en appliquant une fonction linéaire. Cela revient à multiplier les coordonnées images par un coefficient  $K$ . Ce dernier est calculé après plusieurs tests expérimentaux sur l'image de référence tout en tenant compte de la dilatation des images.

### III.2.5 Méthode d'optimisation du temps de traitement des images

Sous Matlab et avec un PC Pentium IV de 3 *Ghz*, le temps d'acquisition de la première image est de 5 secondes, par contre le temps moyen de traitement de la suite des images est de 1,44 secondes par image. Ce temps d'acquisition est assez important, il restreint le déplacement du robot à une vitesse très faible. Pour localiser le robot en temps réel, il

faut diminuer le temps de traitement des images. La solution était de faire le traitement uniquement sur une partie de l'image contenant l'image du robot.

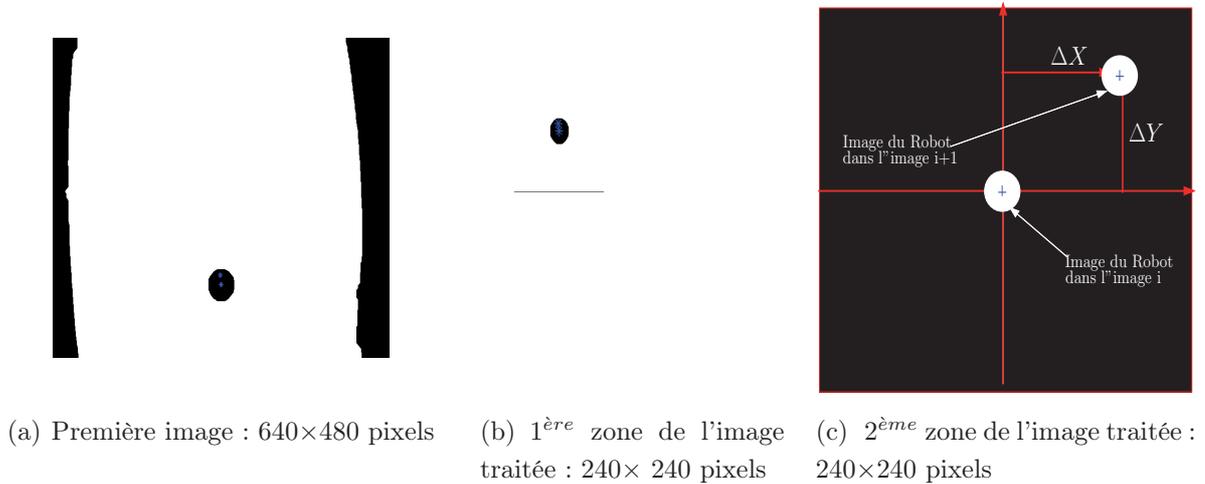


FIG. III.13 – Traitement des trois premières images

La figure III.13 contient trois images. La première image notée (a) est une image binaire de taille  $640 \times 480$  pixels correspondant à la première image acquise par la caméra. C'est à partir de cette image qu'on détermine le centre de gravité de la zone représentant le robot, qui sera utilisée dans le traitement de la suite des images. Les deux autres images notées (b) et (c) sont de taille  $240 \times 240$  pixels. A noter que pour la lisibilité de la troisième image III.13 (c), on l'a agrandie et on a enlevé le fond noir.

Il faut préciser à ce niveau que pendant l'acquisition et le traitement de la première image le robot est immobile. Ensuite, on détermine la position et l'orientation du robot en se basant sur les positions des deux centres de gravité : le premier centre de gravité est celui de la surface du robot et le second est celui de l'étiquette blanche. Ces paramètres (position et orientation du centre de gravité du robot) vont nous permettre dans la suite des images acquises de traiter seulement une zone autour de ce centre de gravité.

L'architecture de l'algorithme d'optimisation du temps de traitement des images est présentée dans la figure III.14. Pour les expériences menées et connaissant la vitesse de déplacement du robot, on a pu fixer la taille de la zone à  $240 \times 240$  pixels. Avec cette taille on est sûr de trouver le robot dans cette zone quelle que soit sa vitesse de déplacement.

Cette architecture est composée de huit étapes. Les quatre premières étapes correspondent à l'acquisition de la première image, l'estimation de la position du robot en pixels (notée  $X_R$  et  $Y_R$ ) la détermination des paramètres de la zone rectangulaire. Les quatre

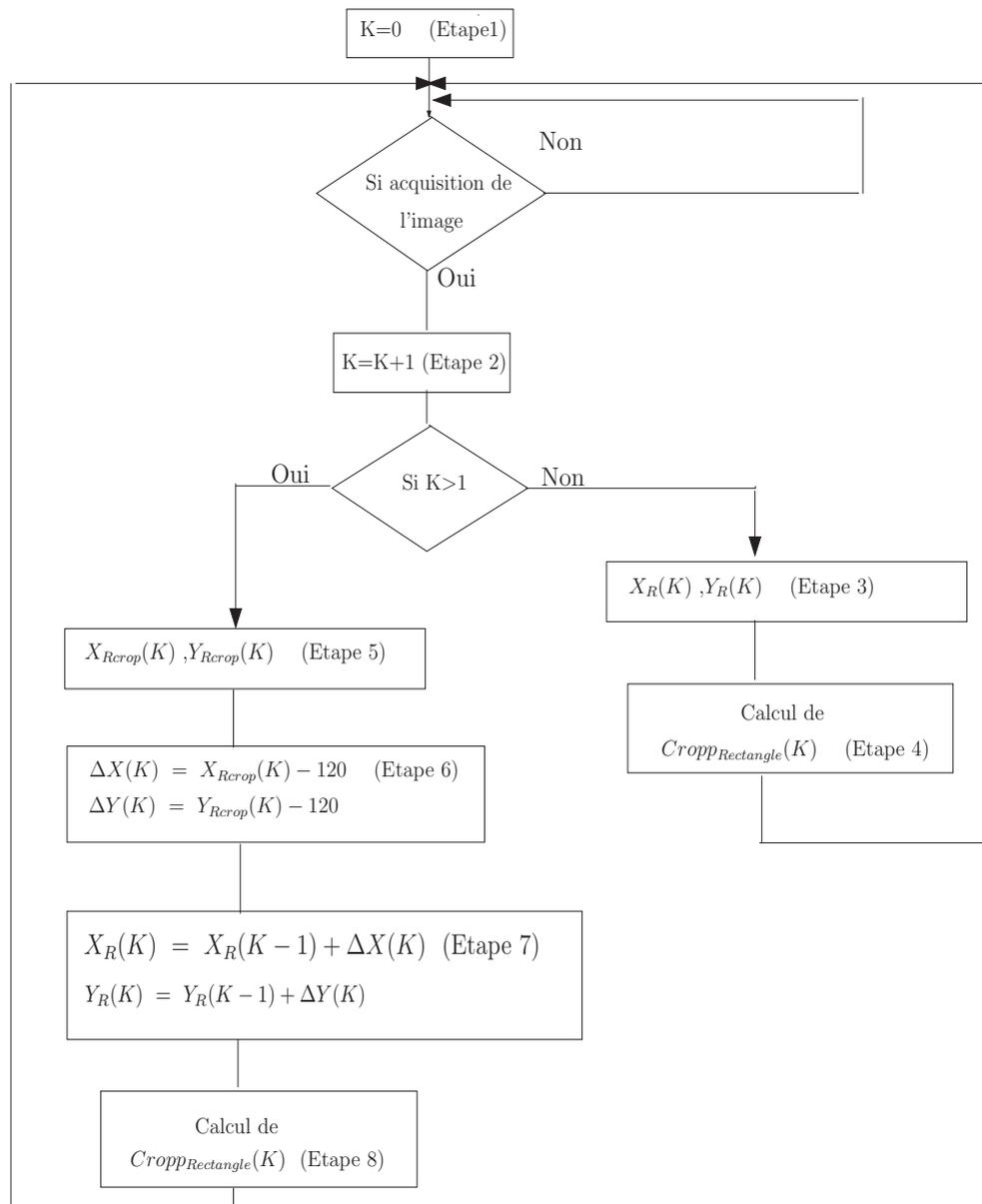


FIG. III.14 – Organigramme d’optimisation du temps de calcul pour le traitement des images

étapes suivantes sont consacrées à l’estimation de la position du robot en pixels dans la suite des images acquises. A noter que les paramètres  $X_{Rcrop}$  et  $Y_{Rcrop}$  correspondent à la position du robot dans la zone image de taille  $240 \times 240$  pixels. En appliquant cette technique de réduction de la zone d’image analysée, le temps moyen de traitement dans le processeur du PC passe de 1,44 secondes à 0,48 secondes ce qui représente un gain de 70%. La figure III.15 donne le principe général de l’approche de localisation retenue .

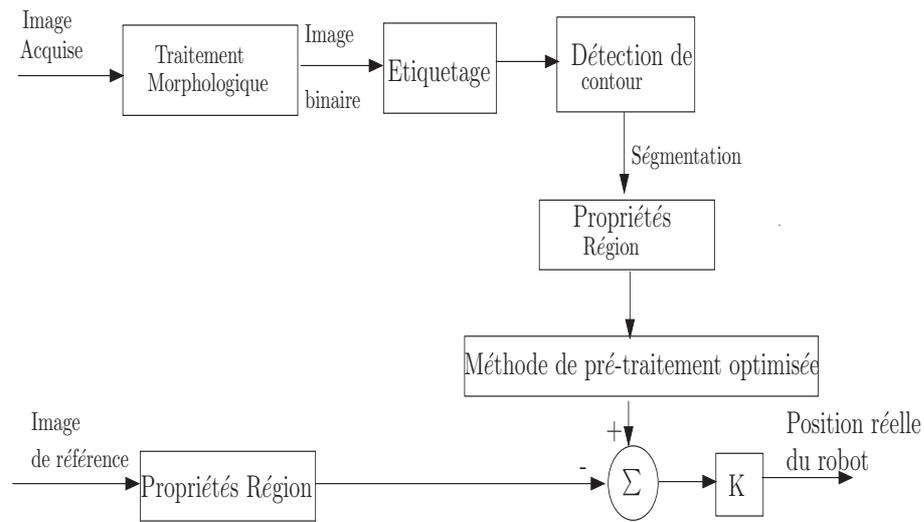


FIG. III.15 – Principe de l’algorithme de localisation absolue par la caméra

### III.3 Résultats Expérimentaux

L’environnement des expériences est illustré par la figure III.16. La webcam est connectée au PC via un port USB. Il faut signaler que nous avons utilisé la localisation par webcam pour déterminer la position de départ du robot nécessaire pour le système de mesure odométrique.



FIG. III.16 – Environnement expérimental.

### III.3.1 Première expérience

L'objectif de cette première expérience est de montrer l'intérêt de la localisation absolue par la caméra de type Webcam et de comparer les résultats obtenus par les encodeurs et la caméra. Avant l'utilisation de cette méthode, les coordonnées du point de départ  $x, y$  ainsi que l'orientation du robot sont déterminées approximativement à partir d'un repère tracé sur la maquette. La localisation du robot par la caméra permet l'automatisation de cette opération avec plus de précision. Dans cette expérience, le point de départ est déterminé par la caméra. Les figures III.17 et III.18 montrent respectivement les images originales et les images segmentées au cours de la réalisation de la trajectoire par le robot pour atteindre le point d'arrivée de coordonnées ( $x_o = 300$  mm,  $y_o = 300$  mm).

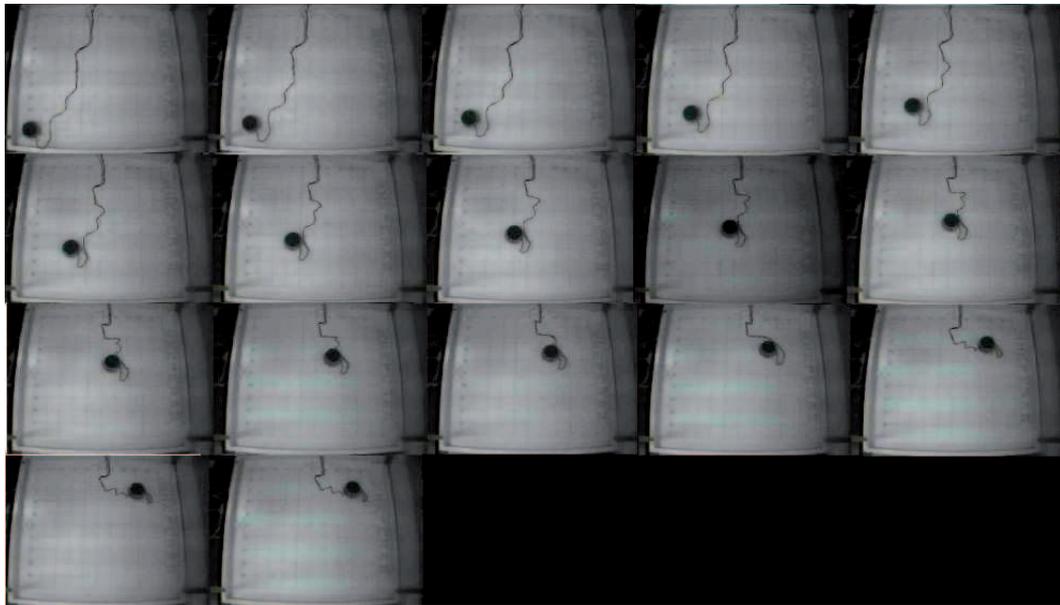


FIG. III.17 – Séquence de 17 images au cours du déplacement du robot

La figure III.19 montre une seconde trajectoire pour laquelle les coordonnées du point de départ sont ( $x = 50$  mm,  $y = 50$  mm) et les coordonnées du point d'arrivée sont ( $x_o = 50$  mm,  $y_o = 350$  mm). Dans cette figure on voit que le point d'arrivée déterminé par le système odométrique (le point noir) est différent de celui déterminé par le système à base de la caméra (le point rouge). En comparant ces deux points dans le repère tracé sur la maquette, on constate que les coordonnées du point d'arrivée fournies par le système de la caméra coïncide avec le repère de la maquette. On a noté qu'il y a une différence entre les deux points de 2,8 cm selon l'axe des  $x$  et 0,3 cm selon l'axe des  $y$ .

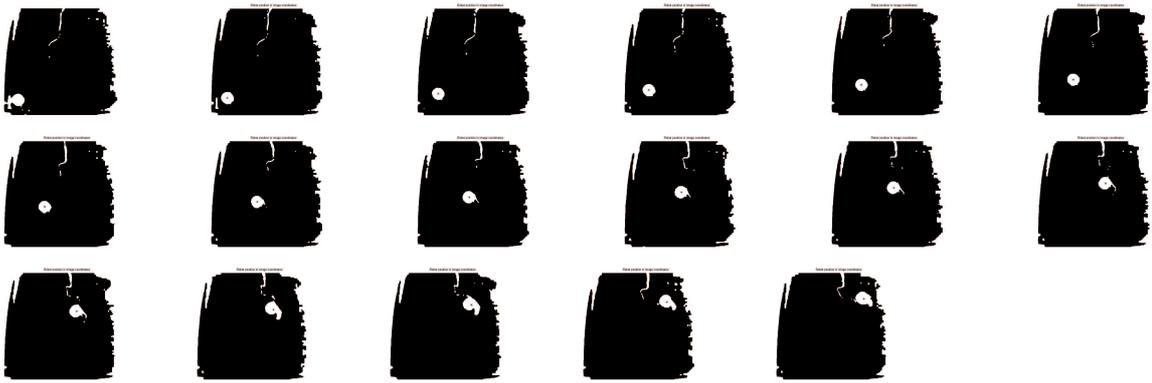


FIG. III.18 – Différentes positions images du robot au cours de son déplacement.

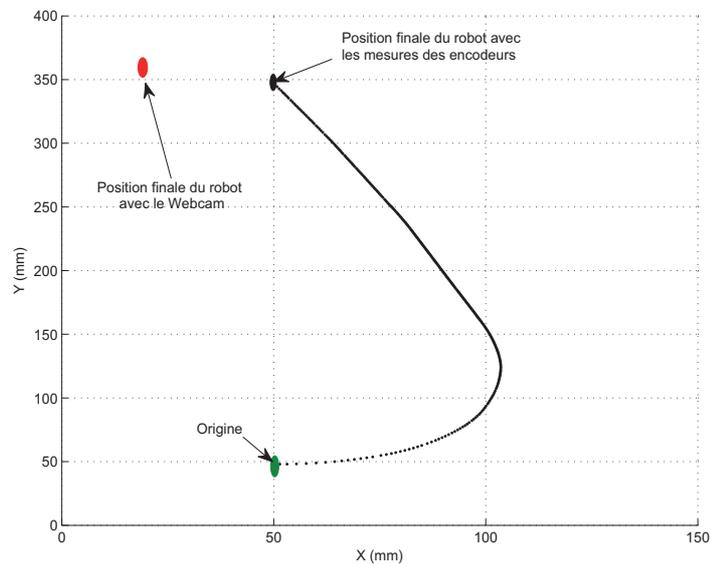


FIG. III.19 – Comparaison des coordonnées du point d'arrivée par les 2 méthodes de localisation

### III.3.2 Deuxième expérience

Le but de cette deuxième expérience est de faire la localisation simultanée du robot en temps réel par les deux méthodes. Les résultats de ces mesures sont donnés dans les figures III.20, III.22 et III.24. Dans ces expériences le robot doit rejoindre respectivement les points d'arrivés ( $x_o = 100$  mm,  $y_o = 100$  mm), ( $x_o = 400$  mm,  $y_o = 100$  mm) et ( $x_o = 300$  mm,  $y_o = 300$  mm). La figure III.20 montre que pour de faibles longueurs de trajectoires, les deux mesures de position (odométriques et par caméra) sont très proches.

