

Projet tuteurés

**Systèmes Robotiques: génération de trajectoires, suivi de trajectoire,
évitements d'obstacles**

État de l'art de la détection de trajectoire Robotique

Sous la direction de

Lynda Seddiki

Par :

Toleon kevin

AMINIYEKTA Zhaleh

Juin 2016

Sommier :

Liste des Figure :	4
Chapitre 1	5
1. Introduction :	5
1.1. La robotique mobile :	5
Chapitre 2	6
2. Modélisation	6
2.1. Classification des types de roues	6
2.2. Principales structures cinématiques	8
Chapitre 3	11
3. Évitement réactif d'obstacles pour robots mobiles non holonomes	11
3.1. État de l'art	11
3.2. Méthodes d'évitement réactif d'obstacles sans trajectoire de référence	12
3.2.1. Champs de potentiel	12
3.2.2. Nearness Diagram	12
3.3. Méthodes d'évitement réactif d'obstacles avec une trajectoire de référence	13
3.3.1. Trajectoires d'évitement	13
3.3.2. Bande élastique	13
Chapitre 4	14
4. Les algorithmes existants :	14
4.1. Introduction :	14
4.2. Méthode par décomposition de l'environnement en cellule	14
4.3. La méthode des champs de potentiel	15
4.4. La méthode de la fenêtre dynamique	15
4.5. La méthode de la bande élastique	16
4.6. La planification par logique floue	16
4.7. La planification par juxtaposition de splines polynomiales	16
4.8. DKP : Deterministic Kinodynamic Planning	17
Chapitre 5	17
5. Le Robot RP6	17
5.1. Introduction	17
5.2. Caractéristiques principales	18
5.3. Caractéristiques détaillées	19
5.4. Microcontrôleur	19
5.5. Détecteurs	20
Chapitre 6	22
6. L'environnement de travail	22

6.1. Software	22
6.2. GCC: GNU Compiler Collection	22
6.3. Programmeur Notepad	23
6.4. RP6 Loader	23
Chapitre 7	25
7. Système de Commande	25
7.1 Le système d'évitement d'objets par ultrasons	25
7.2. Les recherches actuelles sur les stratégies d'évitement objet.....	25
.....	26
7.3. Time of flight	26
7.4. Le choix des capteurs à ultrasons.....	27
7.5. Mode Trigger et Echo sur un même pin	28
7.6. Stratégie pour la prévention de l'objet.....	29
Annexe.....	33
Exemples de robots actuels:.....	33
Conclusion :.....	35
Bibliographie :.....	36

Liste des Figure :

FIGURE 1 : ROUE CENTREE ET ROUE DECENTREE	7
FIGURE 2 : ROUE TRON CSPHERIQUE ET ROUE SUEDOISE	8
FIGURE 3 : EQUIVALENCE MODELE VOITURE ET TRICYCLE	9
FIGURE 4 : MODELE ROUES DIFFERENTIELLES	10
FIGURE 5 : LE ROBOT OMNIDIRECTIONNEL ROMNI	10
FIGURE 6 : OVERVIEW SCHEMATIC OF THE RP6 ROBOT SYSTEM.....	19
FIGURE 7 : ATMEGA32 SCHEMA	20
FIGURE 8 : SCHEMA ANTI-COLLISION	21
FIGURE 9 : ABOUT OF PROGRAMMER'S NOTEPAD 2	23
FIGURE 10 : RP6 LOADER SCREENSHOT	24
FIGURE 11 : LEARNING PHASE	26
FIGURE 12 : GOAL SEEKING.....	26
FIGURE 13 : SRF02 BEAM PATTERN	28
FIGURE 14 : SRF05 BEAM PATTERN.....	28
FIGURE 15 : CAPTURES A ULTRASOUNDS.....	28
FIGURE 16 : SCHEMA SRF05	29
FIGURE 17 : OBJECTS AVOIDANCE STRATEGY	30
FIGURE 18 : ESCAPE ANGLE CALCULASSIONS	30
FIGURE 19 : ESCAPE ANGLE CALCULASSIONS	31
FIGURE 20 : ROBOT MILITAIRE DE RECONNAISSANCE ET DETECTION D'ENGIN EXPLOSIF, IL SE DEPLACE A L'AIDE DE DEUX CHENILLES COMME UN TANK.	33
FIGURE 21 : ROBOT FUTURISTE CONSTITUE D'UN CORPS SPHERIQUE TOURNANT SUR LUI-MEME POUR AVANCER, IMAGINE POUR STAR WARS.	33
FIGURE 22 : ROBOT FRANÇAIS SE DEPLACENT L'AIDE DE 3 ROUE, C ROBOT EST DESTINE A INTEGRER LES FOYERS FAMILIAUX IL EST DOTE D'UNE REMARQUABLE INTELLIGENCE ARTIFICIEL.....	34
FIGURE 23 : ROBOT IMITANT LE DEPLACEMENT D'UN KANGOUROU. VERITABLE BIJOUX D'INGENIERIE CE ROBOT EST TOTALEMENT AUTONOME ENERGIE CAR IL REUTILISE L'ENERGIE CINETIQUE DU DEPLACEMENT PRECEDENT POUR EFFECTUER LE SAUT SUIVANT.....	34

Chapitre 1

1. Introduction :

L'objet de la robotique est l'automatisation de systèmes mécaniques. En dotant le système de capacités de perception, d'action et de décision, l'objectif est de lui permettre d'interagir rationnellement avec son environnement, et de façon autonome. Depuis les premiers automates jusqu'aux systèmes disponibles en ce début de XXI^e siècle, on mesure tout à la fois le chemin parcouru et celui restant à parcourir avant de réaliser les rêves qui animaient les pionniers.

