

Navigation autonome d'un robot mobile à roues

ARGUI Imane
Argui.imane@gmail.com

BOUNOUADER Hasnaa
bounouaderhasnaa@gmail.com

Projet Master 1
Université de Montpellier
Montpellier, FRANCE

Abstract—La navigation autonome de robots mobiles est un des sujets les plus discutés dans la robotique mobile. Ce sujet traite la navigation d'un robot mobile dans un environnement certain, et sa capacité à éviter les obstacles qui peuvent entraver son chemin. Il sera nécessaire d'abord de planifier la trajectoire la plus optimale pour atteindre son objectif, et ensuite d'établir des commandes que le robot exécutera. Cet article évoque également les problèmes qui sont survenus durant la mise en œuvre du projet.

I. INTRODUCTION

Dans le vaste domaine de la robotique, les robots mobiles n'occupent qu'une place relativement modeste qui peut s'expliquer par leur grande complexité [1], comparée à celle des robots manipulateurs. En effet, alors que ces derniers sont destinés à établir des tâches de manière répétitive et dans un environnement connu, les robots mobiles, par contre, évoluent dans un environnement non structuré, et de manière autonome.



figure 1 Robot mobile [2]



figure 2 Robot manipulateur [3]

En outre les contraintes liées à la conception des robots mobiles, il s'ajoute aussi d'autres contraintes telles que :

- La découverte d'un environnement inconnu au fur et à mesure de leur progression.
- La difficulté de traitement des informations obtenues par les capteurs en temps réel.

- La détermination d'un chemin optimal dans un monde inconnu ou changeant.

Pourtant, les robots mobiles sont fortement utilisés pour les activités «ennuyeuses, salissantes ou dangereuses », pour les environnements qui peuvent être inaccessibles pour l'homme, mais aussi pour les activités ludiques comme l'assistance des personnes âgées ou handicapées. Il s'agit notamment de la robotique de service (hôpital, bureaux, maison), robotique de loisir (jouets, robot 'compagnon'), la robotique industrielle (convoyage (3), nettoyage automatique, mines), robotique spatiale (satellites artificiels, véhicules d'exploration de planète ou de satellites).



figure 3 Robot mobile industriel [4]

La définition générale d'un robot mobile est « un véhicule doté d'une certaine autonomie », c'est-à-dire qu'il est capable de percevoir son environnement grâce aux capteurs, de naviguer dans cet environnement, et d'être capable de s'adapter aux changements de situation.

Le but de ce projet est de traiter la navigation d'un robot mobile. Durant ce travail, l'environnement est connu, et il est attendu du robot de définir la trajectoire la plus optimale pour atteindre son objectif et de la suivre. Bien évidemment, il aura des obstacles à éviter et ne devra pas les heurter.

Nous avons dans un premier temps étudié théoriquement la problématique, nous avons réalisé un algorithme, ensuite nous sommes passés à l'implémentation de l'algorithme dans un robot mobile que nous avons simulé.

II. MATERIELS ET METHODOLOGIES

A. Robot Pioneer 3-DX

Les robots mobiles peuvent se présenter sous différents aspects, que ce soit par rapport à leur forme, à leurs caractéristiques, poids... Le robot qui sera utilisé durant ce projet est le robot Pioneer 3dx. Ce robot est connu pour sa principale place auprès des travaux de recherche, et ce grâce à sa taille et son modèle combinés à un matériel raisonnable. C'est un robot différentiel à deux roues, caractérisé par sa robustesse, crédibilité, capacité d'être personnifié, et surtout programmable avec une haute précision sur le contrôle de son mouvement.



figure 4 : robot Pioneer 3-DX [5]

Les robots mobiles sont généralement équipés de capteurs qui permettent d'éviter les collisions avec des obstacles inconnus ou imprévisibles. Il y'a plusieurs types de capteurs dont les robots peuvent être munis, que ce soit des capteurs mécaniques, tactiles ou ultrasonores ou infrarouges.

Le robot Pioneer 3dx est muni de capteurs ultrasons.