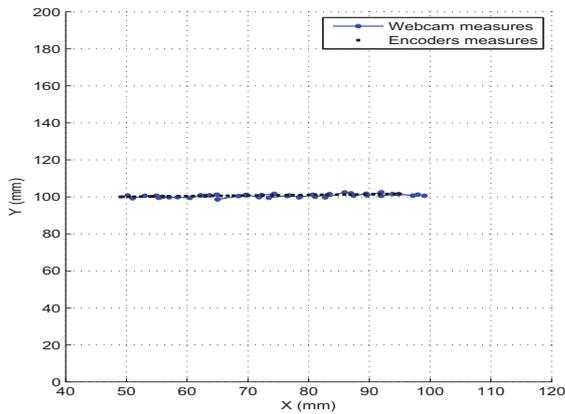


FIG. III.20 – Localisation du Robot (courte trajectoire).

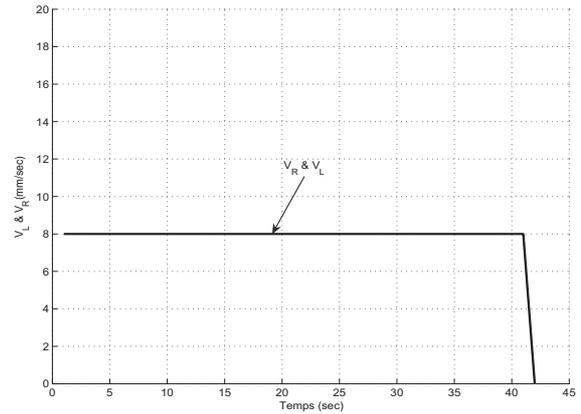


FIG. III.21 – Vitesses droite et gauche pour la trajectoire de la figure III.20.

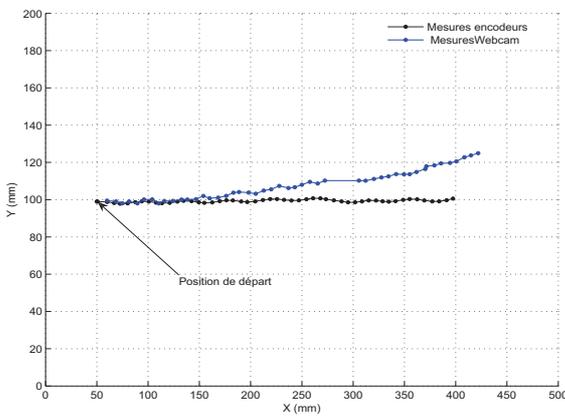


FIG. III.22 – Localisation du Robot (longue trajectoire).

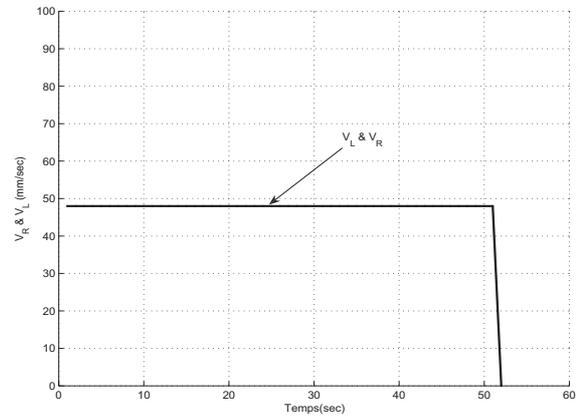


FIG. III.23 – Vitesse des roues droite et gauche.

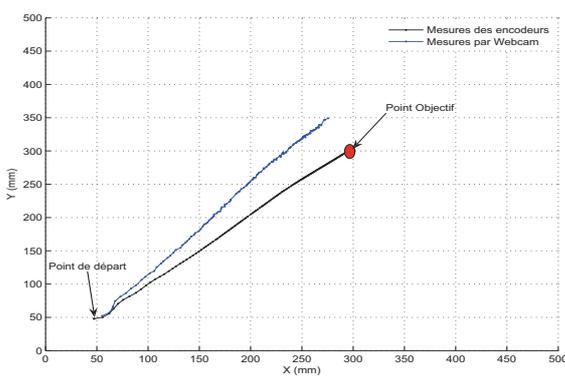


FIG. III.24 – Localisation du Robot (trajectoire rampe).

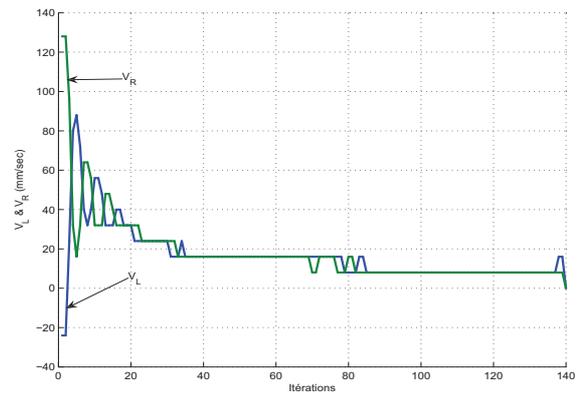


FIG. III.25 – Vitesses de commande pour la trajectoire de la figure III.24.

Par contre pour de longues trajectoires, l'erreur sur la position devient très grande. C'est le cas de l'expérience illustrée par la figure III.22. Pour quantifier les erreurs de chaque méthode de localisation, on a relevé quatre positions ( $M_1$ ,  $M_2$ ,  $M_3$  et  $M_4$ ) à l'aide d'une règle comme le montre la figure III.26. Les résultats de ces mesures sont reportés dans le tableau III.1. On a remarqué que les valeurs réelles sont très proches des mesures utilisant la webcam, bien plus que de celles obtenues par les encodeurs.

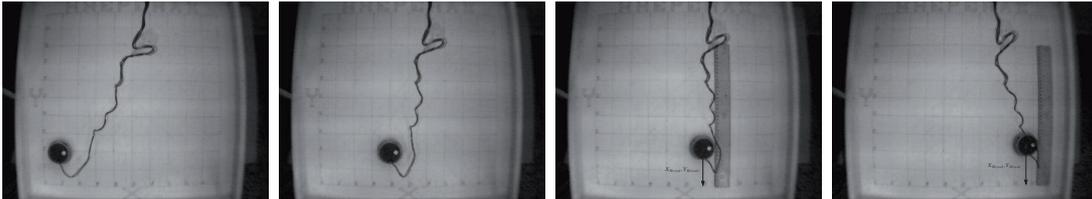


FIG. III.26 – Relevé de 4 positions du robot avec 3 méthodes de mesures .

TAB. III.1 – Evaluation des performances des méthodes de localisation

| $x(mm) / y(mm)$                |                  | <i>Mesures de la position du robot</i> |     |       |       |       |       |       |       |
|--------------------------------|------------------|--|-----|-------|-------|-------|-------|-------|-------|
|                                |                  | $M_1$                                  |     | $M_2$ |       | $M_3$ |       | $M_4$ |       |
|                                |                  | $x$                                    | $y$ | $x$   | $y$   | $x$   | $y$   | $x$   | $y$   |
| Méthodes<br>de<br>localisation | <i>Encodeurs</i> | 50                                     | 99  | 204,4 | 99,06 | 301,9 | 98,57 | 397,4 | 100,6 |
|                                | <i>Webcam</i>    | 50                                     | 99  | 198,1 | 103,8 | 305,6 | 110,2 | 421,9 | 124,9 |
|                                | Règle            | 50                                     | 100 | 200   | 104   | 306   | 115   | 424   | 125   |

### III.3.3 Troisième expérience

L'objectif de cette expérience est de montrer que les deux méthodes de localisation sont complémentaires. Pour être efficace, la méthode de localisation relative basée sur les mesures des encodeurs doit respecter ces deux conditions :

- Avoir une connaissance du point initial pour démarrer le système de localisation.
- Mettre en place un système de recalage après un parcours donné.

Ces deux éléments sont fournis par la méthode de localisation absolue basée sur la caméra. L'inconvénient de cette méthode est le temps de calcul qui va limiter son utilisation en temps réel pour des petites vitesses de déplacement du robot mobile. Par contre, elle est indispensable pour corriger la trajectoire du robot mesurée par le système odométrique. Les résultats de ces expériences sont présentés dans la figure. III.27. Dans cette expérience, le robot doit se déplacer à partir d'un point initial de coordonnées ( $x = 50$  mm,

$y = 50$  mm) pour atteindre un objectif de coordonnées ( $x_o = 300$  mm,  $y_o = 300$  mm). Cette figure présente deux courbes : une courbe bleue qui localise le robot en utilisant les données de la caméra, et une courbe noire déterminant la position avec les mesures odométriques. Nous remarquons un décalage entre les deux types de localisation, notamment quand le robot atteint la cible. Ainsi, nous avons considéré que le point d'arrivée calculé par les données caméra (rectangle bleu) présente les véritable coordonnées du robot. Ces coordonnées sont utilisées par la suite par le contrôleur flou en tant que coordonnées d'un nouveau point de départ. Ce dernier présente un point de recalage puisqu'il permet de recalibrer la position odométrique sur la position par données caméras.

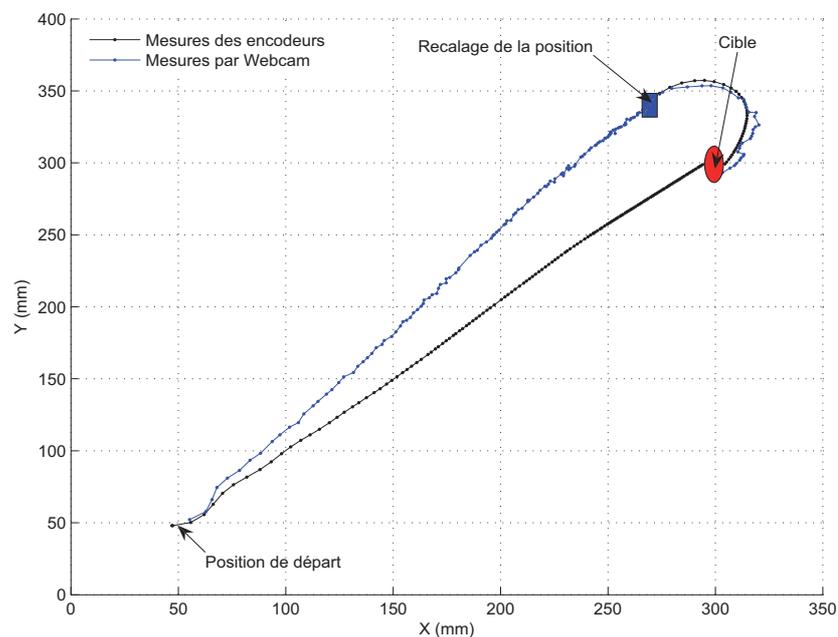


FIG. III.27 – Correction de la trajectoire du robot par le système de localisation utilisant la caméra

### III.4 Interface de commande et de supervision d'un robot mobile avec Webcam

Nous avons développé une interface graphique sous Matlab qui nous permet de commander le robot Khepera II en temps réel. Cette interface est représentée par la figure III.28. Pour utiliser cette interface il faut suivre les étapes suivantes :

- Ouvrir la communication entre le PC et le robot Khepera II par l'appuie sur le bouton correspondant de l'interface,
- Choisir le type de contrôleur flou en appuyant sur l'un des boutons (Non optimisé, Optimisé par la méthode du Gradient),
- Déterminer la position absolue de départ ( $x, y$ ) du robot et l'orientation par l'appui sur le bouton «Start Point» qui fait appel à une fonction de détermination des coordonnées par la webcam.
- Choisir un point d'arrivée en cliquant avec la souris dans la fenêtre de déplacement.
- Enfin appuyer sur le bouton «Lancer» pour démarrer l'expérience.

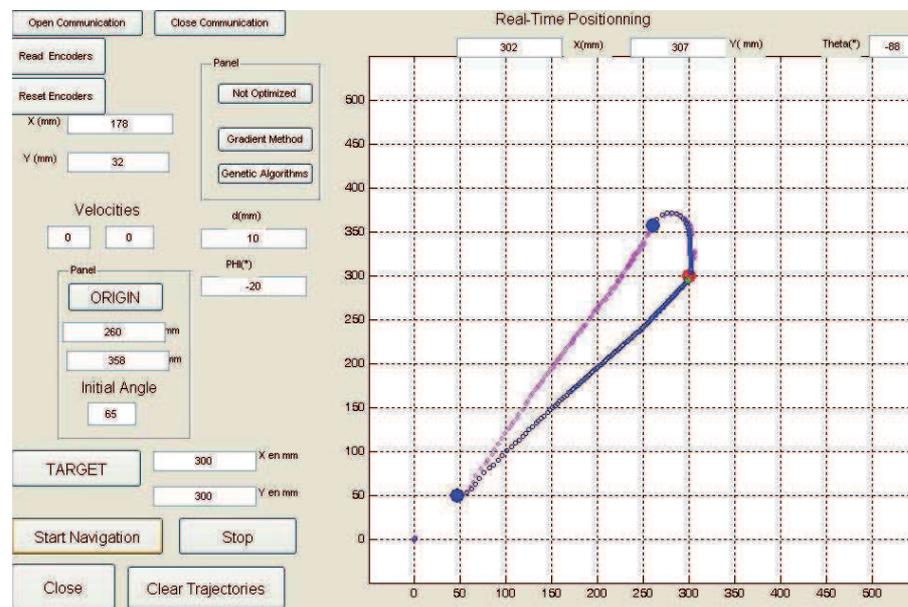


FIG. III.28 – Positions du robot estimées à partir des deux systèmes (odométrique et caméra) avec un recalage du point d'arrivée

## III.5 Conclusion

Le système de localisation relative utilisant des encodeurs est indispensable pour déterminer la position du robot mobile en temps réel. Mais, on doit avoir recours à un deuxième système qui permet de combler les lacunes du premier. Parmi les lacunes on peut rappeler, la connaissance du point initial, le cumul des erreurs qui nécessite de temps en temps un recalage... On a montré que l'utilisation d'une caméra de type webcam, qui est un dispositif bon marché comparé avec d'autres capteurs, permet de déterminer la position absolue du robot.

---

L'inconvénient de cette méthode est le temps de calcul qui est assez important, ne permettant pas de localiser le robot en temps réel. On a pu diminuer ce temps de calcul en effectuant le traitement uniquement sur la partie de l'image qui contient le robot.

La partie ainsi conçue, développée et implémentée a été restreinte à un environnement non encombré. Ainsi, et pour mieux s'approcher de l'environnement réel dans lequel un véhicule autonome doit être capable de naviguer, nous avons choisi d'étudier la partie navigation dans un environnement parsemé d'obstacles. Cette étude fera l'objet du chapitre suivant.

# Chapitre IV

## Navigation référencée multi-capteurs en environnement encombré

**Résumé** – Ce chapitre est consacré à l'étude et la conception d'un contrôleur qui assure la navigation des robots mobiles à roues dans un environnement encombré. Ce contrôleur repose sur la commutation entre la partie commande (le FLC présenté dans le chapitre II) et la partie pilotage. Cette dernière s'appuie sur l'utilisation d'une méthode d'évitement réflexe des obstacles par ZDV. La commutation entre les deux parties (commande et pilotage) a été réalisée en se basant sur un second contrôleur flou. Cependant, une organisation de l'ordre de priorité d'exécution de chacune de ces parties s'avère nécessaire afin de garantir un niveau d'autonomie important pour le robot. L'architecture choisie est de type hybride. L'approche de navigation dans un environnement encombré ainsi proposée, a été simulée et implémentée expérimentalement sur les deux robots mobiles : fauteuil roulant et ROMNI. Les résultats obtenus valident cette approche.

**Mots Clés** – Pilotage d'un Robot mobile, Evitement réflexes des obstacles, ZDV, Commutateur flou, architecture hybride (générique).

### IV.1 Introduction

A partir de l'étude bibliographique établie dans le premier chapitre, nous avons choisi d'utiliser la méthode d'évitement d'obstacles par Zone de Déformation Virtuelle (ZDV) [52]. L'objectif principal de cette stratégie est d'éviter la collision du robot avec les obstacles présents dans l'environnement. Pour cela, le robot doit disposer de capteurs extéroceptifs (détection des obstacles) et proprioceptifs (pour la commande).