La robotique est un domaine de recherche qui se situe au carrefour de l'intelligence artificielle, de l'automatique, de l'informatique et de la perception par ordinateur ; cette interdisciplinarité est à l'origine d'une certaine complexité. Des applications dans des domaines aussi variés que l'industrie manufacturière, le spatial, l'automobile ou plus récemment les loisirs et le secteur médical, démontrent aujourd'hui l'intérêt économique et social de ces recherches. La robotique mobile autonome vise plus spécifiquement à concevoir des systèmes capables de se déplacer de façon autonome. Les applications directes se situent notamment dans les domaines de l'automobile, de l'exploration planétaire ou de la robotique de service par exemple. De nombreuses applications restent à découvrir, qui ne découlent pas directement des avancées de la robotique mais qui utilisent ses méthodes et ses développements. Notre thèse a pour cadre général la navigation autonome des robots mobiles, et elle se focalise sur un type de système et d'application spécifiques : les véhicules articulés avec des applications dans le transport automatique.

1.1. La robotique mobile :

Les années 70 et 80 ont vu émerger un nouveau domaine la robotique. Les premiers robots ont été développés pour la manipulation d'objets, ce qui a permis de les intégrer dans des unités de production pour réaliser des tâches répétitives, l'objectif étant de remplacer l'homme dans les tâches pénibles. Ces premiers robots étaient fixes et plongés dans un environnement parfaitement structuré. Les progrès technologiques ont conduit à la réalisation de robots tout à fait différents dans la conception et dans le type de tâches à accomplir ces robots sont devenus mobiles. Ce dernier se caractérise par l'absence de lien mécanique avec un objet de référence, ce qui ouvre la porte à de nombreuses nouvelles applications

potentielles. La liberté de mouvements leur confère une autonomie qui leur permettent de trouver d'autres utilisations que la manipulation d'objets. Les domaines applications sont légion : le transport, l'exploration en environnement hostile, le domaine agricole, le génie civil, la lutte contre les incendies ou le terrorisme, l'inspection, le domaine militaire, l'aide aux handicapés etc... Cette liste est loin d'être exhaustive et si des besoins de manipulations sont utiles, alors un manipulateur peut être monté sur une base mobile pour associer les deux types de robots et rendre le tout plus efficace.

La fonction principal du robot mobile constitue le mouvement afin de réaliser une tâche par le mouvement et par le transport d'un outil (ex: nettoyage), soit par le transport d'un objet d'un lieu vers un autre (ex: caméra de surveillance). Le mouvement d'un robot est obtenu par la propulsion et l'appui sur le milieu.

Les modes de propulsion sont variés selon le type de mobile. Cela s'étend du moteur thermique ou électrique au moteur hydraulique ou pneumatique. Cet aspect est directement lié à l'application du robot.

Le milieu dans lequel évolue le mobile oriente très fortement la mécanique mise en œuvre pour le mouvement.

Chapitre 2

2. Modélisation

2.1. Classification des types de roues

La mobilité d'un robot mobile dépend grandement du type de roues utilisé. Campion a présenté une classification des différents types de roues et configurations rencontrées en robotique mobile. Les différents types de roues que l'on rencontre sont : la roue fixe : cette roue n'autorise qu'un déplacement dans la direction de son plan médian, l'orientation n'est pas modifiable, la roue centrée orientable : elle possède un axe d'orientation en plus de l'axe de rotation, et cet axe d'orientation passe par le centre de la roue, la roue décentrée orientable ou roue folle : son axe d'orientation ne passe pas par le centre de la roue (c'est le cas par exemple des roues des chaises des bureaux)

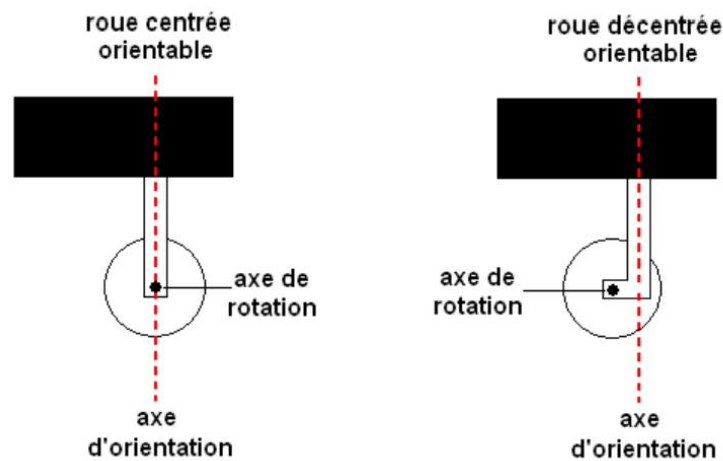


FIGURE 1 : ROUE CENTREE ET ROUE DECENTREE

En plus de ces roues classiques (Figure 1), d'autres roues ont été développées pour accroître la mobilité du robot. Elles permettent d'augmenter les capacités de déplacement dans toutes les directions du plan (Figure 2). Cependant, elles ne sont commandables que dans certaines de ces directions. Dans cette catégorie nous trouverons notamment :

- les roues suédoises : ces roues autorisent les glissements latéraux grâce à un système de galets remplaçant la bande de roulement classique, montés en inclinaison par rapport au plan de la roue. La combinaison de la rotation de la roue avec la rotation libre du galet en contact avec le sol permet le déplacement sans glissement sur le sol dans toutes les directions. Cependant le couple moteur que l'on peut transmettre à ces roues est très limité, ce qui réduit son utilisation en pratique.
- les roues tronçon-sphériques (ou orthogonal wheels) : cette structure utilisant deux roues libres en quadrature présente l'avantage de pouvoir transmettre un couple intéressant par rapport aux roues suédoises, mais elle souffre de petits problèmes de sauts au moment de la transition d'une roue support à l'autre.