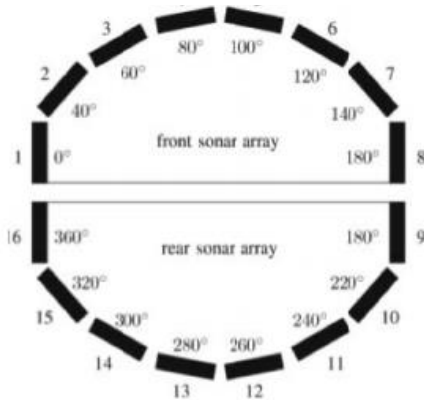


figure 5 : Les capteurs ultrasons du robot Pioneer-3dx avec l'angle de chaque capteur.

B. Capteurs ultrasons

Pour pouvoir utiliser les capteurs du robot, il faudra d'abord comprendre leur fonctionnement.

Les capteurs ultrasonores [6] sont des capteurs qui émettent des impulsions à une fréquence donnée. Ces impulsions se propagent à la vitesse de lumière dans l'environnement. A la présence d'un objet, les impulsions se réfléchissent, et c'est ce qui permet au capteur d'identifier la présence d'un obstacle. Il peut aussi déterminer la distance le séparant de l'obstacle en déterminant le temps de propagation de ces capteurs.

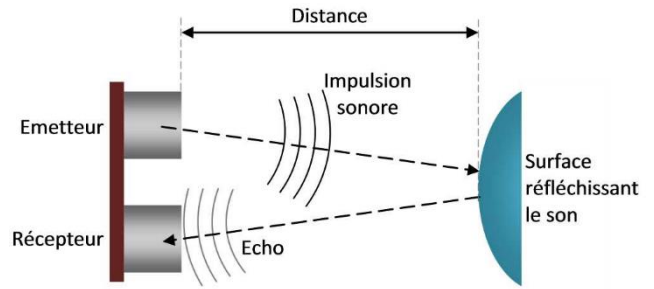


figure 6 : Schéma expliquant le fonctionnement d'un capteur ultrasonore. [7]

C. Robots holonomes et non-holonomes:

Avant de définir ces deux notions il faut d'abord connaître le concept des degrés de libertés d'un robot mobile à roue.

Les degrés de libertés d'un robot mobile sont représentés par le nombre de mouvements autonomes que le robot peut faire par rapport à un repère fixe.

Les robots holonomes sont des robots à trois degrés de libertés

- Translation par rapport à l'axe x
- Translation par rapport à l'axe y
- Rotation autour de l'axe z

Avec trois degrés de libertés les robots holonomes peuvent se déplacer dans toutes les directions quelles que soit leurs orientations.

Les robots non-holonomes [7] sont des robots soumis à des contraintes mécanique ce qui limite la mobilité de ce type de robot. Les mouvements autorisés sont l'ensemble des chemins $q(t)$ dans l'espace de configuration tant que : $q(t) \in \text{vect} \{q(t)\} \subset \mathbb{R}^3$

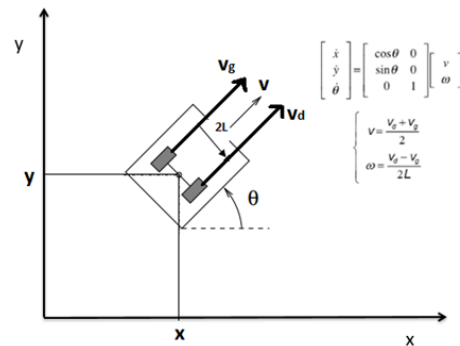


Figure 7 : Modélisation du robot

Le robot Pioneer est considéré comme étant un robot non-holonyme. Effectivement il est restreint par cette contrainte cinématique causée par le fait que la vitesse soit tangente à l'axe du robot. Autrement dit:

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0$$

[8] Il possède 2 roues motrices indépendantes non orientables sur le même axe et une roue décentrée orientable libre « roue folle ». Ses roues sont indéformables et leur contact avec le sol est considéré ponctuel.

D. Planification de trajectoire : Algorithme de DIJKSTRA

Une des premières étapes de la navigation est la planification de la trajectoire. Il est clair qu'un robot peut suivre plusieurs chemins pour atteindre son objectif, mais il reste que le plus intéressant est le chemin le plus optimal à suivre et le plus sûr aussi. Et c'est l'objectif de cet algorithme d'ailleurs. Nous supposons que la carte est déjà connue. (C'est une donnée). Cet algorithme cherche tous les cas possibles pour arriver à l'objectif. Ensuite définit le chemin le plus court.