Dans ce chapitre, nous allons présenter en premier lieu une synthèse de l'algorithme d'évitement d'obstacles pour le fauteuil roulant et le ROMNI. Par la suite, nous allons décrire l'architecture de contrôle adoptée. Cette dernière est de type hybride. Elle assure des actions réactives des trois parties : commande floue (chapitre II) et évitement d'obstacles (chapitre IV) et un commutateur (chapitre IV). Ce commutateur a été réalisé de deux sortes : une solution par hystérésis et une solution floue. Il assure le passage entre la partie commande et la partie pilotage. La dernière partie de ce chapitre sera consacrée à présenter les résultats de simulation et expérimentaux obtenus pour les deux robots mobiles : fauteuil roulant et ROMNI.

## IV.2 Synthèse d'un évitement d'obstacles par ZDV pour le fauteuil

La méthode d'évitement d'obstacles avec les zones de déformations virtuelles est une méthode de pilotage réactif des robots mobiles. La formulation mathématique de ce contrôleur pour le cas du fauteuil roulant est basée sur les travaux élaborés par Zapata [22] et Lapierre [23]. Le concept général de cette méthode est illustré par la figure IV.1. Cette dernière montre que la détection d'une déformation au niveau de la zone virtuelle, qui entoure le robot, engendre une modification de la commande du robot. Le calcul des nouvelles vitesses à appliquer au robot est effectué par le contrôleur par ZDV. En résultat, le robot va changer sa direction pour éviter l'obstacle détecté, ainsi une reconstruction de la zone virtuelle doit être établie afin de suivre l'évolution du robot.

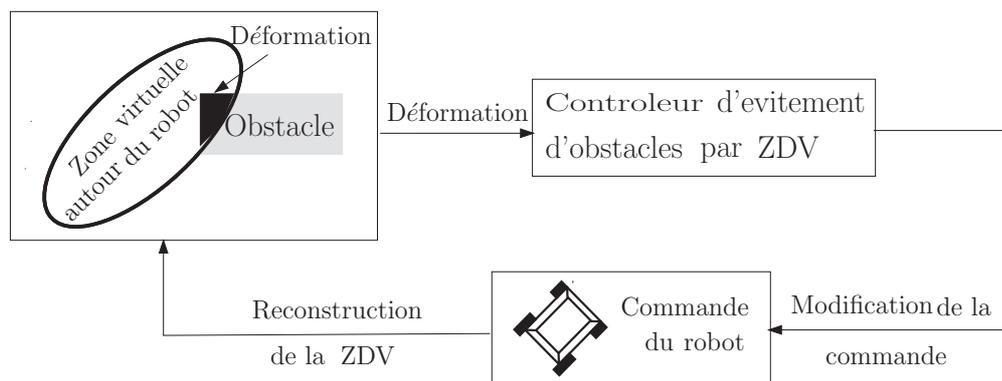


FIG. IV.1 – Principe général de la méthode ZDV

## IV.2.1 Zone de sécurité non déformée

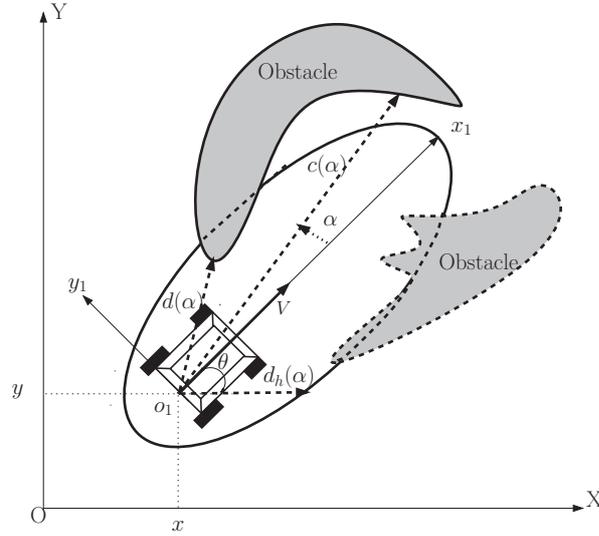


FIG. IV.2 – Evitement d'obstacles réflexes [23]

Nous avons considéré une zone de sécurité sous forme d'ellipse afin d'obtenir une expression polaire du ZDV dans le repère du robot comme le montre la figure IV.2.

L'expression de la zone de sécurité peut être représentée sous la forme suivante [23] :

$$d_h(\alpha) = \frac{-B + \sqrt{B^2 + 4AC}}{2A} \quad (\text{IV.1})$$

Cette zone de sécurité est fonction des variables de commande du robot (vitesses linéaire et angulaire). Pour cela les paramètres  $A$ ,  $B$  et  $C$  sont exprimés comme suit :

$$\begin{cases} A = (c_y \cos(\alpha - \gamma))^2 + (c_x \sin(\alpha - \gamma))^2 \\ B = 2.(a_x \cos(\alpha - \gamma)c_y^2 + (a_y \sin(\alpha - \gamma)c_x^2) \\ C = (a_x c_y)^2 + (a_y c_x)^2 - (c_x^2 c_y^2) \end{cases} \quad (\text{IV.2})$$

Les paramètres  $a_x$ ,  $a_y$ ,  $c_x$  et  $c_y$  sont décrits par le système d'équation (IV.3) et la figure IV.3 (d'après [23]) .

$$\begin{cases} c_x = \lambda_{cy} V^2 + c_x^{min} \\ c_y = \frac{\sqrt{5}}{3} c_x \\ a_x = \frac{-2}{3} c_x \\ a_y = 0 \end{cases} \quad (\text{IV.3})$$

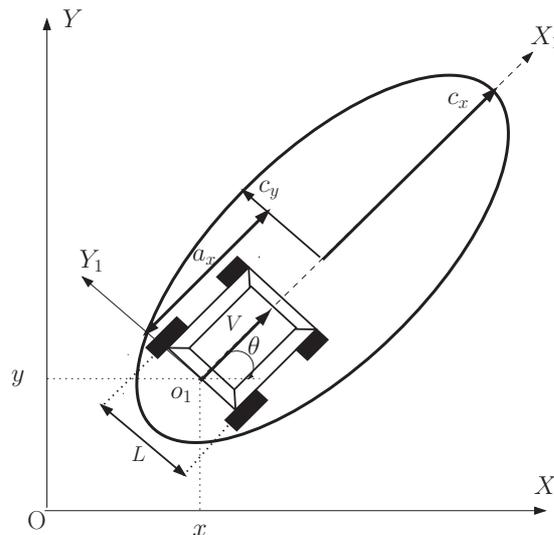


FIG. IV.3 – Zone de sécurité sous forme d'ellipse non déformée [23].

A partir du système d'équations (IV.3), nous remarquons que la zone de sécurité non déformée est exprimée en fonction des vitesses du robot. Ceci implique que lors de l'existence d'un obstacle à l'intérieur de la zone de sécurité, la vitesse du robot va diminuer et en conséquence les dimensions de l'ellipse vont être réduites.

En ce qui concerne la paramètre  $\gamma$ , nous avons supposé que la zone de sécurité est rigidement attachée au robot et qu'elle suit la direction (l'azimut) du robot, ainsi nous avons considéré initialement un angle  $\gamma = 0$ . Ensuite cet angle sera équivalent à l'orientation du robot notée  $\theta$ .

### IV.2.2 Zone de sécurité déformée

La zone de sécurité déformée est obtenue à partir des informations provenant du capteur. Ces informations qui représentent les distances mesurées par le laser sont notées  $c(\alpha)$ . Vu que les mesures utiles pour le pilotage sont celles internes à la zone de sécurité non déformée, un test préliminaire sur les mesures du capteur laser est nécessaire [23].

$$\begin{aligned} d(\alpha) &= c(\alpha) \text{ si } c(\alpha) < d_h(\alpha) \\ &= d_h(\alpha) \text{ ailleurs} \end{aligned} \tag{IV.4}$$

La figure IV.4 montre un test à vide sur le fauteuil roulant équipé du capteur laser. Durant 5 minutes, le capteur détecte des obstacles en faisant des balayages successifs qui donnent 24 mesures/balayage. Pour chaque balayage, le contrôleur d'évitement d'obstacles génère

des vitesses que nous envoyons au fauteuil. Les points en noir présentent les obstacles détectés durant 5 minutes de test à vide. Le résultat montre que les obstacles se situent à gauche du robot.

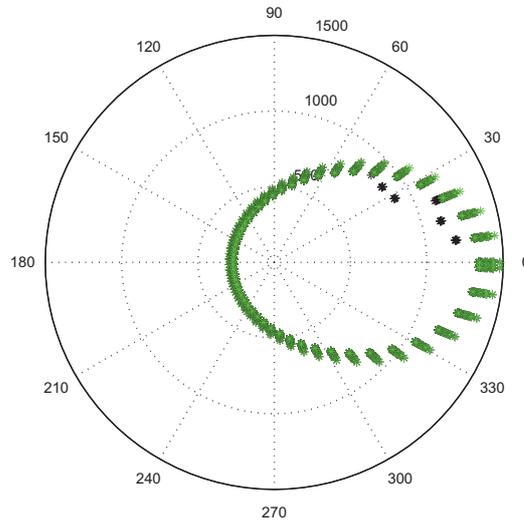


FIG. IV.4 – Zone de sécurité sous forme d'ellipse déformée.

### IV.2.3 Déformation

Le choix d'une expression variable pour l'information d'intrusion est très important dans la conception de la réaction du système de pilotage. L'intrusion n'est pas simplement la position d'un obstacle dans la zone de sécurité (l'ellipse), mais elle présente la somme des distances entre le robot et l'obstacle quand celui-ci se trouve à l'intérieur de l'ellipse. Cette somme présente la déformation de la zone de sécurité. Considérons par  $I$  l'expression de l'information d'intrusion. Nous remarquons que pour une distance robot/obstacle nulle, l'intrusion tend vers l'infini. Ainsi un contrôle qui assure une intrusion bornée permettra au système d'éviter tous les obstacles [23].

$$I = \int_{\alpha=0}^{\pi} \frac{d_h(\alpha, V) - d(\alpha)}{d(\alpha)} d\alpha \quad (\text{IV.5})$$

### IV.2.4 Fonctions Jacobiennes

La dérivation par rapport au temps de l'expression de l'intrusion aboutit au résultat suivant [23].

$$\dot{I} = J_1^V \dot{V} + J_1^\theta \omega + F^{\text{Rob.vel}} V \quad (\text{IV.6})$$

avec :  $\omega$  la vitesse de rotation du robot

$$\begin{aligned}
J_I^V &= \int_{\alpha=0}^{\pi} \frac{1}{d(\alpha)} \left( \frac{1}{2A} \left[ -J_B^V + \frac{B J_B^V - 2C J_A^V - 2A J_C^V}{\sqrt{B^2 - 4AC}} \right] - \frac{-B + \sqrt{B^2 - 4AC}}{2A^2} J_A^V \right) d\alpha \\
J_I^\theta &= \int_{\alpha=0}^{\pi} \frac{-1}{d(\alpha)} \left( \frac{1}{2A} \left[ -J_B^\gamma + \frac{B J_B^\gamma - 2C J_A^\gamma}{\sqrt{B^2 - 4AC}} \right] - \frac{-B + \sqrt{B^2 - 4AC}}{2A^2} J_A^\gamma \right) d\alpha \\
F^{\text{Rob.vel}} &= \int_{\alpha=0}^{\pi} \frac{dh(\alpha)}{d^2(\alpha)} \cos(\alpha) d\alpha
\end{aligned} \tag{IV.7}$$

où les jacobiennes de A, B et C sont définies comme suit :

$$J_A^V = 2(c_y \frac{\partial c_y}{\partial V} \cos^2(\alpha) + c_x \frac{\partial c_x}{\partial V} \sin^2(\alpha)) \tag{IV.8}$$

$$J_A^\gamma = 2 \cos(\alpha) \sin(\alpha) (c_y^2 - c_x^2)$$

$$J_B^V = 2(\cos(\alpha) \times (c_y^2 \frac{\partial a_x}{\partial V} + 2 a_x c_y \frac{\partial c_y}{\partial V}) + \sin(\alpha) \times (c_x^2 \frac{\partial a_y}{\partial V} + 2 a_y c_x \frac{\partial c_x}{\partial V})) \tag{IV.9}$$

$$J_B^\gamma = 2(a_x c_y^2 \sin(\alpha) - a_y c_x^2 \cos(\alpha))$$

$$\begin{aligned}
J_C^V &= 2a_x c_y \times (c_y \frac{\partial a_x}{\partial V} + a_x \frac{\partial c_y}{\partial V}) + 2a_y c_x \times (c_x \frac{\partial a_y}{\partial V} + a_y \frac{\partial c_x}{\partial V}) \\
&\quad - 2c_x c_y \times (c_x \frac{\partial c_y}{\partial V} + c_y \frac{\partial c_x}{\partial V})
\end{aligned} \tag{IV.10}$$

Pour concevoir les consignes de vitesses à appliquer sur les roues du robot mobile, nous devons utiliser le modèle cinématique inverse décrit dans le chapitre I. Nous choisissons par  $V_I = \frac{I^2}{2}$  une équation qui satisfait le théorème de Lyapunov sur la stabilité. La dérivée de cette expression par rapport à l'intrusion donne le résultat suivant [23] :

$$\dot{V}_I = I(J_I^V \dot{V} + J_I^\theta \omega + F^{\text{Rob.vel}} V) \tag{IV.11}$$

Ainsi, nous sommes arrivés à déduire le système de pilotage du fauteuil qui est décrit par le système d'équations (IV.12) [23] (avec  $K_V$  et  $K_\omega$  sont des gains positifs qui donnent  $\dot{V}_I \leq 0 \forall t$ ).

$$\begin{cases} \dot{V} = -K_V J_I^V I - \frac{F^{\text{Rob.vel}}}{J_I^V} V \\ \omega = -K_\omega J_I^\theta I \end{cases} \tag{IV.12}$$

Après avoir élaboré le contrôleur d'évitement d'obstacles pour le fauteuil, notre contribution consiste à étendre cette étude pour le cas du ROMNI.

## IV.3 Synthèse d'un évitement d'obstacles par ZDV pour le ROMNI

Dans cette partie nous allons présenter les principales modifications à apporter sur le contrôleur par ZDV présenté dans la section précédente pour le cas du ROMNI. Les modifications se situent essentiellement au niveau de :

- la forme de la zone de sécurité
- le système d'équations décrivant cette zone de sécurité
- le système de pilotage final

En effet, ROMNI est un robot à trois degrés de mobilités, ce qui implique des formulations différentes de la ZDV et de l'intrusion.

### IV.3.1 Zone de sécurité circulaire du ROMNI

La figure IV.5 illustre le principe de la méthode ZVD appliquée pour le cas du robot ROMNI et dans laquelle nous avons utilisé une zone de sécurité de forme circulaire. Le choix de cette forme est justifié par les caractéristiques cinématiques du ROMNI qui font que pour n'importe quelle situation par rapport à un obstacle (situation coin, suivi de mur, couloir ou porte), il est capable de faire des déplacements latéraux dans toutes les directions, quelle que soit l'orientation. Il s'agit d'un robot avec trois degrés de mobilité, alors que les robots de type voiture n'ont qu'un degré de mobilité et les robots de type unicycle ont deux degrés de mobilité.

Ainsi, ce robot n'a pas besoin d'une zone de sécurité large au devant (pour faire les manœuvres comme pour le cas du fauteuil) et une autre moins réduite sur les côtés mais par contre toutes les situations de déformations (devant, côté droite ou côté gauche) ont le même poids vis à vis de la réaction du système en lui-même.