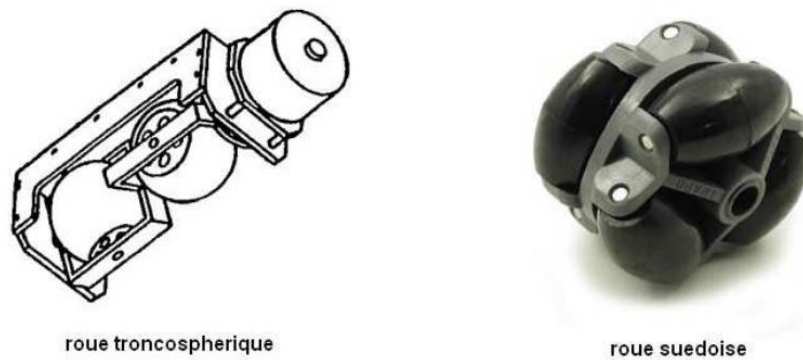


FIGURE 2 : ROUE TRONCSPHERIQUE ET ROUE SUEDOISE

2.2. Principales structures cinématiques

En associant les différents types de roues selon une structure mécanique donnée, le robot mobile disposera de plus ou moins de mobilité. Le nombre, le type et la disposition des roues engendrera ou non la contrainte de non holonomie du robot. Si on néglige les phénomènes dynamiques tel que l'inertie, un robot holonome est un robot capable à chaque instant de se déplacer dans n'importe quelle direction du plan, sans avoir à effectuer une reconfiguration de ses roues. Tout système évoluant dans un plan 2D possède 3 degrés de liberté : une translation selon l'axe x , une translation selon l'axe y et une rotation autour d'un axe z normal à (x,y) . Cependant une roue classique ne possède que 2 degrés de mobilité : elle ne peut que faire une translation (avancer ou reculer), ou une rotation sur elle-même. Elle ne peut pas déraper transversalement pour effectuer un mouvement de translation horizontal.

Cette contrainte empêche la plupart des véhicules « traditionnels » d'effectuer un déplacement instantané transversal (parallèlement à l'axe de rotation de la roue). Une voiture ne peut pas effectuer de créneau pour se garer, sans faire un certain nombre de manœuvres. C'est une contrainte que l'on retrouve sur tous les robots mobiles de type voiture ou à roues différentielles. De tels véhicules, possédant un nombre de degrés de mobilité inférieur au nombre de degrés de liberté, sont dits non-holonomes, Cette contrainte touche principalement les robots mobiles aux roues.

Les configurations non-holonomes les plus courantes sont :

- le tricycle / la voiture : ces deux structures sont constituées d'un axe fixe (généralement à l'arrière) et d'un axe directeur (Figure 3). Dans le cas du tricycle, seule une roue est présente sur l'axe directeur, contrairement à la voiture qui en possède deux. La théorie d'AckermanJeantaud donne les conditions

théoriques de non glissement et non dérapage pour les configurations de type voiture. Notamment les axes de rotation des 4 roues doivent s'intersecter en un point unique ; le Centre Instantané de Rotation. Pour cela, la vitesse de la roue extérieure doit être légèrement supérieure à celle de la roue intérieure. La structure de type voiture peut être modélisée par une structure équivalente à celle de 3 roues, ce qui revient au modèle du tricycle.

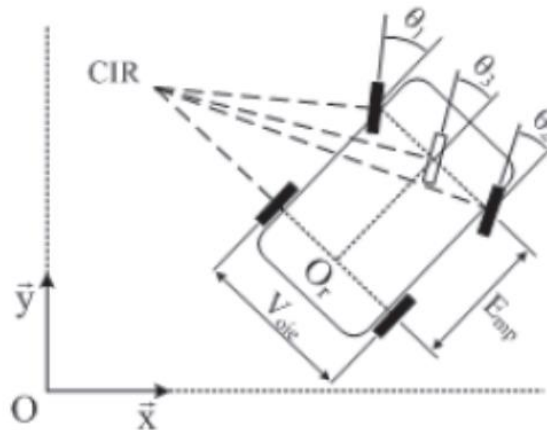


FIGURE 3 : EQUIVALENCE MODELE VOITURE ET TRICYCLE

– les roues différentielles (Figure 4) : cette structure également classique est constituée de deux roues motrices placées sur le même axe, et d'au moins un appui supplémentaire (généralement une ou deux roues folles). L'avantage de cette structure est qu'elle permet au véhicule de tourner sur place. Suivant si les vitesses de rotation des deux roues motrices sont de signe opposées ou pas. Ainsi le robot peut pivoter rapidement, ce qui donne des capacités de déplacement intéressantes. Cependant, le déplacement latéral n'étant pas directement réalisable, cette structure n'est pas non plus holonome.

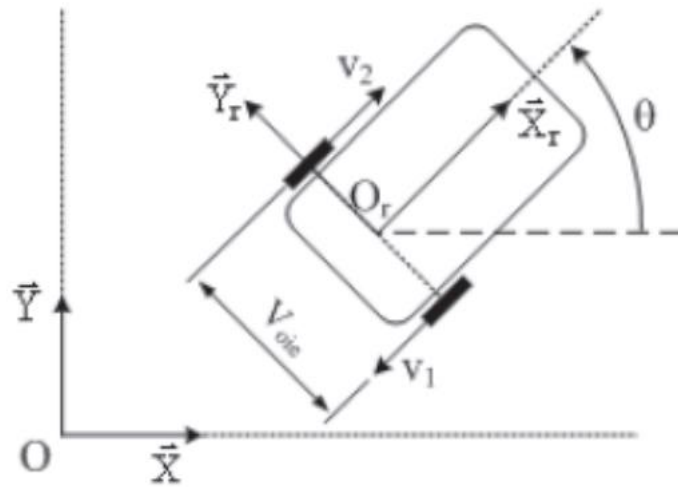


FIGURE 4 : MODELE ROUES DIFFERENTIELLES

En utilisant des roues telles que les roues suédoises ou tronco-sphériques sur des plateformes mobiles, des robots ayant la capacité de se mouvoir dans toutes les directions ont été créés. Ces robots, à 3 degrés de mobilité dits omnidirectionnels, permettent de s'affranchir de la contrainte de non holonomie. Leurs structures spécifiques leur permettent de se déplacer instantanément (à la dynamique près) dans toutes les directions en ayant n'importe quelle orientation, rendant possible le suivi de trajectoires de forme quelconque (Figure 5).