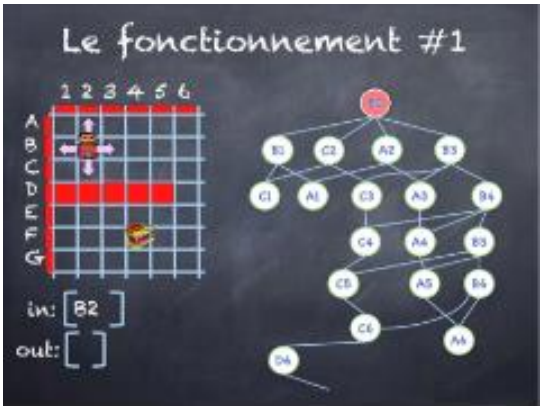


figure 8: schématisation de l'algorithme de Dijkstra

Cet algorithme traite deux types de cartes : les cartes géométriques et cartes continues.[9]
 Les cartes géométriques permettent de diviser l'environnement en cellules dont l'occupation peut être représentée par un 0 ou 1.
 Les cartes continues, représentent les différentes zones où le robot évolue.

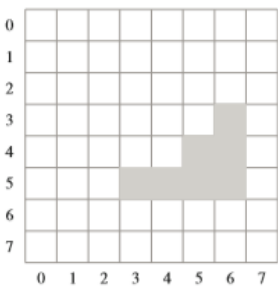


figure 9a: Carte géométrique.

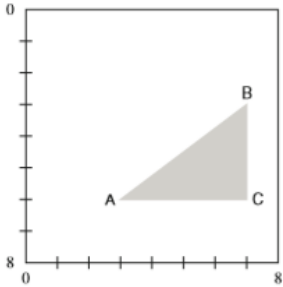
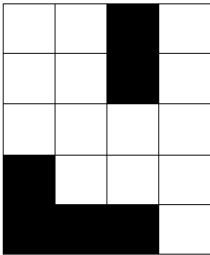


figure 9b: Carte continue

Durant ce projet, les cartes utilisées seront les cartes géométriques. Cela facilite le traitement de données. En effet, si les cellules de la carte sont représentées par des 0 et 1, la carte peut être représentée dans l'algorithme comme étant une matrice de 0 et 1. De ce fait, on peut voir dans la figure 10 l'analogie discutée.



$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

figure 10a: Carte grillée figure 10b: Représentation de la carte

Une fois l'environnement est défini, l'algorithme permet d'abord de parcourir toutes les cellules, tout en définissant le nombre de pas nécessaires pour arriver à l'objectif. La figure 11 représente la carte après avoir été parcouru en entier.

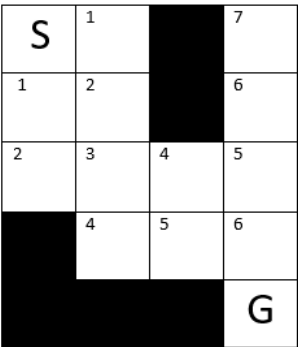


figure 11 : Carte parcourue par l'algorithme

Il devient alors plus facile de définir la trajectoire la plus courte en allant du point G au point S tout en suivant l'ordre chronologique.

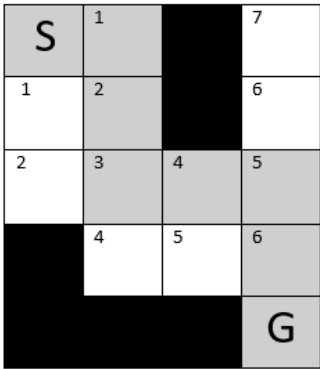


figure 12 : La trajectoire définie par l'algorithme

L'algorithme qui a été appliqué est le suivant :

Algorithme de Dijkstra pour la carte géométrique	
Entier n=0	//Distance depuis point de départ
Matrice carte	//Carte
Tableau chemin	// Le chemin le plus court
Entier actuel	//La cellule actuelle dans le chemin
Entier c	//L'index des cellules
Entier S= ...	//Point de départ
Entier G=...	//Point d'arrivée
1 : S reçoit n	
2 : tant que G n'est pas marquée	
3 : n=n+1	
4 : Pour chaque cellule non marquée dans carte	
5 : à côté d'une cellule marquée	
6 : Marquer c avec n	
7 : fin pour	
8 : fin tant que	
9 : actuel = G	
10 : Ajouter actuel à chemin	
11 : tant que S n'est pas dans chemin	
12 : Ajouter le plus petit voisin de c à chemin	
13 : actuel = c	
14 : fin tant que	
15 : chemin	

figure 13 : Algorithme de Dijkstra

Le résultat final est une liste de cellules qu'il faudra par la suite convertir en commandes que le robot comprendra, et pour ce faire, nous utilisons un autre algorithme qui traite les résultats obtenus sous forme de coordonnées (x, y) afin d'établir des commandes que le robot recevra