L'expression de la zone de sécurité de forme circulaire peut être représentée sous la forme mathématique suivante :

$$d_h(\alpha) = \frac{-B + \sqrt{B^2 + 4AC}}{2A} \quad (\text{IV.13})$$

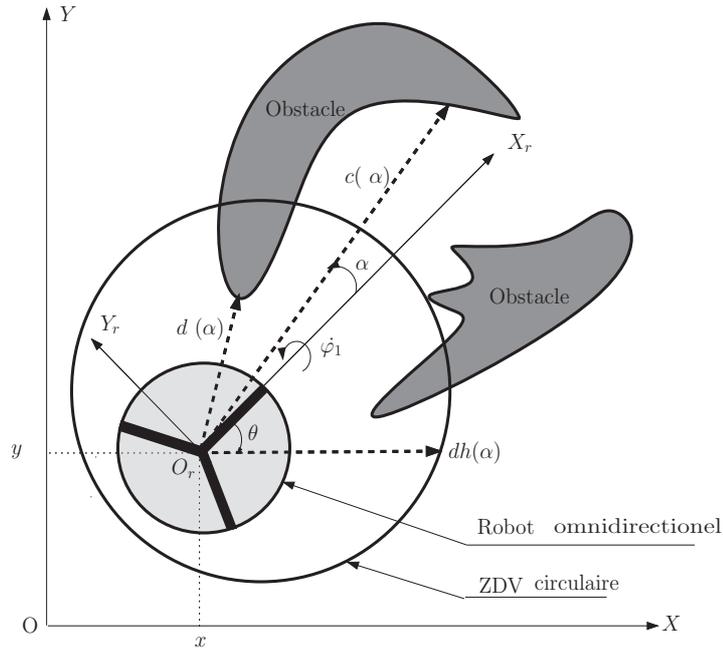


FIG. IV.5 – Illustration du principe de la ZDV appliquée pour le ROMNI.

avec :

$$\left\{ \begin{array}{l}
 A = (c_{y1} \cos(\alpha - \gamma))^2 + (c_{x1} \sin(\alpha - \gamma))^2 + (c_{y2} \cos(\alpha - \gamma))^2 \\
 \quad + (c_{x2} \sin(\alpha - \gamma))^2 + (c_{y3} \cos(\alpha - \gamma))^2 + (c_{x3} \sin(\alpha - \gamma))^2 \\
 B = 2.(a_{x1} \cos(\alpha - \gamma)c_{y1}^2 + (a_{y1} \sin(\alpha - \gamma)c_{x1}^2) + 2.(a_{x2} \cos(\alpha - \gamma)c_{y2}^2 \\
 \quad + (a_{y2} \sin(\alpha - \gamma)c_{x2}^2) + 2.(a_{x3} \cos(\alpha - \gamma)c_{y3}^2 + (a_{y3} \sin(\alpha - \gamma)c_{x3}^2) \\
 C = (a_{x1}c_{y1})^2 + (a_{y1}c_{x1})^2 - (c_{x1}^2c_{y1}^2) + (a_{x2}c_{y2})^2 + (a_{y2}c_{x2})^2 - (c_{x2}^2c_{y2}^2) + \\
 \quad (a_{x3}c_{y3})^2 + (a_{y3}c_{x3})^2 - (c_{x3}^2c_{y3}^2)
 \end{array} \right. \quad (IV.14)$$

Les paramètres  $a_{xi}$ ,  $a_{yi}$ ,  $c_{xi}$  et  $c_{yi}$  ( $i = 1..3$ ) représentent la longueur et la largeur de la zone de sécurité comme le montre la figure IV.6.

La formulation mathématique de ces paramètres doit aussi tenir compte des vitesses des trois essieux du robot ROMNI, pour cela nous avons choisi les fonctions de pilotage décrites par le système d'équations (IV.15). En effet, ces équations permettent de positionner le robot au centre de la zone de sécurité qui l'entoure.

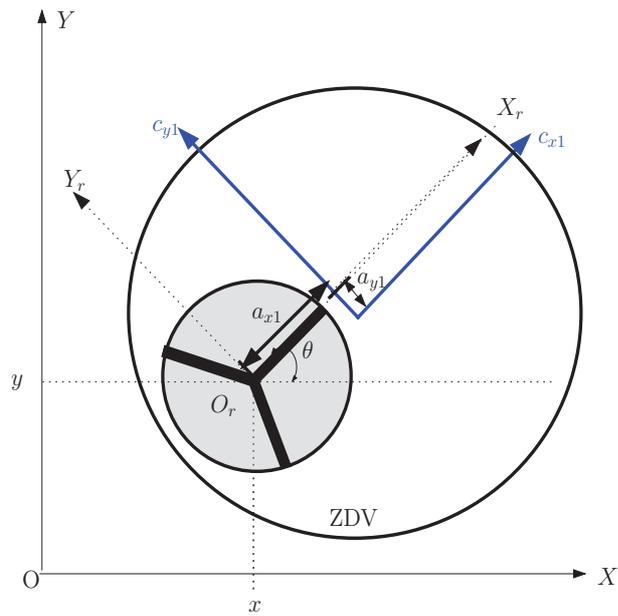


FIG. IV.6 – Zone de sécurité circulaire non déformée du ROMNI.

$$\left\{ \begin{array}{l} c_{x1} = \lambda_{cy1} \times \varphi_1^2 + c_{x1}^{min} \\ c_{x2} = \lambda_{cy2} \times \varphi_2^2 + c_{x2}^{min} \\ c_{x3} = \lambda_{cy3} \times \varphi_3^2 + c_{x3}^{min} \\ c_{y1} = c_{x1} \\ c_{y2} = c_{x2} \\ c_{y3} = c_{x3} \\ a_{x1} = a_{x2} = a_{x3} = 0 \\ a_{y1} = a_{y2} = a_{y3} = 0 \end{array} \right. \quad (IV.15)$$

Nous remarquons que les variables  $c_{x_i}$  et  $c_{y_i}$  avec  $i = 1..3$  sont fonctions des vitesses des trois essieux  $\varphi_1$ ,  $\varphi_2$  et  $\varphi_3$ . Ainsi les dimensions de la ZDV (longueur et largeur) varient en fonction du changement de la commande du robot. Ceci implique une reconstruction de la zone virtuelle à chaque modification de la commande du robot.

### IV.3.2 Fonctions Jacobiennes

Ensuite, nous établissons la jacobienne afin d'aboutir à une représentation des trois vitesses du ROMNI en fonction de la déformation. La représentation obtenue est décrite par le système d'équations (IV.16).

$$\begin{cases} -K_{\varphi_1} J_1^{\varphi_1} \mathbf{I} - \frac{F^{\text{Rob.vel}}}{J_1^{\varphi_1}} (-\dot{x} \sin(\theta + \alpha_1) + \dot{y} \cos(\theta + \alpha_1)) + (K_{\omega} J_1^{\theta} \mathbf{I}) l_1 = -r\dot{\varphi}_1 \\ -K_{\varphi_2} J_1^{\varphi_2} \mathbf{I} - \frac{F^{\text{Rob.vel}}}{J_1^{\varphi_2}} (-\dot{x} \sin(\theta + \alpha_2) + \dot{y} \cos(\theta + \alpha_2)) + (K_{\omega} J_1^{\theta} \mathbf{I}) l_2 = -r\dot{\varphi}_2 \\ -K_{\varphi_3} J_1^{\varphi_3} \mathbf{I} - \frac{F^{\text{Rob.vel}}}{J_1^{\varphi_3}} (-\dot{x} \sin(\theta + \alpha_3) + \dot{y} \cos(\theta + \alpha_3)) + (K_{\omega} J_1^{\theta} \mathbf{I}) l_3 = -r\dot{\varphi}_3 \end{cases} \quad (\text{IV.16})$$

avec  $K_{\varphi_i}$  ( $i=1..3$ ) et  $K_{\omega}$  sont des gains positifs qui donnent  $\dot{V}_1 \leq 0 \forall t$ .

Nous pouvons (maintenant) synthétiser l'algorithme d'évitement d'obstacles pour les deux robots mobiles : fauteuil et ROMNI. L'intégration de cet algorithme sur chacune des deux plateformes doit tenir compte évidemment de la présence de l'algorithme de commande floue. Ceci implique la nécessité d'utilisation d'une architecture de contrôle permettant de choisir instantanément un comportement à activer : déplacement vers le but ou évitement d'obstacles. La synthèse de cette architecture sera établie dans le paragraphe suivant.

## IV.4 Synthèse de l'architecture de contrôle

Durant les dernières années, beaucoup d'intérêts ont été orientés à asservir le système de contrôle à l'utilisateur en adoptant différentes architectures. Principalement, cette recherche nous a dérivé du champ de la robotique mobile, au champ des fauteuils roulants alimentés électriquement et jouant un rôle vital à réduire le problème d'indépendance souffert par les personnes handicapées. Cette recherche a montré que traditionnellement, les fauteuils roulants sont basés sur une solution intuitive (joystick). Cependant, afin de pouvoir les conduire efficacement et sûrement, l'utilisateur doit utiliser des méthodes alternatives d'interaction.

Plusieurs travaux de recherche qui traitent ces méthodes alternatives d'interaction ont été développés. A titre d'exemple, dans [53] et [54] les chercheurs proposent des méthodes qui assurent la conduite des fauteuils en utilisant les champs de visage ou la reconnaissance des gestes de la main sur le joystick. Ensuite, les chercheurs se sont investis dans de nouveaux champs afin de compenser les faibles réactions du joystick. Parmi ces travaux on peut citer ceux développés par Taha *et al.* [55] Millan *et al.* [56] offrant un

degré d'autonomie assez important au robot. Cette autonomie a été assurée à travers l'interface cerveau de l'utilisateur lui offrant une faible résolution de l'entrée. Il existe aussi des systèmes de contrôle mixte qui assurent la commutation entre différents modes d'opérations comme le système « Navchair » proposé par [57] ou celui de Carlson [58] qui utilisait un contrôle collaboratif permettant d'augmenter la sécurité du fauteuil robotique en se basant sur des expériences de mobilité référencée tâche secondaire : ils gardent le contrôle amorcé par l'utilisateur et adaptent les signaux où c'est nécessaire.

Dans ce présent travail, nous avons proposé une méthodologie de contrôle générique, qui permet de déduire les intentions de l'utilisateur à partir de l'entrée appliquée au joystick, et ceci en supposant inconnu l'environnement local qui entoure le fauteuil roulant [59, 60, 61]. Nous avons suivi les recommandations de Carlson [62] et ceci dans le but d'exécuter des manœuvres précises et particulières, ou bien pour éviter les collisions. Notre plateforme robotique utilise respectivement les trois modes d'opération suivants :

- Déplacement vers un point objectif et/ou suivi de chemin : navigateur.
- Evitement d'obstacles en utilisant un capteur laser : un pilote de réaction.
- Commutation entre les deux objectifs en utilisant un contrôleur flou

#### IV.4.1 Architecture du système de contrôle

Nous utilisons l'architecture générique de Mourioux [28] qui nous permet de structurer la commande en plusieurs niveaux. L'architecture proposée est composée de deux parties appelées la partie de décision et la partie de perception. Ces deux parties du même niveau sont directement connectées : dans chaque niveau des modules sont définis, chaque module correspond à une fonctionnalité, qui permet une coupe fonctionnelle. L'architecture de contrôle mise en oeuvre sur un fauteuil roulant robotisé est décrite par la figure IV.7 :

- Dans le niveau 0 nous trouvons le Système Mécanique Articulé (AMS) qui peut être contrôlé par ses deux moteurs  $moteur_d$  et  $moteur_g$  qui désignent respectivement les moteurs droit et gauche.
- $\omega_d$  et  $\omega_g$  sont respectivement les modules pour la commande des roues motorisées droites et gauches. Ce module utilise les mesures provenant des encodeurs  $Encodeur_d$  et  $Encodeur_g$ .
- *Télémètre Laser* est le module qui contrôle le télémètre laser. Il délivre des informations pour le niveau supérieur qui est une liste de distances polaires dont chacune dépend de l'angle de mesure.
- Le module *Carte de Proximité* est le modèle des obstacles détectés. Il contient toutes les distances entre les obstacles et le robot.

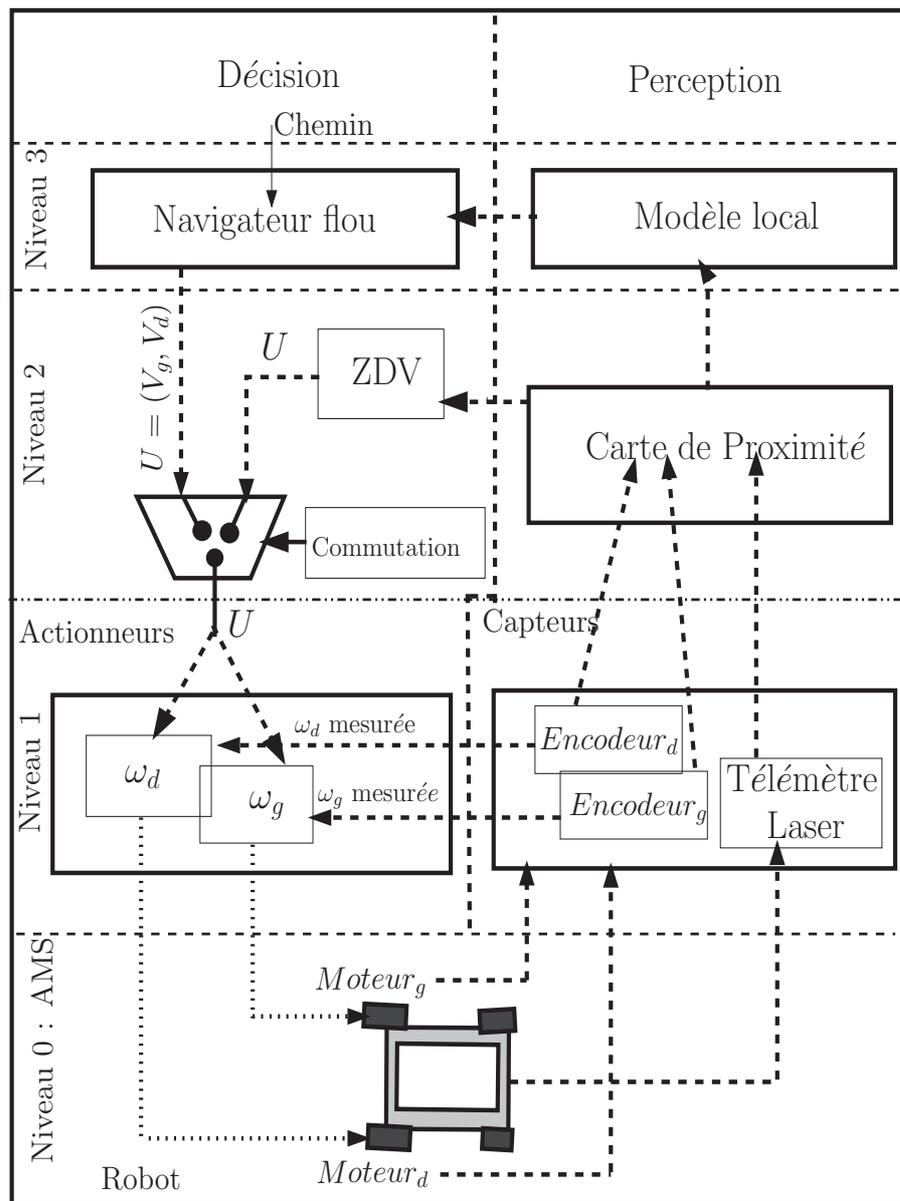


FIG. IV.7 – L'architecture de contrôle générique du fauteuil roulant

- Le module *DVZ* fournit au robot un comportement réflexe pour éviter des obstacles en tenant compte des informations de proximité fournies par le module *Proximity Map* une zone de sécurité qui assure une navigation sans risque.
- *Commutation* assure la commutation entre deux fonctionnalités : la commande et l'évitement d'obstacles pour que le robot exécute la trajectoire désirée en évitant des obstacles.
- Le module *Navigateur Flou* calcule les vitesses angulaires du robot dans la carte

locale en utilisant le concept de la logique floue sans tenir compte des obstacles de l'environnement.

Avec cette architecture générique, le fauteuil roulant robotisé peut assurer une navigation autonome et sans risque même si l'environnement qui l'entoure est inconnu. Il est ainsi apte à éviter les obstacles (en utilisant le pilote ZDV) et à atteindre un point cible (le navigateur flou) ou même suivre une trajectoire désirée.

#### IV.4.2 Le problème de commutation : solution hystérésis

L'intégration des deux algorithmes doit concilier a priori deux exigences antagonistes. La première exigence est qu'en présence d'informations de déformations importantes le pilote réactif doit guider le fauteuil roulant robotisé loin de l'obstacle. La deuxième exigence consiste à la recherche de la cible (l'objectif) désirée en utilisant le contrôleur flou et ceci bien sûr en absence d'obstacles.

La solution pratique à considérer dans cette commutation correspond à appliquer le principe d'hystérésis comme illustré dans la figure IV.8.

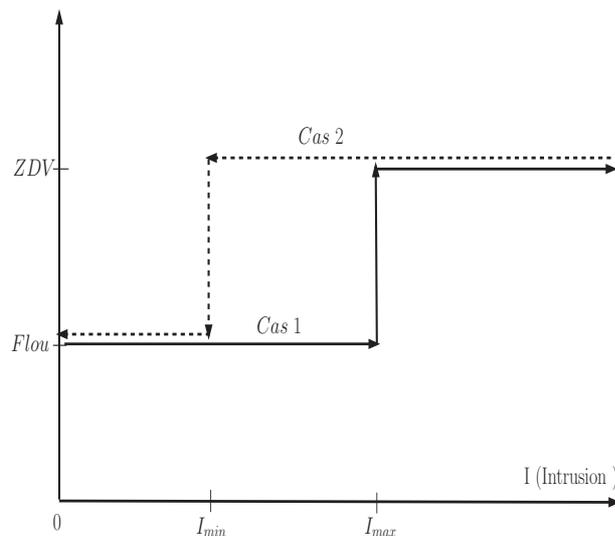


FIG. IV.8 – Diagramme par Hystérésis appliqué au robot

Dans le premier cas noté *Cas 1*, le robot vise de rejoindre une cible (un objectif) tandis que la distance à un obstacle présent dans l'environnement diminue, autrement dit les augmentations d'informations d'intrusion de 0 à  $I_{max}$  qui est la déformation maximale permise pour commencer l'évitement des obstacles et ceci en tenant compte des capacités

du robot. En conséquence le contrôleur commute du navigateur flou au pilote ZDV et persiste dans cette situation tant que  $I > I_{max}$ .