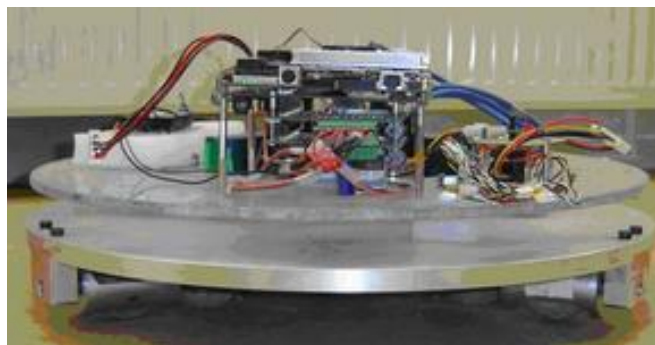


FIGURE 5: LE ROBOT OMNIDIRECTIONNEL ROMNI

Chapitre 3

3.Évitement réactif d'obstacles pour robots mobiles nonholonomes

Ce chapitre traite de la fonctionnalité d'évitement réactif d'obstacles. Tout d'abord nous présentons un état de l'art des méthodes d'évitement réactif d'obstacles, et nous montrons leur inadéquation aux spécificités de notre problématique. Ensuite nous proposons une méthode originale d'évitement réactif d'obstacles pour des robots mobiles nonholonomes. Cette méthode utilise une trajectoire de référence qui est déformée de façon à s'éloigner des obstacles et à satisfaire les contraintes cinématiques du système. Enfin nous présentons quelques optimisations de l'algorithme de cette méthode et des extensions de ses applications.

3.1. État de l'art

La plupart des méthodes d'évitement réactif d'obstacles sont des méthodes locales. En effet les méthodes globales, qui considèrent un modèle complet de l'environnement, se ramènent au problème de la planification de mouvement. Et pour des systèmes possédant de nombreux degrés de liberté (plus de 3), la complexité de la planification d'une trajectoire interdit son utilisation en cours d'exécution. Par exemple l'approche de replanification rapide D^* de Stentz en 1994 découpe l'espace de travail en cellules étiquetées «libre» ou «occupée», pour diminuer la complexité.

Pour les systèmes considérés dans notre étude, la planification d'une trajectoire sans collision dans un environnement fortement contraint est très coûteuse en temps de calcul. Le modèle de l'environnement utilisé est construit auparavant par un autre robot ou à partir d'un plan métrique. On emploie une méthode de guidage (calcul d'une trajectoire admissible entre deux configurations sans prise en compte des obstacles) pour le système considéré, si elle existe, pour connecter les configurations tirées aléatoirement. On supposera donc toujours que l'on possède une trajectoire de référence calculée dans un modèle de l'environnement. Parmi les méthodes locales d'évitement réactif d'obstacles, on peut distinguer deux catégories : les méthodes utilisant une trajectoire de référence et celles n'en utilisant pas. Le type de système et d'application orientent le choix vers une catégorie ou l'autre. Enfin un autre type de méthodes doit être mentionné, qui tient compte explicitement de la vitesse estimée des obstacles pour réaliser un mouvement sans collision, ce qui définit une nouvelle problématique. Nous renvoyons

à Fraichard 2004 pour un cadre formel et un état de l'art de cette problématique. Dans notre étude nous considérons que les obstacles sont statiques.

3.2. Méthodes d'évitement réactif d'obstacles sans trajectoire de référence

3.2.1. Champs de potentiel

Les méthodes de champs de potentiel pour la navigation en robotique, initialement proposées par [Khatib 1986] pour un bras manipulateur, consistent à construire une fonction de potentiel qui résume les objectifs de la navigation : éviter les obstacles (potentiel répulsif) et atteindre une configuration but (potentiel attractif). À chaque position du robot, une «force» résultant de l'action conjuguée des obstacles et du but est calculée, qui correspond à une direction à suivre par le robot. De nombreuses adaptations de cette technique ont été proposées. On peut par exemple citer Borenstein en 1991, qui calcule une direction de mouvement à partir d'informations proximétriques. Ces méthodes purement réactives sont sujettes à des minima locaux, et peuvent nécessiter une replanification globale. Par ailleurs, elles peuvent entraîner un mouvement oscillatoire du robot dans certaines situations (des passages étroits par exemple).

3.2.2. Nearness Diagram

Cette approche proposée par Minguez en 2000 repose sur un diagramme de proximité des obstacles, mis à jour au fur et à mesure du déplacement du robot. En fonction de l'allure de ce diagramme (nombre de passages sans obstacles, taille des passages, etc.), un comportement adapté (exploration, avancement vers le but, retour en arrière, etc.) est sélectionné. La direction de mouvement la plus prometteuse par rapport au but est alors choisie. Cette méthode a été appliquée à un véhicule de type uni cycle grâce à une expression des obstacles dans l'espace des points accessibles en un mouvement élémentaire (un arc de cercle en l'occurrence), obtenue par une transformation dans l'espace dit «Ego-cinématique» Minguez en 2002. L'exposé de ces différentes méthodes et de leurs hypothèses fait immédiatement apparaître leur inadéquation à des systèmes à cinématique plus complexe évoluant dans des environnements fortement contraints. On ne peut en effet généralement pas présupposer de la forme des trajectoires. De même les discrétisations en grille de l'espace de travail, proposées par certaines méthodes, ne permettent pas les mouvements arbitrairement proches des obstacles.