Algorithme pour commande robot	
Chemin =	// Tableau de cellules obtenues
b= 0	// Indice
1 : t = taille (chemin)	
2 : Pour i allant de 1 à chemin - 1	
3 : Si chemin (i+1,2)=chemin (i, 2) +25 et b=2	
4 : Robot à gauche // Commande	
5 : Robot tout droit // Commande	
6 : b=1	
7 : Sinon Si chemin (i+1,1)=chemin (i, 1) +25 et b=1	
8 : Robot à droite // Commande	
9 : Robot tout droit // Commande	
10 : b=2	
11 : Sinon Si chemin (i+1,2)=chemin (i, 2) +25 et b=1	
12 : Robot tout droit // Commande	
13 : b=1	
14 : Sinon Si chemin (i+1,1)=chemin (i, 1) +25 et b=2	
15 : Robot tout droit // Commande	
16 : b=2	
17 : Fin Si	
18 : Fin Pour	

Les deux algorithmes ont été implémentés dans le logiciel MATLAB pour se connecter par la suite avec l'outil de simulation Coppelia Sim qui recevra des instructions dédiées au robot Pioneer 3dx.

III. RESULTATS ET DISCUSSION :

Afin de permettre au robot de naviguer dans un environnement connu en toute autonomie, nous avons utilisé les résultats de l'algorithme de Dijkstra afin de créer une commande qui permette de commander le robot indépendamment sur Coppelia Sim. La carte est représentée sur la figure 14

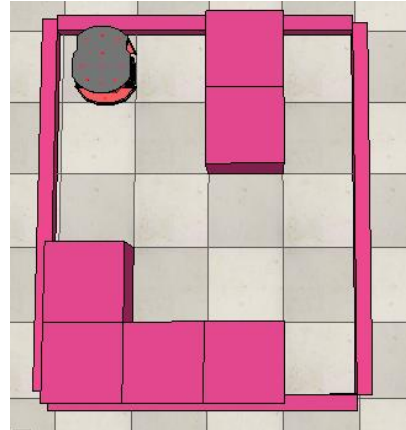


figure 14: Espace de travail sur Coppelia Sim

Plusieurs essais ont été mis en place afin de voir le comportement du robot. Les premières simulations ont été réalisées en se basant uniquement sur les commandes établies par l'algorithme. Les figures 15, 16 et 17 montrent la trajectoire établie par le robot et les problématiques majeures qu'il a rencontrées.

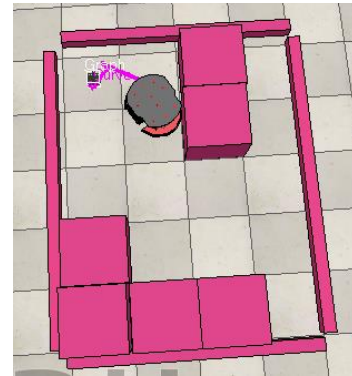


figure 15 : Le robot après la 3ème itération

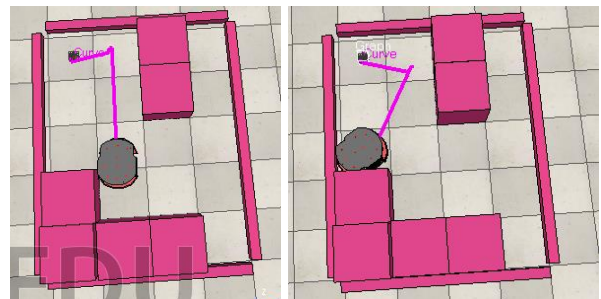


figure 16: le robot après la 5ème itération

Pour une même commande, le robot réalisait des trajectoires différentes, et surtout heurtait les obstacles et continuait quand même sa trajectoire.