Dans le deuxième cas, le robot est en train d'éviter un obstacle et en conséquence les informations de déformation diminuent au cours du temps. Dans ce cas, le contrôle de commutation permet le passage au contrôleur flou seulement quand  $I_{min} \leq I < I_{max}$  et ceci dans le but d'éliminer la transition entre les deux contrôleurs ou encore appelé la zone de recouvrement.

Nous avons réalisé plusieurs tests afin de juger l'efficacité de cette méthode de commutation. La figure IV.9 présente un déplacement réel du fauteuil roulant à partir d'un point initial ( $x = 0$  mm,  $y = 0$  mm et  $\theta = 0$ ). Le point objectif à atteindre est défini par les coordonnées ( $x_o = 4\ 000$  mm,  $y_o = 3\ 500$  mm). La courbe en rouge présente la trajectoire réelle réalisée par le robot et les points bleus sont les obstacles inconnus détectés par le capteur laser. Cependant, la figure présente des points bleus qui sont très proches de la trajectoire, ces points sont des codes d'erreur.

Dans cet exemple, le robot est contraint à un obstacle de forme circulaire juste au début de son déplacement. Nous remarquons qu'il arrive à éviter l'obstacle et atteint l'objectif désiré et le dépasse. Ce problème est dû à la localisation relative qui présente des erreurs cumulatives. Cependant, cette solution pratique reste insuffisante essentiellement lorsque

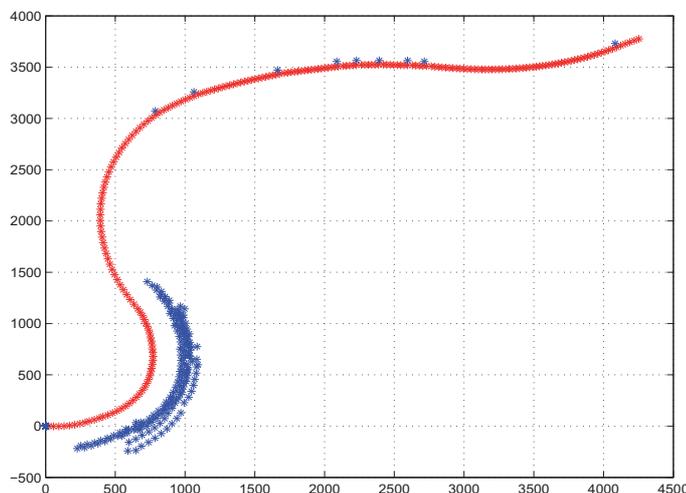


FIG. IV.9 – Trajectoire avec un changement de la direction du robot et des obstacles dynamiques

le robot est contraint à une situation de coin. Pour remédier à ce problème, nous avons proposé de développer un second contrôleur flou qui jouera le rôle de commutateur.

### IV.4.3 Problème de commutation : solution floue

Dans ce commutateur basé sur le concept flou, nous avons introduit la notion de mini-intrusion sans risque. Celles-ci sont des informations d'intrusion fournies par le pilote ZDV. Elles permettent de traiter en ligne les situations de minima locaux. En effet, ce problème apparaît lorsque les deux critères : commande floue et évitement d'obstacles engendrent deux réactions antagonistes. Cette situation est engendrée par une symétrie entre les déformations droites et gauches de la zone virtuelle, et ceci en respectant la direction de la vitesse linéaire. En conséquence, la vitesse générée par le contrôleur par ZDV sera réduite à zéro ou à une valeur négative. Le robot va s'orienter de manière à minimiser l'intrusion. Cependant, les vitesses négatives appliquées ne lui permettent pas de s'en sortir.

La solution proposée consiste à réduire la vitesse du robot et tourner jusqu'à ce que l'obstacle soit présent dans un seul côté. Pour cela, nous avons considéré quatre mini-intrusion comme le montre la figure IV.10. Les zones de variations de ces mini-intrusion se présentent comme suit :

- une mini-intrusion devant-gauche  $I_g$  calculée pour  $\theta \leq \alpha < \theta + 60^\circ$
- une mini-intrusion devant-droite  $I_d$  calculée pour  $\theta - 60^\circ \leq \alpha < \theta$
- une mini-intrusion de-côté-gauche  $I_{gc}$  calculée pour  $\theta + 60^\circ \leq \alpha < \theta + \pi$
- une mini-intrusion de-côté-droite  $I_{dc}$  calculée pour  $\theta + \pi \leq \alpha < \theta - 60^\circ$

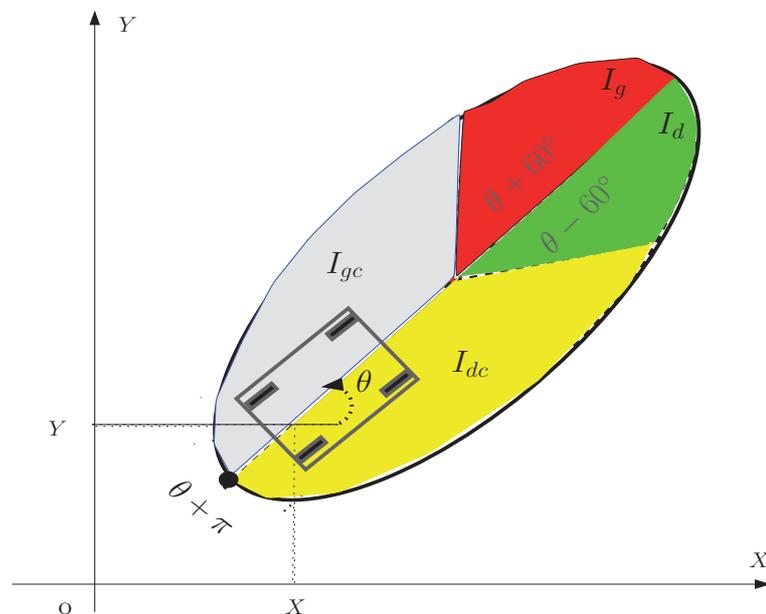


FIG. IV.10 – Cette figure montre les quatre mini-zone de sécurité :  $I_g$ ,  $I_d$ ,  $I_{gc}$  et  $I_{dc}$ . Les règles sont conçues pour des scénarios multiples afin d'aider l'utilisateur

L'idée de base de ce commutateur est le produit de deux parties. Dans la première, les obstacles sont présents devant le fauteuil roulant et dans la direction du point objectif ( $I_g$  ou  $I_d$ ) ou tous les deux), ainsi il doit les éviter obligatoirement. Dans la deuxième partie, les obstacles sont présents dans l'un ou les deux côtés droit et gauche et bien sûr ces obstacles ne sont pas dans la direction du point objectif ( $I_{gc}$  ou  $I_{dc}$ ). Ainsi le robot peut essayer d'atteindre son objectif sans éviter ces obstacles puisqu'ils ne présentent pas un risque pour l'utilisateur. Pour faire ainsi, nous avons développé un commutateur flou. Les entrées de ce commutateur sont l'intrusion devant-gauche, l'intrusion devant-droite, l'intrusion de-côté-gauche et les informations d'intrusion de-côté-droite et qui sont nommées respectivement ( $I_g, I_d, I_{gc}$  et  $I_{dc}$ ), et calculées en respectant le repère de référence lié au robot.

Les variables liées aux intrusions de devant du robot ( $I_g$  et  $I_d$ ) sont associées à quatre sous-ensembles flous comme indiqué dans la figure IV.11 : TP (Très petit), P (Petit), G (Grand), TG (Très grand). Les variables liées aux intrusions dans les deux côtés ( $I_{gc}$  et  $I_{dc}$ ) sont associées à cinq sous-ensembles flous : VS (Très petit), S (Petit), M (de Milieu), G (Grand)((Super)), VG (Très grand) comme présenté dans la figure IV.12.

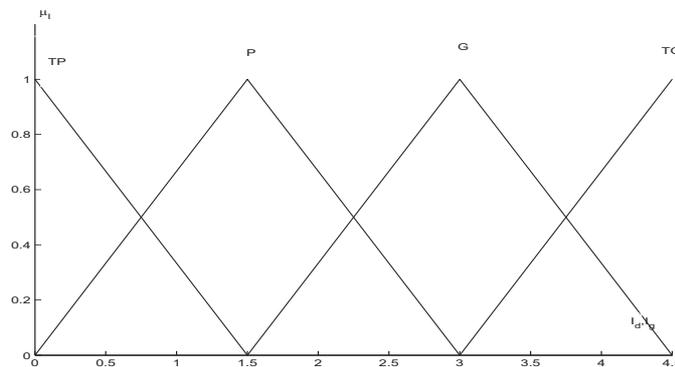


FIG. IV.11 – Les fonctions d'appartenance de l'intrusion de devant (droite et gauche). Ceci implique les déformations avec un angle  $\pm 60^\circ$  par rapport à l'orientation

La sortie du commutateur flou est la variable nommée «Switch» qui est associée à deux sous-ensemble flous de type constante : P (Petit) and G (Grand). La figure IV.13 présente les fonctions d'appartenance de la sortie du commutateur flou proposé.

Pour chaque combinaison d'une variable d'entrée, une action est associée à la variable de sortie. La base de règle de contrôle contient les règles suivantes, pour  $i = 1 \dots r$  :

Si ( $I_g$  est  $A_i$ ) et ( $I_d$  est  $B_i$ ) et ( $I_{dc}$  est  $C_i$ ) et ( $I_{gc}$  est  $D_i$ ) alors ( $Switch = y_i$ )

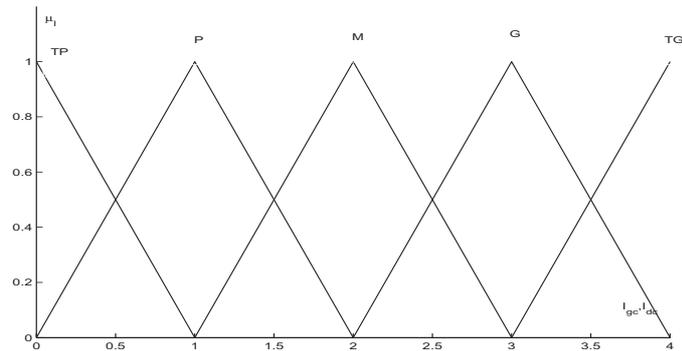


FIG. IV.12 – Les fonctions d'appartenances de l'intrusion de côté (droite et gauche)

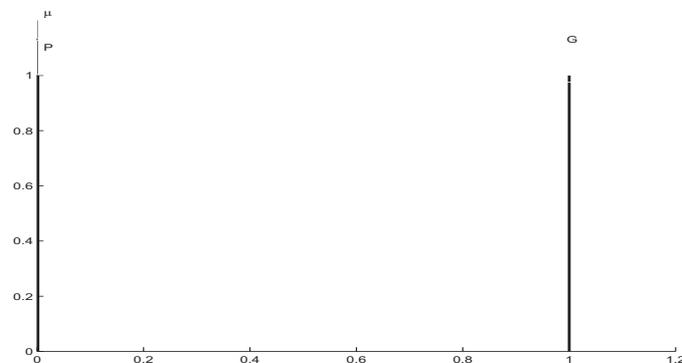


FIG. IV.13 – Variables de sortie du commutateur flou

L'ensemble des règles floues générées en se basant sur une relation "situation/action", est résumé dans le tableau IV.1. Ce tableau est construit à la suite de plusieurs simulations et expérimentations sur le fauteuil roulant pour handicapés.

A ce niveau nous avons développé la variation des consignes de vitesses à appliquer sur les roues d'un robot mobile : fauteuil roulant ou ROMNI. Ces consignes proviennent soit de l'algorithme de commande soit de l'algorithme d'évitement d'obstacles. Il reste donc à tester ces algorithmes en simulation et en pratique pour les deux robots mobiles étudiés. Les résultats obtenus pour le fauteuil et le ROMNI seront présentés respectivement dans les sections IV.5 et IV.6.

TAB. IV.1 – Tableau des inférences linguistiques du commutateur flou.

| <i>Switch</i>                              |    | <i>I<sub>g</sub> &amp; I<sub>d</sub></i> |   |   |    |
|--|----|--|---|---|----|
|  |    | TP                                       | P | G | TG |
| <i>I<sub>gc</sub> &amp; I<sub>dc</sub></i> | TP | Z  | G | G | G  |
|  | P  | Z  | G | G | G  |
|  | M  | Z  | G | G | G  |
|  | G  | G  | G | G | G  |
|  | TG | G  | G | G | G  |

## IV.5 Résultats de simulation et expérimentaux du robot unicycle : le fauteuil

Le premier système robotique étudié est un fauteuil roulant contrôlé avec une communication série à partir d'un PC portable ou d'un PC embarqué de type *PC – 104*. L'ordinateur portable exécute les tâches suivantes : calcul de la trajectoire optimisée se basant sur la logique floue, détermination de la position relative du robot en utilisant les mesures des encodeurs, et le calcul des vitesses nécessaires pour éviter les obstacles en se basant sur les mesures du capteur laser. La communication avec le capteur laser est assurée via le port USB ou série en appliquant le protocole d'acquisition et de calcul de la distance adéquate. Ces spécifications de communication avec le laser sont décrites en détails en annexes.

Ces caractéristiques nécessitent de réaliser des expériences dans un environnement interne étendu et avec des obstacles différents. En fait, l'environnement de travail est un laboratoire de recherche de dimension 14m× 14m comme le montre la figure 5. Les obstacles existants sont des chaises, tables, etc, comme le montre la figure suivante. Les principales valeurs numériques appliquées durant les tests avec le fauteuil roulant robotisé sont résumées dans le tableau IV.2.

TAB. IV.2 – Paramètres de contrôle du fauteuil roulant

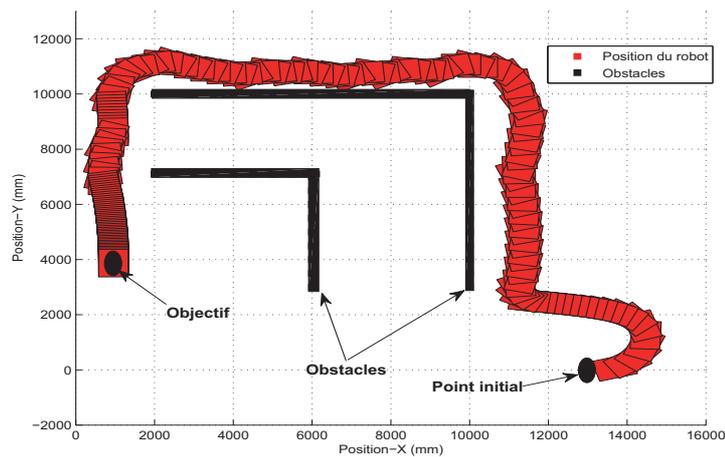
|                              |                          |
|------------------------------|--------------------------|
| $K_V = 0,0001$               | $\lambda_{cx} = 0,00096$ |
| $K_\omega = 0,15$            | $V = 300$                |
| $c_x^{min} = 800 \text{ mm}$ | $\omega_c = 1,5$         |



FIG. IV.14 – Environnement de travail du fauteuil

### IV.5.1 Expérience 1 : trajectoire complexe

Le but de cette première expérience est de montrer l'efficacité de la méthode proposée dans une situation non coin mais avec une trajectoire complexe. La figure IV.15 présente des déplacements qui sont effectués en adoptant un point de début défini par les coordonnées ( $x = 13\ 000\text{ mm}$ ,  $y = 0\text{ mm}$  et  $\theta = 0$ ). Le conducteur souhaite atteindre l'objectif défini par les coordonnées ( $x_o = 1\ 000\text{ mm}$ ,  $y_o = 4\ 000\text{ mm}$ ). Le parcours adopté présente des obstacles dans les deux côtés (droite et gauche). Nous remarquons que si les obstacles

FIG. IV.15 – Trajectoire complexe parcourue par le fauteuil avec  $\theta = 0$ 

détectés rentrent à l'intérieur de la zone de sécurité (position (11 000 mm, 2 500 mm) pour laquelle la distance entre le bord du robot et l'obstacle est presque de 1 000 mm) le

robot commence à les éviter, suit les murs jusqu'à atteindre la cible : ce qui explique les courbes de vitesse présentées dans la figure IV.16. Ces courbes montrent que initialement la vitesse droite est largement plus grande que celle gauche. Ceci implique que le robot va tourner à gauche, ce qui est le cas à la position (14 000 mm, 2 000 mm). Le même principe de raisonnement peut être prouvé à travers toute la trajectoire parcourue.