3.3. Méthodes d'évitement réactif d'obstacles avec une trajectoire de référence

Dès lors que la cinématique du système est plus complexe, le calcul préalable d'une trajectoire de référence qui sera adaptée lors de l'exécution s'avère nécessaire. Nous présentons deux méthodes utilisant une trajectoire de référence pour l'évitement réactif d'obstacles.

3.3.1. Trajectoires d'évitement

Une méthode proposée dans [Laugier 1999] repose sur l'enchaînement de trajectoires élémentaires (Sensor Based Maneuvers SMB), dans le contexte d'une voiture automatisée roulant sur une route. Les SMB sont par exemple le suivi d'un couloir de route, le changement de file ou le parking parallèle. Les paramètres de chacune de ces modalités sont définis par les informations sensorielles. Un contrôleur, amélioré dans [Large 2000] par un réseau de neurones, permet d'enchaîner ces différentes modalités.

3.3.2. Bande élastique

La méthode proposée par Quinlan en 1993 utilise une trajectoire initialement planifiée qui est représentée par une série de boules adjacentes (appelées «bulles») dans l'espace des configurations. Le rayon d'une boule centrée en une configuration est la distance de cette configuration à l'obstacle le plus proche. Ainsi une trajectoire est sans collision dès lors que les «bulles» qui la composent se recouvrent. Développée pour des systèmes sans contraintes cinématiques, cette technique considère la trajectoire comme une bande élastique, se modifiant sous l'action de forces répulsives générées par les obstacles, et de forces internes de contraction ou d'élasticité. Du fait de la représentation en bulles, la mise à jour de la trajectoire sous l'action des forces est très rapide.

Cette technique a été étendue à un robot de type voiture par Khatib en 1997. La «forme» des «bulles» est ici donnée par la métrique des trajectoires de Reeds et Shepp (combinaisons d'arcs de cercle et de lignes droites) Soueres en 1996. Le lissage de la courbe joignant les centres des «bulles» est réalisé en utilisant une courbe de Bézier. Mais pour certains systèmes on ne connaît pas la plus courte distance entre une configuration et un obstacle. La méthode de la bande élastique ne peut donc s'appliquer qu'au prix d'approximations sur la forme des trajectoires de tels systèmes, ce qui rend impossible la navigation en environnement très contraint.

L'ensemble de ces considérations, concernant les limitations des méthodes utilisant une trajectoire de référence ou des techniques proposées pour les systèmes à cinématique plus simple, motive le développement d'une méthode générique d'évitement d'obstacles pour systèmes nonholonomes.

Chapitre 4

4. Les algorithmes existants :

4.1. Introduction :

Dans le cadre de de notre projet tutoriel qui consiste à concevoir un système de détection de trajectoire robotique, nous établissons l'état de l'art de la détection de trajectoire robotique.

En effet depuis quelque décennie de nombreux chercheurs cherchent à permettre aux robots de devenir plus autonome, il existe de nombreux algorithmes très différents les uns des autres qui permettent aux robots d'avoir une mobilité plus ou moins élaborée. Dans ce rapport nous définissons ce qui est pour nous la robotique mobile puis nous établirons la liste des algorithmes existants venant du résultat de nos recherches, enfin nous concluons sur les améliorations à venir.

4.2. Méthode par décomposition de l'environnement en cellule.

Une première méthode de planification de trajectoire consiste à décomposer l'environnement du robot en cellules Latombe en 1991 (en un ensemble de régions connexes adjacentes). Il suffit ensuite de trouver un algorithme travaillant sur la discrétisation de l'environnement. Là, différentes techniques existent, par exemple, la partition de Voronoï Choset en 1996 ou les graphes de visibilité Chazelle et Guibas en 1989.

Dans ces cas, l'environnement discrétisé est représenté dans un graphe et trouver une trajectoire revient à chercher un chemin dans un graphe [Shin et McKay 1986].

D'autres algorithmes de planification travaillent dans un environnement discrétisé. Parmi les plus connus, on peut citer A* [Hart 1968] qui planifie une trajectoire optimale connaissant l'environnement ou D* Stentz et Anthony 1994 qui planifie une trajectoire dynamiquement, ce qui lui permet de découvrir son environnement au fur et à mesure.

Les avantages de tels algorithmes sont entre autre la facilité d'implémentation et de représentation. On travaille sur une grille et chaque case est en 8-connexité. C'est-à-dire que d'une case, on a que 8 choix de déplacement possible.

Par contre, ces algorithmes possèdent de gros défauts. Notamment le fait que les contraintes cinématiques des robots ne sont pas prises en compte (accélération, vitesse, vitesse angulaire maximale, encombrement etc.), ce qui autorise le planificateur à trouver une trajectoire qui ne sera pas exécutable par le robot. De plus, le fait de discrétiser l'espace de recherche limite beaucoup le nombre de trajectoire possible.

4.3. La méthode des champs de potentiel

Les champs de potentiel Khatib en 1986 est une méthode est assez originale qui assimile le robot à une particule soumise à un champ de forces répulsives et attractives.

Un obstacle génère un champ de potentiel répulsif tandis que l'objectif à atteindre génère un champ de potentiel attractif. L'algorithme calcul donc un vecteur résultant qui indiquera au robot comment effectuer son déplacement.