Remarque : Pour mieux comprendre la problématique, nous considérons le robot comme étant aveugle, il reçoit des commandes qu'il doit exécuter, mais il a besoin d'une vérification de cette exécution. C'est dans cette partie où intervient l'importance des capteurs. Les capteurs seront considérés comme étant les yeux du robot qui lui permettrait de vérifier au fur et à mesure si la commande a bien été exécutée.

Pour cette fin, nous avons alors décidé d'utiliser les capteurs ultrasoniques dont disposait le robot. Les commandes restaient inchangées, mais les capteurs intervenaient uniquement quand le robot était trop près d'un obstacle afin qu'il s'en éloigne doucement tout en continuant sa trajectoire. Nous avons alors obtenu le résultat représenté dans la figure 17.

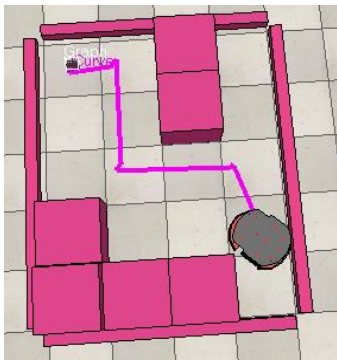


figure 17: La trajectoire finale du robot.

La trajectoire n'est pas encore parfaite, mais l'on peut déjà voir une amélioration sur le chemin effectué par le robot. Il n'y a plus de collisions entre le robot et les obstacles.

Afin de vérifier si l'algorithme est bien correct, il a été impératif de changer la carte et de voir la trajectoire qui sera parcourue pour un objectif donné. La deuxième carte sur laquelle nous avons travaillé est représentée sur la figure 18.

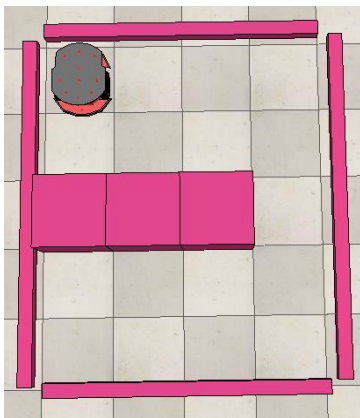


figure 18: La nouvelle carte

La trajectoire a été planifiée avec les capteurs ultrasons et le résultat de la trajectoire est le suivant

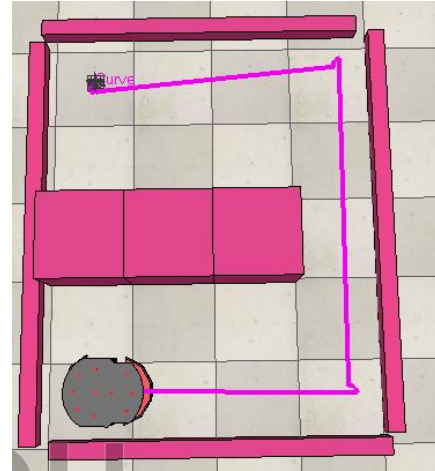


figure 19: La trajectoire effectuée.

Le résultat reste le même, la trajectoire n'est pas encore parfaite, et il arrive que le robot n'exécute pas convenablement les commandes qu'il reçoit ce qui peut changer son chemin. Par ailleurs, les capteurs sont d'une grande utilité en évitant au robot toute collision avec les obstacles, mais ne peuvent en aucun cas changer la trajectoire effectuée par le robot après avoir mal exécuté une instruction.

IV. CONCLUSION :

La navigation autonome [11] de robots mobiles demeure un domaine de recherche actif depuis ces dernières années. Afin qu'un robot puisse naviguer d'une manière sûre dans un environnement connu, il est d'abord nécessaire de calculer préalablement la trajectoire qu'il devra effectuer. L'algorithme de Dijkstra a permis d'effectuer ce calcul. Cependant, l'optimisation du temps est un atout nécessaire dans la navigation, l'algorithme de Dijkstra effectue plusieurs itérations en parcourant toutes les cellules de la carte. Il existe d'autres moyens d'établir le chemin le plus court sans pour autant parcourir toutes les cellules, mais uniquement les cellules pertinentes. Des algorithmes comme A* peuvent en effet aboutir à cette fin en se basant sur une fonction heuristique permettant d'indiquer la direction au but.