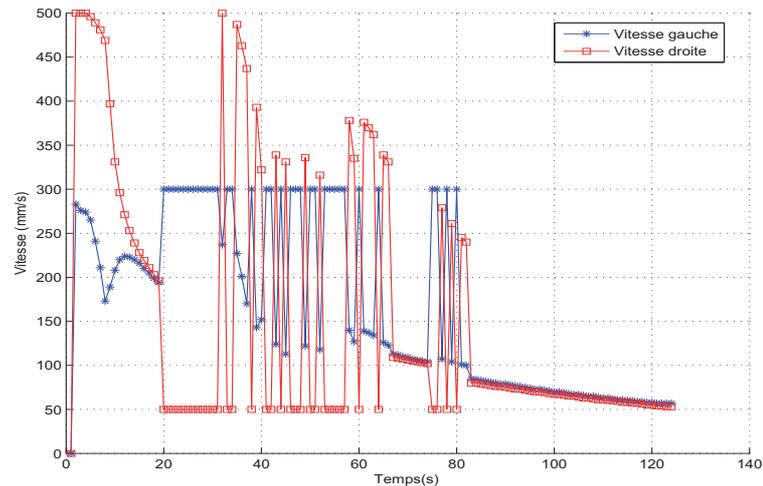


FIG. IV.16 – Vitesses linéaires des roues droite et gauche

### IV.5.2 Expérience 2 : situation de minimum local (coin)

Le but de cette expérience est de montrer l'efficacité de la méthode proposée dans une situation de coin. Dans la figure IV.17, le robot doit essayer de rejoindre une cible définie par les coordonnées ( $x_o = 9\,000$  mm,  $y_o = 8\,000$  mm). Le contrôleur réussit à sortir d'un minimum local (coin), ensuite il effectue des manœuvres (vitesse) résumées dans la figure IV.17 lui permettant de naviguer dans un couloir jusqu'à atteindre le point objectif.

En effet, nous pouvons remarquer que la trajectoire suivie n'est pas optimisée, ainsi les solutions proposées consistent à :

- soit introduire un historique dans l'algorithme afin de construire une carte de l'environnement dans lequel le robot se déplace.
- soit donner plus de valeur à l'interaction avec l'utilisateur ou encore augmenter le poids accordé à la collaboration du conducteur.

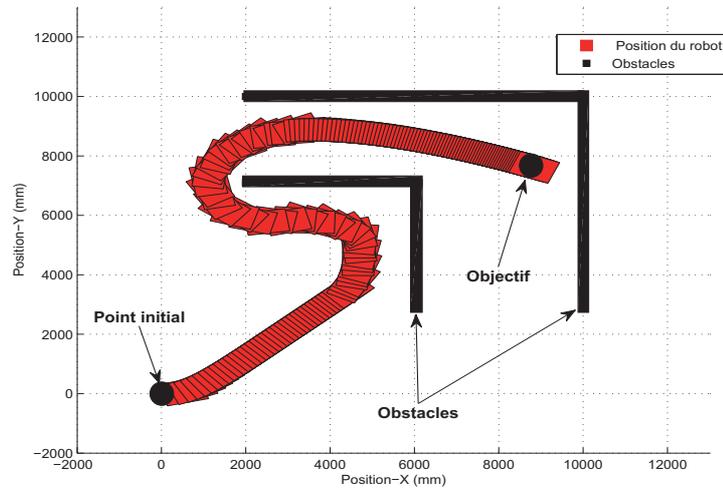


FIG. IV.17 – Trajectoire parcourue par le fauteuil dans une situation de minimum local (coin)

### IV.5.3 Expérience 3 : interaction homme-machine

Dans le but d'optimiser la trajectoire suivie par le robot, nous avons choisi la deuxième solution permettant de mettre en valeur l'interaction homme-machine. Ainsi, nous avons développé une interface utilisateur 2D (GUI) permettant d'exécuter le système de pilotage du fauteuil roulant (commande et évitement d'obstacles). Cette interface est présentée par les figures IV.18 et IV.19.

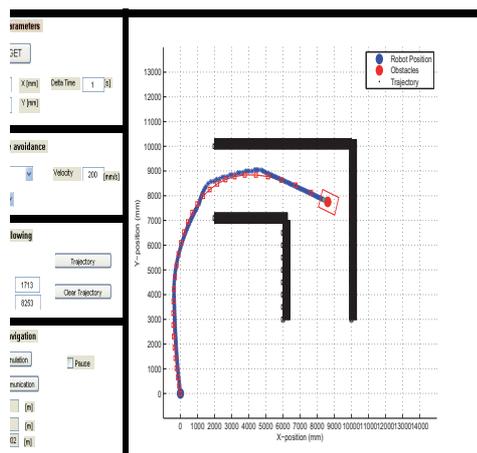


FIG. IV.18 – Interface utilisateur 2D. L'obstacle choisi est un couloir.

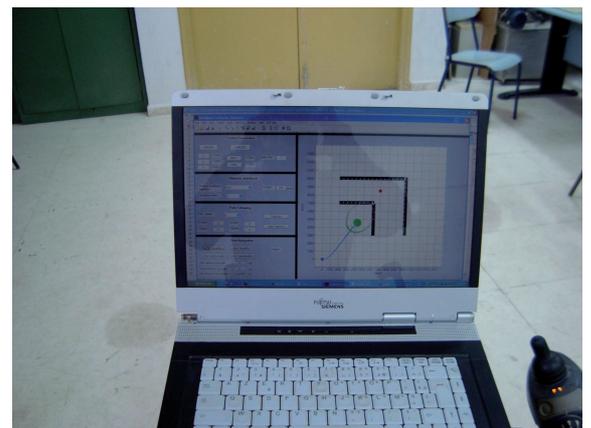


FIG. IV.19 – Interface 2D au cours de la navigation du fauteuil.

Cette interface présente trois modes d'opération : atteindre un point objectif, suivi

de trajectoire et évitement d'obstacles. L'utilisateur peut choisir un ou plusieurs modes d'opération en même temps d'exécution tout en suivant les étapes suivantes :

- L'utilisateur peut interactivement placer le point initial et le point objectif sur l'environnement décrit sur l'interface.
- Puis, il faut choisir un mode de disposition pour les obstacles (couloir, porte, formes différentes).
- Par la suite, si l'utilisateur désire réaliser un suivi de trajectoire, il peut aussi placer des points de passage entre le point initial et le point objectif, pouvant être automatiquement interpolés avec l'outil B-splines, pour créer un chemin.
- Enfin, un appui sur le bouton «Start Simulation» permet de lancer un test de simulation en commande floue et en pilotage réactif par DVZ. L'appui sur le bouton «Start Communication» permet de lancer un test expérimental sur la plateforme robotique.

A travers cette interface 2D, nous avons pu réaliser une série de tests permettant de voir l'impact de la collaboration de l'utilisateur dans l'optimisation de la trajectoire. Parmi les tests réalisés, nous allons nous focaliser sur les deux trajectoires suivantes. La première trajectoire exécutée est une série de points cibles utilisant le B-splines comme le montre la figure IV.20. Dans cette interface, l'utilisateur a placé deux points de passage

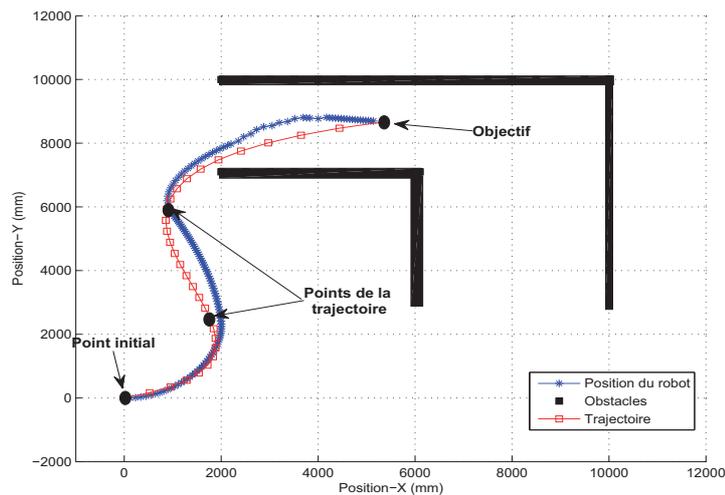


FIG. IV.20 – Suivi de trajectoire : une série de points objectifs avec un minimum local.

dans l'environnement qui s'ajoutent aux points de début et de fin précédent. La trajectoire obtenue est présentée par la courbe rouge (avec des rectangles) tandis que la position réelle du robot est présentée par la courbe bleue (avec des étoiles). L'obstacle choisi dans cette application est un couloir. En comparaison avec la trajectoire de la figure IV.17, nous remarquons que l'interaction avec l'utilisateur nous a permis d'obtenir une trajectoire plus efficace (si on considère le critère de la distance parcourue par le robot).

La deuxième trajectoire réalisée est présentée dans la figure IV.21. Elle montre un exemple de pilotage avec des obstacles de formes différentes et dans lequel le robot réussit en premier lieu à suivre la trajectoire et en deuxième lieu à éviter les obstacles et enfin à rejoindre le point cible. On peut donc conclure qu'une assistance générique nous a permis d'aboutir à un pilotage réactif, rapide et sans risque pour l'utilisateur ; ce qui répond bien aux besoins de l'handicapé.

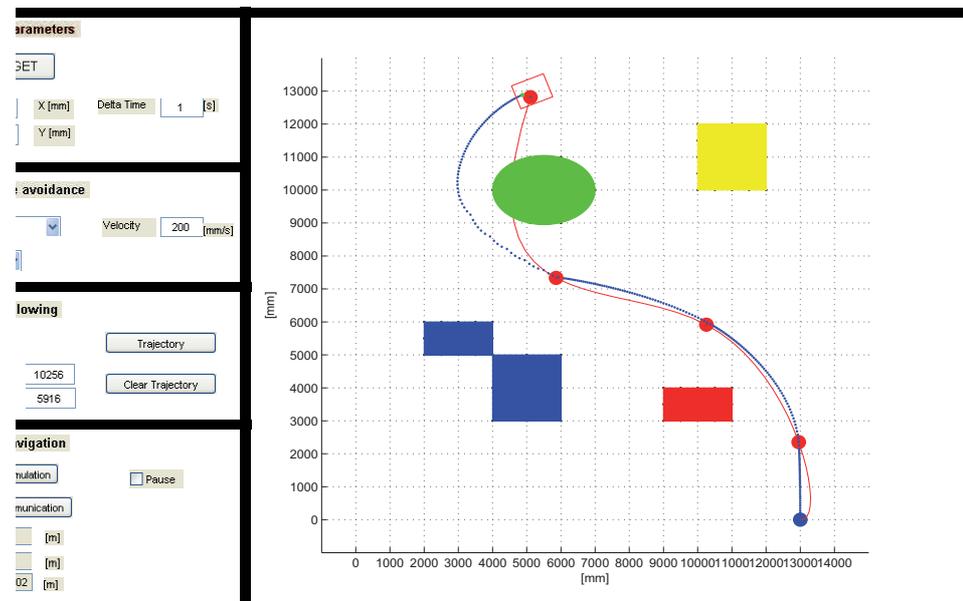


FIG. IV.21 – Suivi de trajectoire avec des obstacles de formes différentes.

## IV.6 Résultats de simulation sur le robot omnidirectionnel : ROMNI

Après avoir testé les algorithmes de commande, de pilotage et de contrôle sur le fauteuil roulant pour handicapés, nous allons maintenant présenter les tests de simulation sur le

robot omnidirectionnel. Les principales valeurs numériques appliquées durant les tests de simulation avec le ROMNI sont résumées dans le tableau IV.3.

TAB. IV.3 – Paramètres de contrôle du ROMNI

|                                    |                                |
|------------------------------------|--------------------------------|
| $K_{\varphi_i} = 0,0001$           | $\lambda_{cx} = 0,00096$       |
| $K_{\omega} = 0,00015$             | $\varphi_i = 300 \text{ mm/s}$ |
| $c_{xi}^{min} = 2\,000 \text{ mm}$ | $\omega_c = 1,5$               |

### IV.6.1 Expérience 1 : situation couloir

L'objectif de la première expérience est de voir la navigation du ROMNI dans un environnement présentant un couloir comme présenté dans la figure IV.22.

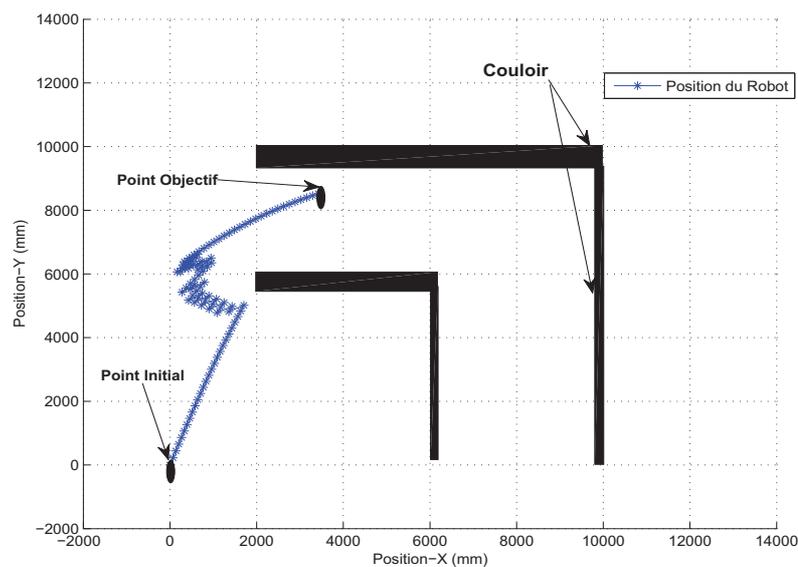
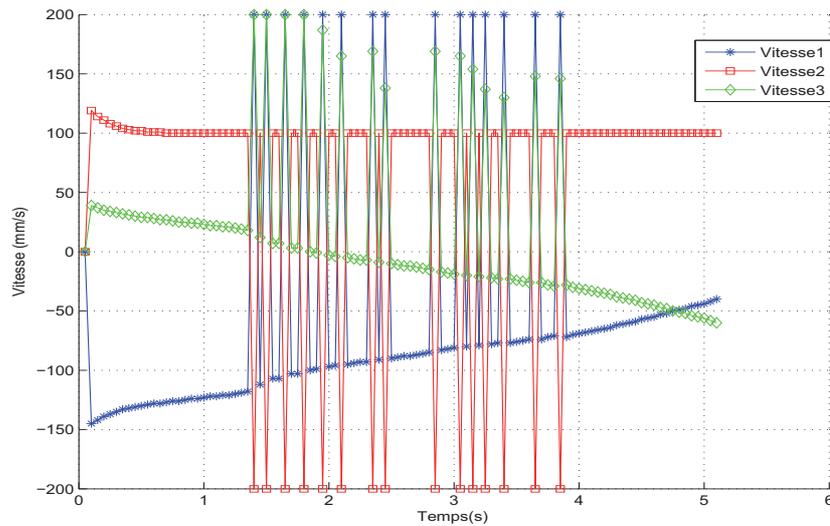


FIG. IV.22 – Trajectoire parcourue par le ROMNI avec un angle initial  $\theta = 0$

La position initiale du robot est définie par les coordonnées ( $x = 0 \text{ mm}$ ,  $y = 0 \text{ mm}$ ) et la cible est définie par les coordonnées ( $x_o = 3\,500 \text{ mm}$ ,  $y_o = 9\,000 \text{ mm}$ ). Dans cet exemple le robot suit la trajectoire définie par le contrôleur flou jusqu'à la position ( $x = 1\,800 \text{ mm}$ ,  $y = 5\,000 \text{ mm}$ ) présentant un obstacle devant le robot, donc c'est le contrôleur DVZ qui intervient pour éviter l'obstacle. Ensuite, c'est le contrôleur de commande qui prend la décision pour mener le robot à destination. La variation des trois vitesses linéaires

FIG. IV.23 – Vitesses linéaires du ROMNI pour  $\theta = 0$ 

du robot est décrite par la figure IV.23. Il faut noter ici que nous avons utilisé un seuil de saturation des vitesses engendrées par le contrôleur ZDV et qui sont appliquées sur le robot. Ce seuil est fixé à la valeur  $\pm 200 \text{ mm/s}$  qui représente la vitesse moyenne du ROMNI.

## IV.6.2 Expérience 2 : obstacles de formes différentes

Le but de la deuxième expérience est de montrer l'efficacité de la méthode proposée dans une situation non coin mais avec un environnement encombré d'obstacles de formes différentes. La figure IV.24 montre la position du robot. Cet essai a été effectué en adoptant un point initial défini par les coordonnées  $(x = 6\,000 \text{ mm}, Y = 6\,000 \text{ mm}, \theta = 180^\circ)$  et un point objectif de coordonnées  $(x_o = 6\,000 \text{ mm}, y_o = 13\,000 \text{ mm})$ .