Cet algorithme est totalement réactif et peut donc être très facilement implémenté en temps réel. Cependant, comme l'environnement est très peu souvent totalement convexe, cette méthode entraine facilement le robot dans des minima locaux. De plus, les problèmes d'oscillation peuvent, dans certain cas, être constatés. Ici non plus, on ne tient pas compte des contraintes du robot.

4.4. La méthode de la fenêtre dynamique

Cette méthode Fox en 1997 travaille dans l'espace de commande du robot. La méthode calcule les vitesses possibles du robot pour que celui-ci ne rentre en collision avec aucun obstacle. Une fois l'espace de recherche des vitesses calculées, l'algorithme trouvera la commande optimale à envoyer au robot en minimisant une fonction de coût sur ce domaine de recherche. (Minimisation du temps de parcourt, de l'énergie dépensée, maximisation de la vitesse etc.)

Cette méthode un peu plus abstraite permet de prendre en compte les contraintes cinétiques du robot. Par contre, elle manque en flexibilité, ce qui rend difficile son implémentation dans un cadre multi-robot.

4.5. La méthode de la bande élastique

Une méthode Quinlan en 1994 qui tend une "bande élastique" entre le robot et l'objectif. Cette bande étant capable de se déformer en présence d'un obstacle, celle-ci génère donc une trajectoire envisageable par le robot.

4.6. La planification par logique floue

Cette méthode de planification se base sur la logique floue Zadeh en 1965 : Chaque grandeur physique (par exemple une distance d'un obstacle) est convertie en une variable linguistique (petit, moyen ou grand par exemple). Alors qu'en logique booléenne classique, une distance serait soit petite, soit grande, la logique floue autorise une distance à être à la fois grande et petite (20% petite et 80% grande) suivant une fonction d'appartenance précise.

La planification floue va donc raisonner non pas sur des grandeurs physiques, mais sur des variables linguistiques. On appliquera des règles précises sur ces variables. Finalement, on obtiendra la commande sous la forme de variables linguistiques de l'on reconvertira en grandeurs physiques (lors de la défuzzification).

Cette approche permet de ne pas raisonner au niveau de la grandeur physique, mais plus comme un humain le ferait. Exemple : Si la distance au feu est faible et le feu est rouge, je freine fort. Ici, on ne raisonne pas sur la valeur de la distance directement. (On ne va pas se dire, dans la vie de tous les jours : *si je suis à moins de 6mètre du feu, alors, je fais telle action*. Un humain raisonnera plutôt comme ça : *si je suis moyennement proche, si je suis assez proche, si je suis très proche alors, je fais telle action*)

La logique floue permet donc d'obtenir des trajectoires progressives qui évitent les obstacles.

4.7. La planification par juxtaposition de splines polynomiales

L'algorithme de Michael D'effort Defoort 2007, 2009 génère des bouts de splines polynomiales qui respectent les contraintes cinématiques du robot. On peut simplifier l'algorithme en le forçant à générer non pas des splines, mais simplement des polynômes. La trajectoire du robot est alors assimilée à la juxtaposition de plusieurs polynômes.

Un bout de trajectoire est alors représenté par un système paramétré de deux polynômes du troisième degré. Les coefficients des degrés zéro et un fixent la position initiale et la vitesse initiale du robot (ce

qui permet de garder la continuité entre les différents polynômes). Ainsi, il ne reste plus qu'à optimiser les coefficients du second et du troisième degré du polynôme à l'aide d'algorithmes d'optimisation comme CFSQP [Lawrence, Zhou et Tits].

Cette méthode permet donc de respecter les contraintes cinématiques du robot. De plus, l'optimisation des paramètres des polynômes peut prendre en compte des fonctions de coût, ce qui permet de choisir selon quelle critère l'on décide qu'une trajectoire est optimale ou non.

4.8. DKP : Deterministic Kinodynamic Planning

Cette méthode de planification Gaillard 2009 est assez complexe, mais permet de respecter toutes les contraintes du robot et d'éviter les problèmes des minimas locaux.

DKP peut être décomposé en deux parties. La première planifie localement des bouts de trajectoires polynomiales (d'ordre deux) sur différents temps de parcours. Pour trouver un bout de trajectoire optimal, il suffit de chercher une solution optimale dans l'espace des paramètres des polynômes via une résolution géométrique des contraintes. Avec ce planificateur local, on construit un arbre de solutions locales.

La seconde partie de l'algorithme est un planificateur global basé sur un A*-like qui va choisir les bouts de trajectoire à adopter pour résoudre le problème.

L'avantage de cet algorithme est qu'il génère une grande diversité de bouts de polynôme, ce qui permet de trouver une solution même dans les environnements très complexes.

Chapitre 5

5. Le Robot RP6

5.1. Introduction

Le RP6 est un système de robot mobile conçu dans le but d'introduire à l'utilisateur du monde de la robotique. Le produit est un système complètement assemblé, qui comprend un microcontrôleur ATMEGA32 et une grande variété de capteurs pour interagir. Le robot est l'appareil parfait pour la recherche et le développement de nouvelles fonctionnalités depuis l'un de ses principaux points est la grande possibilité d'expansion. Cela signifie qu'il est prêt pour l'ajout de nouvelles les modules et

permettent à l'utilisateur d'interagir avec eux en utilisant un large éventail de connexions. Avec I2 C bus est possible d'ajouter jusqu'à 127 périphériques et il comprend également un bus d'extension qui aide à connecter des périphériques externes au microcontrôleur.