Une fois la trajectoire planifiée, il faudra à présent exécuter cette trajectoire à l'aide de commandes. Cette partie est délicate. Le robot devra exécuter des instructions tout en rattrapant les erreurs de suivi effectuées dans les itérations précédentes. Le rôle des capteurs dans cette partie est primordial, c'est ce qui permet d'éviter au robot toute collision avec les obstacles présents dans son environnement. Durant ce projet, nous avons travaillé avec des capteurs ultrasons qui permettent d'indiquer la présence d'objet/obstacles à une distance donnée. Mais il existe également d'autres capteurs qui peuvent tout autant être utilisés à cette fin. Or, les capteurs ne permettent pas d'assurer le suivi de la trajectoire donnée, mais uniquement de rattraper l'erreur de suivi. Cela ne garantit en aucun cas le suivi exact d'une trajectoire. Il existe d'autres capteurs comme la caméra qui peuvent être aussi utilisés. Nous avons à la fin du projet essayé d'établir un lien entre les données reçues par la caméra dans

Coppelia Sim et les commande dans Matlab. La figure suivante représente l'aperçu de la part de la caméra. Il faudra par ailleurs plus que des notions en traitement d'images afin d'utiliser la caméra comme un capteur pour éviter les obstacles.

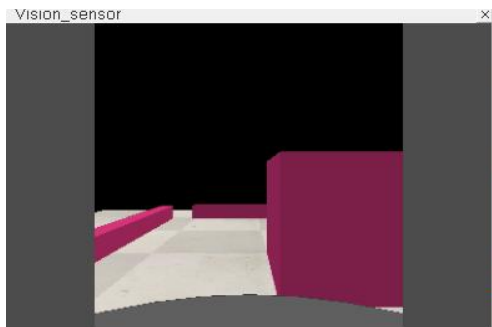


figure 20 : Aperçu de la caméra sur Coppelia Sim

Hormis l'importance des capteurs dans la navigation autonome de robots, il y'a un concept primordial à aborder. Il est désormais devenu presque impossible d'évoquer la navigation d'un robot sans évoquer la localisation du robot. Les roboticiens ont très tôt compris le fort lien entre la navigation et la localisation au point que [12] le "SLAM" pour "Simultaneous Localization and Map Building" est maintenant répandu et compris dans la communauté. La capacité pour un robot de connaître son emplacement à chaque instant est ce qui lui permet d'assurer l'exécution correcte de la trajectoire planifiée. D'ailleurs, c'est ce qui manquait à notre projet, et c'est ce qui répond à notre problématique par rapport à l'exécution des commandes, et la trajectoire établie par le robot. Il a été en fait nécessaire d'avoir une connaissance précise des déplacements du robot.

REFERENCES

- [1] Point en robotique, volume 1, Coordonnateur : J.C. Pruvost, Lavoisier
- [2] <https://free3d.com/3d-model/rigged-boston-dynamics-spot-robot-dog-6994.html>
- [3] https://www.barewalls.com/art-print-poster/industrial-robot_bwc24630724.html
- [4] <https://www.bprfrance.com/2019/10/21/les-robots-mobiles-de-mir-seront-presents-au-salon-europack-euromanut-cfia-2019-de-lyon/>
- [5] http://www.diag.uniroma1.it/~labrob/pub/papers/Springer_EncSystContr_WheeledRobots.pdf
- [6] <http://www.sitedunxt.fr/articles/print.php?id=14>
- [7] <https://www.microsonic.de/fr/support/capteurs-%C3%A0-ultrasons/principe.htm>
- [8] Olivier Lefebvre, "Navigation autonome sans collision pour robots mobiles non holonomes", thèse de doctorat, Institut National Polytechnique de Toulouse, 2006.
- [9] Salah Eddine Ghamri, « Commande neuronale d'un robot mobile non holonome avec glissement des roues », mémoire de magister, Université de Batna 2, Algérie.
- [10] Elements of Robotics, Mordechai Ben-Ari, Francesco Mondada, Springer, Cham, 2018
- [11] Hela Ben Said, "Navigation autonome et commande référencée capteurs de robots d'assistance à la personne", thèse de doctorat, Université de Limoges, 2018.
- [12] Anthony Mallet, "Localisation d'un robot mobile autonome en environnements naturels", thèse de doctorat, Institut National Polytechnique de Toulouse, 2001