Initialement le robot vise à atteindre le point objectif placé à un angle  $\varphi = 90^\circ$ . Ainsi il se déplace de 2 mètres jusqu'à la position  $(x = 6\,000 \text{ mm}, Y = 8\,000 \text{ mm})$  où l'obstacle de forme rectangulaire devient à l'intérieur de la zone de sécurité dont la longueur est de 2 mètres comme le montre le tableau IV.3. A ce niveau, le robot commence à l'éviter en faisant un déplacement latéral suivant l'axe des X. Quand le robot devient à l'extérieur de la zone de sécurité (direction devant le robot) à la position  $(x = 2\,000 \text{ mm}, Y = 8\,000 \text{ mm})$ , le contrôleur de navigation reprend le pilotage du robot pour le mener à l'objectif. On remarque que le robot s'approche de l'obstacle à la position  $(x = 3\,900 \text{ mm}, Y = 11\,000 \text{ mm})$ , pourtant il achève son parcours jusqu'au point objectif. Ceci s'explique par le fait que cet obstacle est détecté du côté droit. Cette déformation est de poids faible au niveau

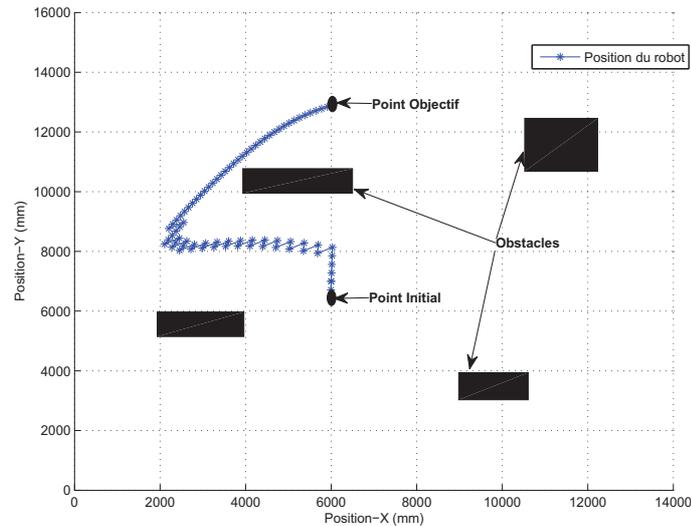


FIG. IV.24 – Trajectoire parcourue par le ROMNI avec un angle initial  $\theta = 180^\circ$

du commutateur flou. Ainsi ce dernier attribue une priorité au contrôleur de navigation dans une telle situation.

## IV.7 Conclusion

Ce chapitre a été consacré à l'étude de la navigation d'un robot mobile en présence d'obstacles détectés au cours du déplacement. Pour cela une étude sur les équipements expérimentaux utilisés a été nécessaire (capteur laser, fauteuil roulant). Dans un premier temps, nous avons supposé que l'espace de navigation contient un couloir que le robot doit éviter ses murs pour atteindre son objectif. La partie commutation a été basée sur l'hypothèse d'hystérésis. Cependant, cette méthode présente des inconvénients une fois testée en pratique. Une interface de commande en 2D a été bénéfique au moment de la simulation et lors de l'expérimentation, ce qui a facilité amplement nos travaux.

Dans un second lieu, nous avons intégré une architecture de contrôle générique permettant d'assurer des commandes rapides au robot, et assurant une commutation continue entre la partie commande floue et la partie pilotage réactive. Des résultats de simulation et expérimentaux sur le fauteuil roulant pour handicapés ont permis de valider l'architecture proposée, spécialement pour des situations de coin. Ensuite ce contrôleur a été implémenté et simulé sur le robot omnidirectionnel ROMNI tout en tenant compte des caractéristiques cinématiques et mécaniques de cette plateforme qui requiert des modifications au niveau

de l'algorithme d'évitement d'obstacles.

Pour améliorer les résultats obtenus et les rendre plus proches des besoins de l'handicapé qui, en plus de l'autonomie, exige un certain degré de sécurité, nous avons intégré une partie collaboration entre l'utilisateur et le robot.

# Conclusions et Perspectives

L'étude menée dans ce rapport a été orientée vers l'amélioration de l'autonomie des systèmes robotiques. Deux aspects de l'autonomie ont été mis en valeur : l'autonomie de la mobilité et l'autonomie de la décision.

Notre contribution à ce niveau consiste d'une part à la proposition de contrôleurs flous optimisés permettant l'amélioration de l'autonomie de déplacement. D'autre part, nous avons mis au point un pilote réactif pour des robots mobiles à roues tenant compte des informations extéroceptives.

La mobilité d'un robot est apportée par son architecture structurelle qui peut être commandée à travers son modèle cinématique. Cette étude de la structure mécanique et du modèle mathématique a été menée sur trois robots mobiles à roues : deux robots unicycles et un robot omnidirectionnel. L'autonomie de déplacement de ces robots s'avère différente, évidemment puisque le robot omnidirectionnel présente une autonomie générale comparée au robot unicycle caractérisé par deux degrés de mobilité. Ainsi, un contrôleur à base de la logique floue a été synthétisé dans le but d'augmenter l'autonomie des robots unicycles comparée à ceux de type omnidirectionnel.

Cependant, l'autonomie de mobilité est insuffisante pour garantir une autonomie complète. Des informations sur l'environnement externe du robot sont nécessaires pour réaliser des prises de décision autonomes. Notre première information extéroceptive a été basée sur l'utilisation d'une caméra de type Webcam permettant la localisation absolue d'un système robotisé. L'étude de cette partie décisionnelle a été menée sur un robot unicycle (Khepera II). Les résultats obtenus montrent une amélioration de la précision de localisation du robot. Néanmoins, cette proposition a été restreinte à un environnement non encombré ce qui engendre une autonomie de décision incomplète.

Notre deuxième information sur l'environnement externe a été basée sur un capteur laser, qui à travers un algorithme d'évitement d'obstacles réactif par Zone de Déformations Virtuelles (ZDV), nous a conduit à concevoir un système robotisé ayant plus de degré d'autonomie. Cette étude a été menée sur deux robots mobiles à roues (fauteuil roulant

et ROMNI).

Cependant, l'organisation de l'ordre d'exécution des deux parties : commande (autonomie de déplacement) et pilotage (autonomie décisionnelle) nécessite l'utilisation d'une architecture de commande. L'architecture que nous avons utilisée est de type générique. Elle se base sur l'utilisation de modules hiérarchiques qui intègrent la notion d'interaction homme-machine dans les différents niveaux.

Cette architecture de contrôle réactive a été introduite dans le but d'assurer plus d'autonomie au robot, des actions plus rapides et une interaction plus fiable entre les capteurs (perception) et les actionneurs (décision-action). Une interface de supervision en 2D nous a facilité la phase de test de situations critiques et d'intégrer la notion de collaboration homme-machine dans notre concept.

En perspective, nous envisageons, dans nos prochains travaux, de nous focaliser sur :

- l'intégration de la fusion de données capteurs afin d'améliorer encore la précision obtenue essentiellement dans le cas du fauteuil roulant.
- l'implémentation d'une commande du fauteuil roulant pour handicapés en utilisant les impulsions nerveuses du cerveau (capteur EEG) aussi bien que les signaux oculaires.
- l'extension de l'interface utilisateur 2D pour aboutir à une solution en 3D qui peut être bénéfique dans le sens où elle procure une réalité virtuelle sur l'évolution de n'importe quel système robotique.

# Bibliographie

- [1] H. Takahashi, h.l. biitsuharu et K. Ohnisd : " Mobility of a Mobile Robot". AMC-Kawasaki, Japan, 2004.
- [2] L. Amouri-Jmaiel, M. Jallouli et N. Derbel : "An Effective Sensor Data Fusion Method for Robot Navigation Through Combined Extended Kalman Filters and Adaptive Fuzzy Logic". Transactions on Systems, Signals and Devices, vol. 4(1), pp. 1–18, 2009.
- [3] G. Mourioux : "Proposition d'une architecture multifonctions pour l'autonomie globale des robots". Université d'Orléans, Rapport de thèse, 2006.
- [4] G. Mourioux, G. Poisson, C. Novalés : " Robot Omnidirectionnel : concepts et analyse". 17ème Congrès Français de Mécanique, Troyes, France, Septembre 2005.
- [5] L.C. Bento, U. Nunes : " Autonomous Navigation Control With Magnetic Markers Guidance of a Cybernetic car Using Fuzzy Logic". Machine Intelligence and Robotic Control, vol. 6(1), pp. 1–10, 2004.
- [6] D. Filliat : " Robotique mobile". Cours ENSTA, 2004.
- [7] O. Trullier et J. A. Meyer : "Biomimetic navigation models and strategies in animats". AI Communications, vol. 10, pp. 79–92, 1997.
- [8] O. Trullier et J. A. Meyer : "Animat navigation using a cognitive graph". Biological Cybernetics, vol. 83(3), pp. 271–285, 2000.
- [9] O. Trullier, S. Wiener, A. Berthoz, et J. A. Meyer : "Biologically-based artificial navigation systems : Review and prospects". Progress in Neuro- biology, vol. 51, pp. 483–544, 1997.
- [10] B. A. Cartwright et T. S. Collett : "Landmark maps for honeybees". Biol. Cybern., vol. 57, pp. 85–93, 1987.
- [11] D. Lambrinos, R. Muller, T. Labhart, R. Pfeifer et R. Wehner : "A mobile robot employing insect strategies for navigation". Robotics and Autonomous Systems, special issue : Biomimetic Robots, vol. 30, pp. 39–64, 2000.

- 
- [12] P. Gaussier, C. Joulain, J. P. Banquet, S. Lepretre et A. Revel : "The visual homing problem : an example of robotics/biology cross-fertilisation". *Robotics and autonomous systems* , vol. 30(1–2), pp. 155–180, 2000.
- [13] E. Maalouf, M. Saad, H. Saliah et F. Mnif : "Integration of a novel path planning and control technique in a navigation strategy". *International Journal of Modelling, Identification and Control (IJMIC)*, vol. 1(1), pp. 52–62, 2006.
- [14] L. Ming, G. Zailin et Y. Shuzi : "Mobile robot control optimization using genetic algorithm". *Artificial Intelligence in Engineering*, vol. 10, pp. 293–298, 1996.
- [15] D. Fox, W. Burgard, S. Thrun : "The Dynamic Window Approach to Collision Avoidance". *IEEE Robotics and Automation Magazine*, vol. 4(1), pp. 23–33, 1997.
- [16] O. Khatib : "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots". *International Journal on Robotics Research*, vol. 5(1), pp. 90–98, 1986.
- [17] P. Ogren et N. Ehrich-Leonard : "A Convergent Dynamic Window Approach to Obstacle Avoidance". *IEEE Transactions on Robotics and Automation*, vol. 21(2), pp. 188–195, 2005.
- [18] N. Franceschini, J. Pichon et C. Blanes : "From insect vision to robot vision". *Philosophical Transactions of the Royal Society of London, Series B*, pp. 283–294, 1992.
- [19] Gourichon, S. Meyer, J. A. Pirim : "Using colored snapshots for short-range guidance in mobile robots". *International Journal of Robotics and Automation : Special Issue on Biologically Inspired Robotics*, vol. 17(4), pp. 154–162, 2002.
- [20] J. Serres, F. Ruffier, S. Viollet, N. Franceschini : "Toward optic flow regulation for wall following non centring behavior". *International Journal of Advanced Robotics systems*, vol. 3(2), pp. 147–153, 2006.
- [21] M. Carballeda, C. Leroux : "Gestion de missions de Sécurité Civile incluant un micro-UAV". *Projet FP6-IST-045248,  $\mu$ Drones, WISG'2009*.
- [22] A. Cacitti, R. Zapata : "Reactive Behaviours of Mobile Manipulators Based on the DVZ Approach". *IEEE International Conference on Robotics and Automation*, Seoul, Korea, pp. 680–685, May 21-26, 2001.
- [23] L. Lapierre, R. Zapata et P. Lepinay : "Simultaneous Path Following and Obstacle Avoidance Control of a Unicycle-type Robot". *IEEE International Conference on Robotics and Automation*, Roma, Italy, pp. 2617–2622, April 10-14, 2007.
- [24] L. Lapierre, R. Zapata et P. Lepinay : "Simultaneous Path Following and Obstacle Avoidance Control of a Unicycle-type Robot". *The International Journal of Robotics Research*, vol. 26(361), pp. 261–375, 2008.
- [25] R. Arkin : "Behavior-Based Robotics". *The MIP Press*, 1998.

- 
- [26] R. A. Brooks : "Intelligence without representation". *Artificial Intelligence*, vol. 47, pp. 139–159, 1991.
- [27] R. R. Murphy : "Introduction to AI Robotics". The MIT Press, 2000.
- [28] C. Novalés, G. Mouriaux, G. Poisson : "Une Architecture Modulaire de Commande de Robots : de l'Autonomie à la télé-opération". *Journal Européen des Systèmes Automatisés (JESA)*, vol. 5, pp. 1–20. 2008.
- [29] Ph. Hoppenot, M. Benreguieg, H. Maaref, E. Colle, C. Barret : "Control of medical aid mobile robot based on fuzzy navigation". *IEEE/IMACS international conference (CESA) on robotics and cybernetics*, Lille, pp. 388–393, 1996.
- [30] V. Kanakakis, K. P. Valavanis, N. C. Tsourvaseloudis : " Fuzzy-Logic Based Navigation of Underwater Vehicles". *Journal of Intelligent and Robotic System*, vol. 40, pp. 45–88, 2004.
- [31] L. A. Zadeh : "Fuzzy sets". *Information and control*, vol. 8, pp. 338–353, 1965.
- [32] H. Surmann, J. Husser, L. Peters : "A Fuzzy system for indoor mobile robot navigation". *Proceeding of IEEE international Conference on Fuzzy Systems*, Yokohama, Japon, vol. 1, pp. 83–88, 1995.
- [33] H. Lu, C. Chuang : "The Implementation of Fuzzy - Based Path Planning for Car-Like Mobile Robot". *International Conference on MEMS, NANO and Smart Systems (ICMENS)*, (2005).
- [34] M. Sugeno, M. Nishida : "Fuzzy control of a model car". *Fuzzy Sets Systems*, vol. 16, pp. 103–113, 1985.
- [35] F. Cupertino, V. Giordano, D. Naso et L. Delfino : " Fuzzy Control of a Mobile Robot". *IEEE Robotics and Automation Magazine*, pp. 74–81, 2006.
- [36] A. Fahmi, L. Khriji, A. Al-Yahmadi, N. Masmoudi : " Contribution of fuzzy logic based autonomous robot navigation in unknown environment". *Fourth International Multi-Conference on system, Signals and Devices*, Tunisie, 2007.
- [37] L. X. Wang, J. M. Mendel : "Generating fuzzy rules by from numerical data, with application applications". USC SIPI, n°. 169, Univ. Southern. Calif, Los Angeles, 1991.
- [38] V. Prahlaad, O. C. M. Xiao, P. H. L. Tong : "Fuzzy Behavior-Based Control of Mobile Robots". *IEEE Transactions On Fuzzy Systems*, vol. 12(4), 2004.
- [39] Manuel de K-Team : "Khepera II user manuel". Switzerland, 2002.
- [40] D. L. Thomas : "Optimal Fusion of Sensors". Department of Automation Technical University of Denmark, PHD-Essay, pp. 1–9, 1998.

- 
- [41] R. Carelli, J. Santos-Victor, F. Roberti, S. Tosetti : "Direct visual tracking control of remote cellular robots". *Robotics and Autonomous Systems Journal*, vol. 54, pp. 805–814, 2006.
- [42] M. Jallouli : "Localisation absolue d'un robot mobile dans un univers 3D par un système multibalisés". Thèse, Université Paris XII, 1991.
- [43] D. N. Green et J. Z. Sasiadek : "Path tracking obstacle avoidance and deal reckoning by an autonomous planetary rover". *International journal of Vehicle design*, vol. 1, 1998.
- [44] F. Chenavier et J. L. Crowley : "Position estimation for a mobile using vision and odometry". *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2588–2593, France, 1992.
- [45] M. Ho Kim, S. C. Lee et K. H. Lee : "Self Localization of Mobile Robot With Single Camera in Corridor Environnement". *Conf ISIE*, vol. 3, pp. 1619–1623, 2001.
- [46] L. Freda, G. Oriolo : "Vision-based interception of a moving target with a nonholonomic mobile robot". *Robotics and Autonomous Systems Journal*, vol. 55, pp. 419–432, 2007.
- [47] J. Serra : "Image analysis and mathematical morphology". Academic Press, 1982.
- [48] J. Serra : "Eléments de théorie pour l'optique morphologique". Thèse, Université Paris VI, Janvier 1986.
- [49] F. Laporterie : "Représentations hiérarchiques d'images avec des pyramides morphologiques. Application à l'analyse et à la fusion spatio-temporelle de données en observation de la Terre". Thèse, Ecole Nationale Supérieure D'Aéronautique et de l'Espace, Mai 2002.
- [50] S. Y. Choi, J. M. Lee, C. K. Song et H. H. Choi : "The detection of Lanes and Obstacles in Real Time Using Optimal Moving Window". *JSME International Journal*, vol. 44(2), 2001.
- [51] O. R. Vincent, O. Folorunso : "A Descriptive Algorithm for Sobel Image Edge Detection ". *Proceedings of Informing Science and It Education Conference (InSITE)*, 2009.
- [52] L. Amouri, C. Novales, M. Njah, M. Jallouli et G. Poisson : "DVZ-based Obstacle Avoidance Control of a Wheelchair Mobile Robot". *IEEE International Conference on Mechtronics (ICM)*, Turquie, 2011.
- [53] Y. Matsumoto, T. Ino, et T. Ogasawara : "Development of intelligent wheelchair system with face and gaze based interface." *10th IEEE Int. Workshop on Robot and Human Communication (ROMAN 2001)*, pp. 262–267, 2001.