5.2. Caractéristiques principales

- Atmel ATMEGA32 8
- Bit Microcontroller
- Flexible expansion system, based on the I2C
- Bus
- Symmetrical mounting possibilities for expansion modules at front and rear
- USB PC Interface for program uploads from PC to microcontroller
- Powerful caterpillar drive unit in combination with a new gearing system for minimising noise
- Two powerful 7.2V DC-Motors
- Two powerful MOSFET6 Motor-drivers (H-Bridges)
- Two high resolution encoders for speed- and motion-control
- Anti-collision-system (ACS) with an integrated IR7 receiver and two IR diodes aligned to left and right
- Infrared Communication-system (IRCOMM)
- Two light sensors
- Two bumper sensors for collision detection
- 6 Status LED's for sensor and program status displays
- Two free Analogue/Digital Converter (ADC) channels for external sensor systems
- Accurate 5V voltage regulation (maximum current supply of 1.5A)

- Replaceable 2.5A fuse
- Low standby current of less than 5mA (4mA typ. and ca. 17 up to 40mA in use)
- Power supply with 6 NiMh8 accumulator batteries.
- The main board provides 6 small expansion areas

5.3. Caractéristiques détaillées:

Ce chapitre décrit les systèmes utilisés dans le projet en détail.

Dans la figure 6 montre un aperçu général des systèmes de robots et de ses principales connexions.

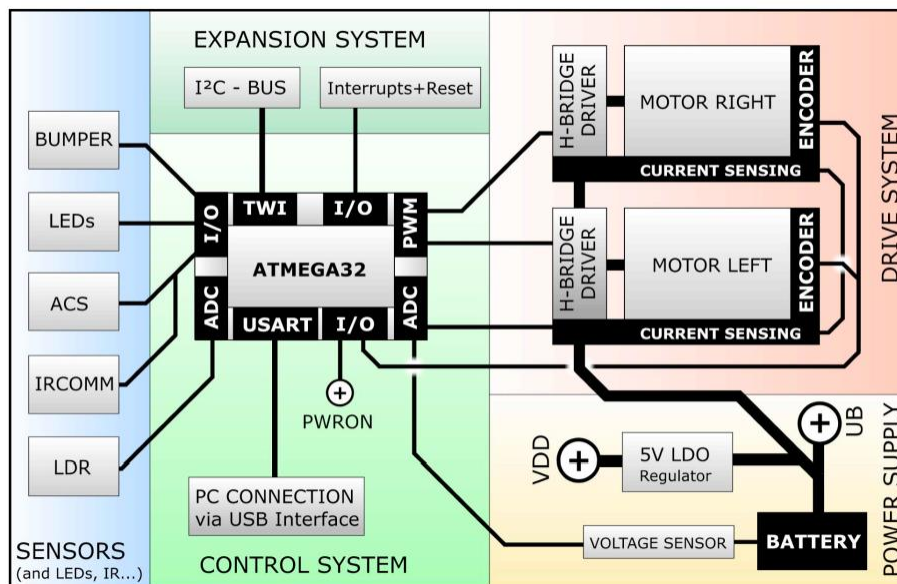


FIGURE 6 : OVERVIEW SCHEMATIC OF THE RP6 ROBOT SYSTEM

Toutes les informations dans ce chapitre, y compris des schémas et des fiches est tiré de [Figure 6] et est inclus dans le CD joint.

5.4. Microcontrôleur

Le robot utilise un microcontrôleur Atmega32 fabriqué par Atmel [2].

Il est un microcontrôleur 8 bits qui a une mémoire 32KB (32768 octets) Flash ROM et 2KB (2048 octets) de mémoire RAM. Il fonctionne à une fréquence de 8 MHz mais elle est capable de travailler avec une horloge de 16 MHz. Le système utilise la 8MHz fréquence d'horloge pour des raisons

d'économie d'énergie. Le microcontrôleur comporte deux modules de communication: l'UART9 pour la communication série avec l'ordinateur et le TWI-module utilisé pour l'I2 C bus d'extension.

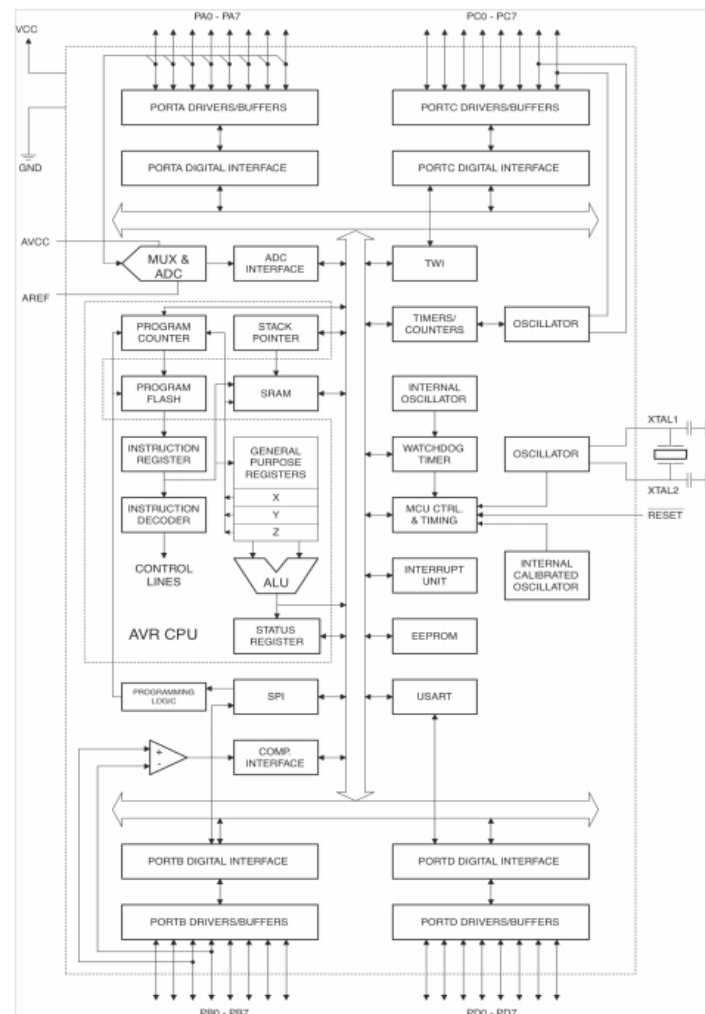


FIGURE 7 : ATMEGA32 SCHEMA

La présente fiche technique avec les caractéristiques complètes est incluse dans le CD joint.