- 
- [54] P. Jia, H. H. Hu, T. Lu et K. Yuan : "Head gesture recognition for hands-free control of an intelligent wheelchair". *International Journal of Industrial Robot*, vol. 34(1), pp. 60–68, 2007.
- [55] T. Taha, J. Miro et G. Dissanayake : " Pomdp-based long-term user intention prediction for wheelchair navigation". *IEEE International Conference on Robotics and Automation(ICRA)*, Pasadena, LA, pp. 3920–3925, 2008.
- [56] J. Millan, F. Renkens, J. Mourino et W. Gerstner : " Non invasive brain-actuated control of a mobile robot by human eeg". *IEEE Trans. Biomed Eng*, vol. 51(6), pp. 1026–1033, 2004.
- [57] S. Levine, D. Bell, L. Jaros, R. Simpson, Y. Koren et J. Borenstein : "The nav-chair assistive wheelchair navigation system". *IEEE Transactions on Rehabilitation Engineering*, vol. 7(6), pp. 443–451, 1999.
- [58] T. Carlson et Y. Demiris : "Increasing Robotic Wheelchair Safety With Collaborative Control : Evidence from Secondary Task Experiments". *IEEE International Conference on Robotics and Automation (ICRA 2010)*, Anchorage, Alaska, pp. 5582–5587, 2010.
- [59] D. Wang et M. Pham : "A high-fidelity co-simulation platform for motion and control research for vehicle platooning". *International Journal Vehicle Autonomous Systems*, vol. 6(1–2), pp. 104–121, 2008.
- [60] M. Yguel, O. Aycard et C. Laugier : "Efficient GPU-based construction of occupancy grids using several laser range-finders". *International Journal Vehicle Autonomous Systems*, vol. 6(1–2), pp. 48–83, 2008.
- [61] R. Benenson, S. Petti, T. Fraichard et M. Parent : "To wards urban driverless vehicles". *International Journal Vehicle Autonomous Systems*, vol. 6(1–2), pp. 4–23, 2008.
- [62] T. Carlson et Y. Demiris : " Human-Wheelchair Collaboration Through Prediction of Intention and Adaptive Assistance". *IEEE International Conference on Robotics and Automation (ICRA 2008)*, Pasadena, CA, pp. 3926–3931, 2008.
- [63] L. Qinan, C. Weidong et W. Jingchuan : " Dynamic Shared Control for Human-Wheelchair Cooperation". *IEEE International Conference on Robotics and Automation (ICRA 2011)*, Shangai Jiao Tang University, Chine, 2011.
- [64] M. Jallouli, L. Amouri-Jmaiel et N. Derbel : "An Effective Localization Method for Robot Navigation Through Combined Encoders Positioning and Retiming Visual control." *Journal of Automation, Mobile Robotics and Intelligent Systems (JAMRIS)*, vol. 3(2), pp. 15–23, 2009.

- 
- [65] J. Minguez, L. Montano, J. Santos-Victor : "Reactive navigation for nonholonomic robots using the ego-kinematic space". International Conference on Robotics and Automation, Washington, DC, USA, 2002.
  - [66] M. Hentschel, O. Wulf et B. Wagner : "A Hybrid Feedback Controller For Car-Like Robots Combining Reactive Obstacle Avoidance and Global Replanning". International Conference on Informatics in Control, Automation and Robotics, Portugal, 2006.
  - [67] J. Saunders et R. Beard : "Reactive Vision Based Obstacle Avoidance with Camera Field of View Constraints". AIAA Guidance, Navigation and Control Conference and Exhibit, Hawaii, 2008.
  - [68] W. Feiten, R. Bauer, G. Lawitzky : "Robust obstacle avoidance in unknown and cramped environments". International Conference on Robotics and Automation, 1994.

# Annexes

Le capteur extéroceptif que nous avons utilisé est un télémètre laser "HOKUYO URG-04LX ". Ce télémètre laser procure deux types de liaisons qui sont la liaison série RS232 et la liaison USB. Par ailleurs, la communication est assurée grâce à un protocole de communication développé par le constructeur. Dans cette partie nous allons présenter les différentes spécifications du capteur laser Hukoyo ainsi que les programmes d'acquisition et de conversion de données développées.

## 1 Spécifications du capteur laser

Le télémètre laser Hukoyo nécessite une alimentation stabilisée fournissant un courant minimal de 500mA et une tension de 5V. Une fois mis sous tension, il couvre une zone de balayage de 240 degrés comme décrit dans la figure 1. La résolution angulaire par pas est de  $360 \text{ degré}/1024 = 0.3515625 \text{ degré}$ . La zone hachurée de la figure 1 représente l'angle de balayage du télémètre laser utilisé. Les mesures ne sont effectives qu'à partir du pas N°44 au pas N°725. En conséquence, le nombre total de pas disponible est de 682 pas représentant un champ de vision de 240 degrés. Par contre, la partie non hachurée de la figure 1 représente la zone morte du télémètre laser dans laquelle aucune mesure ne peut s'effectuer. De même, les zones comprises simultanément entre les pas 0 à 44 et 725 à 768 sont considérées comme zones aveugles car le champ de balayage du capteur laser est limité à 240 degrés. Les pas numéros "0", "384" et "768" impliquent respectivement -135 degrés, 0 degré et +135 degrés tout en considérant le pas numéro 384 comme un point de référence ou le centre de la zone de balayage du capteur laser.

## 2 Protocole de communication du capteur laser

Une fois connecté, le port USB permet une communication automatique avec le capteur laser. Cette liaison est prioritaire si un port RS232 est utilisé en même temps. Pour ce

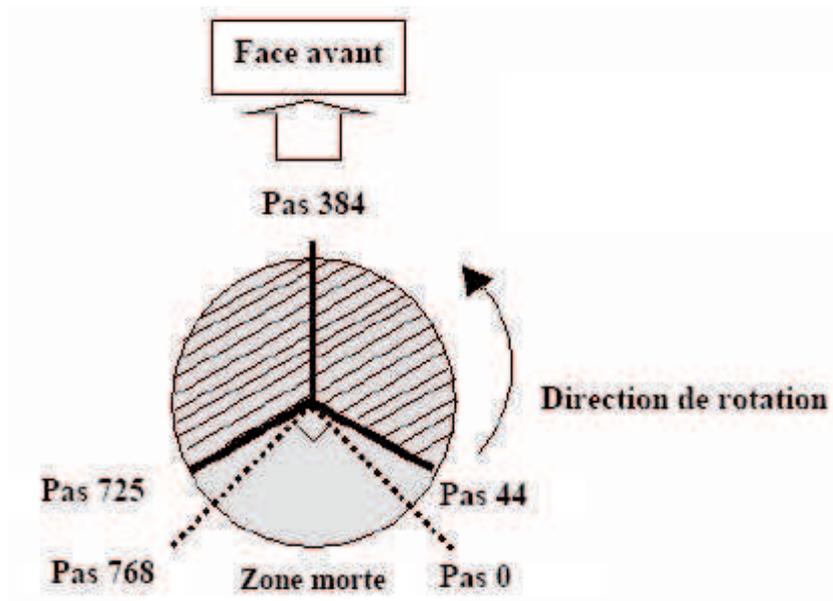


FIG. 1 – Intervalle de balayage du capteur laser Hukoyo

type de liaison, la vitesse de communication entre l'ordinateur et le télémètre est unique et elle est de 12Mbps. L'accès au télémètre laser ne devrait se faire que si la configuration PC-télémètre est complète et correcte. En effet, l'ordinateur reconnaît le capteur laser via un pilote de communication USB (*URGUSBdriver*) fourni par le constructeur et qui contient les outils nécessaires pour l'exploitation du télémètre à partir de la liaison USB. Le format de la communication à partir de l'ordinateur est "Commande", "Paramètres" (si nécessaire) et "symbole de terminaison" (ex. Line feed[0a16] or carriage return[0d16]) en succession. La trame émise sera ensuite transmise par le capteur suivie du "Statut", "Line feed" et suivant la commande émise nous recevons les "Données", et finalement le "Symbole de terminaison". Le symbole de terminaison pour la trame émise par l'ordinateur est de 1 seul bit. Par contre dans la trame transmise par le capteur, il est de 2 bits pour pouvoir séparer les deux trames. Concernant les commandes, le constructeur en a défini quatre types afin de pouvoir utiliser le télémètre laser :

1. **Version command(V-command)** : permet de tester la communication.
2. **Laser illumination command(L-command)** : permet l'arrêt ou la remise en marche du laser.
3. **Communication Speed Setting Command for RS-232C (S-command)** : permet de configurer la communication(débit, bit de stop...).
4. **Distance Data Acquisition Command (G-command)** : permet l'acquisition des mesures tout en définissant leurs nombres et le champ de balayage.

La figure 2 est une représentation générale de n'importe quelle trame émise ou reçue du capteur laser Hukoyo. Nous allons présenter maintenant un exemple de trame d'acqui-

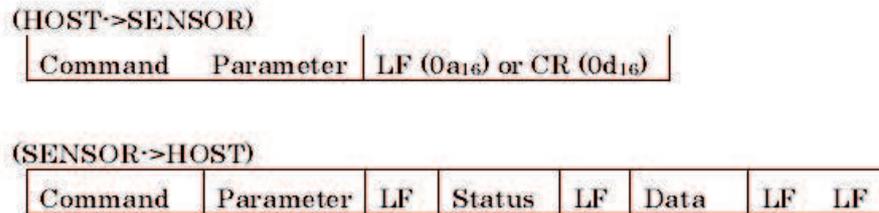


FIG. 2 – Requêtes émises et reçues

tion de données avec la commande G-command. L'acquisition des mesures, leurs nombres et la détermination du champ de balayage s'effectuent au moyen des commandes envoyées sous formes de trames vers le télémètre laser. Cette tâche est assurée par le G-Command (Distance Data Acquisition). Les figures 3 et 4 présentent respectivement la trame envoyée de l'ordinateur hôte vers le capteur et la trame de réponse du capteur. Plusieurs

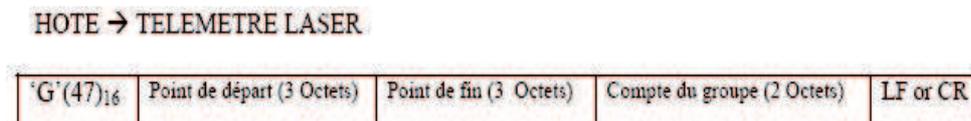


FIG. 3 – Requête émise par l'hôte

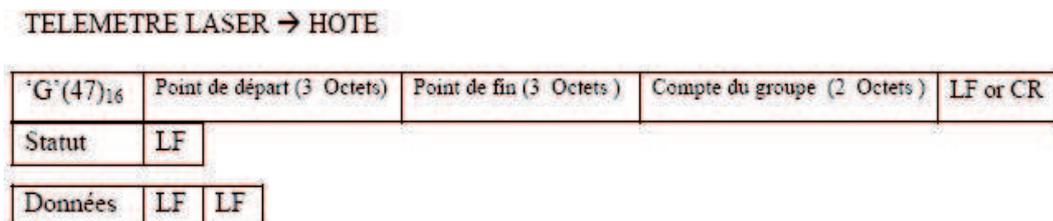


FIG. 4 – Requête de réponse du capteur

paramètres sont à préciser pour obtenir les mesures de distance à partir du télémètre laser. Ces paramètres sont essentiellement : le pas de départ, le pas de fin et le nombre de mesures par groupe.

1. **Le pas de départ** : il varie entre 0 et 768. C'est le pas de la première mesure. Exemple pour le pas 044 on doit envoyer la trame suivante : (30)<sub>16</sub>, (34)<sub>16</sub>, (34)<sub>16</sub>.

2. **Le pas de fin** : il varie entre 0 et 768. C'est le pas de limitation de la zone de mesure. Exemple pour le pas 725 on doit envoyer  $(37)_{16}$ ,  $(32)_{16}$ ,  $(35)_{16}$ .
3. **L'ensemble de mesure** : il varie entre 0 et 99. Il permet de définir le nombre de mesure par scan. Exemple pour un pas de début 300 et un pas de fin 500 et un ensemble de mesure 50 nous aurons  $((500 - 300)/50) + 1$  qui égale à 5 mesures.

La résolution du télémètre laser «HOKUYO URG-04LX» est de 1 mm. Le compteur de distance permet de lire des distances comprises entre 0 et 4095 mm. La distance est codée sur 12 bits. Elle est envoyée sur 2 octets. Chaque 6 bits sont associés à un caractère en code ASCII. Pour trouver la distance en mm on doit faire la procédure de conversion décrite dans la figure 5.

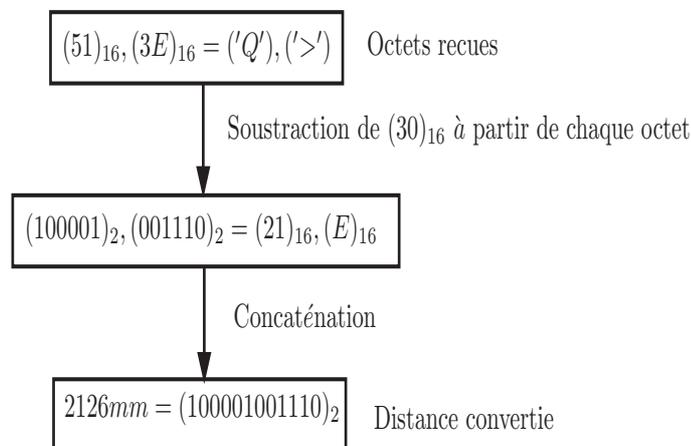


FIG. 5 – Procédure de conversion des données en distance

### 3 Programmes d'acquisition du capteur laser

Le télémètre laser a une grande précision et une très bonne résolution angulaire. Mais vu le nombre important de mesures par balayage (682) et le temps de la sortie synchrone qui fournit un balayage pour 12,5 msec comparé avec la vitesse du processeur, on peut parfois se limiter à un certain nombre de mesures. Les programmes d'acquisition de données sont développés avec le langage de programmation C. La figure 6 présente les mesures de distances acquises par rapport à un obstacle cylindrique entourant le capteur laser.

FIG. 6 – Les distances acquises du Hukoyo en fonction de l'angle

Pour les particularités de la partie programmation, il faut mentionner que le langage C contient une bibliothèque "bios.h" qui permet d'assurer la communication série mais

pour une vitesse de transmission maximale égale à 9600bps. Ainsi, nous avons développé un programme qui permet en premier lieu de créer un fichier dans lequel sont enregistrés les paramètres de la liaison série y compris la vitesse de transmission, le numéro de port, la parité, le nombre de bits de stop et le nombre de bit de données. Cette opération est assurée grâce à la fonction "CreateFile". Ensuite, nous transmettons la trame de commande au capteur laser via la fonction "WriteFile". Enfin, la lecture de la trame transmise par le capteur sera assurée avec la fonction "Readfile". La trame ainsi reçue est codée en hexadécimal et elle contient tous les LF(Line Feed).

**Lobna AMOURI-JMAIEL**

**Contribution à la Commande et au Pilotage réactif de robots  
mobiles à roues**

Résumé :

Dans cette thèse nous avons contribué à la commande floue de deux types de robots mobiles : deux robots de type unicycle (Khepera II et fauteuil roulant). Ensuite, nous avons utilisé une architecture de pilotage réactive permettant d'intégrer la commande floue ainsi qu'un algorithme d'évitement d'obstacles réactif utilisant la théorie de Zones de Déformation Virtuelles (ZDV). Des résultats de simulation et expérimentales ont permis de valider l'approche développée.

Mots clés : Commande floue, évitement d'obstacles ZDV, robot unicycle, robot omnidirectionnel, pilotage réactif.

**Contribution on the Control and the Reactif Pilot of Wheeled  
Mobile Robots**

Abstract :

In this thesis we contributed on developing a fuzzy control of two types of mobile robots : two unicycle robots (Khepera II and wheelchair). Then, we used a reactive piloting architecture insuring the integration of both the fuzzy controller and an obstacle avoidance algorithm using the deformable virtual zones theory (DVZ). Simulation and experimental results validate the developed approach.

Keywords : Fuzzy controller, DVZ obstacle avoidance, unicycle robot, omnidirectionnal robot, reactive pilot.



[CEMLab Ecole Nationale Ingnieurs de Sfax  
PRISME IUT de Bourges]