5.5. Détecteurs

5.5.1. Système Anti Collision (ACS)

Le détecteur le plus complexe d'un point de vue logiciel est l'ACS – le système anti collision (en anglais: Anti-Collision System). Il est composé d'un récepteur IR (voir photo) et de deux LED IR montées à gauche et à droite à l'avant de la platine de détection. Les LED IR sont directement commandées par le

microcontrôleur. Vous pouvez librement modifier et améliorer les routines de pilotage. Sur le modèle précédent, il fallait pour cela un autre contrôleur dont le programme ne pouvait pas être modifié par l'utilisateur...

Les LED IR envoient des impulsions infrarouges courtes modulées à 36kHz auxquelles le récepteur IR réagit. Si les impulsions IR sont réfléchies par un objet devant le robot et détectées par le récepteur IR, le microcontrôleur peut réagir et p.ex. Engager une manœuvre de contournement. Afin d'éviter une réaction trop rapide de l'ACS due à des interférences, le logiciel attend un certain nombre d'impulsions sur une durée déterminée.

Il exécute également une synchronisation avec la réception RC5 afin de ne pas réagir aux signaux RC5 de télécommandes TV et Hifi. Cette garantie n'existe cependant pas pour d'autres codes et il est possible que l'ACS détecte des obstacles là où il n'y en a pas! Puisqu'il y a une LED gauche et une LED droite, l'ACS arrive à différencier grossièrement si l'objet se trouve à droite, à gauche ou au milieu devant le robot. Par ailleurs, vous pouvez régler l'intensité du courant des impulsions envoyées aux deux LED IR en 3 niveaux. Même au niveau le plus élevé, l'ACS n'est pas toujours fiable car la nature de la surface de l'objet joue un rôle important. Un objet noir réfléchit la lumière IR beaucoup plus mal qu'un objet blanc et un objet angulaire et réfléchissant renverrait la lumière IR surtout dans une certaine direction. La portée dépend donc toujours de l'objet en question. C'est une faiblesse commune à quasiment tous les détecteurs IR (en tout cas dans notre catégorie de prix).



FIGURE 8 : SCHEMA ANTI-COLLISION

5.5.2. Détecteurs de Pare-chocs (Bumper)

Deux micro commutateurs avec un long levier chacun sont montés sur le devant du robot sur une platine à part. Cela protège les LED IR sur la platine qui ne se déforment plus si facilement lorsque le robot heurte un obstacle. Les deux commutateurs permettent au microcontrôleur d'enregistrer le choc et de reculer, tourner un peu et continuer sa route. Les commutateurs sont connectés sur deux des ports qui sont déjà reliés aux LED et n'occupent ainsi pas de port supplémentaire. C'est pourquoi les LED s'allument toujours lorsqu'un commutateur est appuyé. Comme cela arrive relativement rarement,

cela ne gêne en rien. Il est également possible de démonter le pare-chocs et le remplacer par un dispositif de tir ou de ramassage de balles ou similaire.

Chapitre 6

6. L'environnement de travail

6.1. Software

L'objectif principal du projet est de développer le logiciel nécessaire pour gérer le robot.

Ce chapitre décrit toutes les ressources logicielles utilisées pour la création de logiciel de robot et le processus ultérieur de la programmation du microcontrôleur. Tous les programmes sont exécutés sous Windows XP Professional Version 2002 SP3.

6.2. GCC: GNU Compiler Collection

Le CCG abréviation signifie GNU¹ C Compiler. Quand il a été créé, il ne comportait C langue, mais de nos jours, d'autres langages comme Ada, Java ou Fortran ont été ajoutés au compilateur. La GNU projet² est un système d'exploitation créé par les utilisateurs et sa caractéristique principale est qu'il est totalement gratuit. Il a été lancé en 1984 par Richard Stallman et aujourd'hui, il est toujours en cours développé. Le GCC est le compilateur créé pour ce système d'exploitation. La première version du compilateur a été libérée le 22 Mars 1987, et la plus récente est de la 25 de Mars de l'an³ actuelle. Le compilateur GCC est distribué par Free Software Foundation (FSF) ⁴. Les bibliothèques et les archives du compilateur complets et les manuels peuvent être trouvés à <http://gcc.gnu.org/>

¹GNU: GNU is Not Unix

²more info in <http://www.gnu.org/>

³<http://gcc.gnu.org/releases.html>

⁴<http://www.fsf.org/>¹

6.3. Programmeur Notepad

Le Notepad du programmeur est un logiciel gratuit code de licence editor. Il prend en charge le langage C avec la syntaxe mettant en évidence et il dispose d'une interface MDI à onglets très utile.

Les principales caractéristiques du programme peuvent être trouvées dans le web site développeur.

L'élection de cet éditeur de code de texte a été faite parce qu'il est facile à utiliser et en même temps est un outil puissant pour le développement de logiciels. Il est aussi le programme recommandé par le fabricant du robot.



FIGURE 9 : ABOUT OF PROGRAMMER'S NOTEPAD 2

6.4. RP6 Loader

Il est un programme développé par Arexx Ingénierie et il est utilisé pour télécharger les fichiers .hex généré avec l'éditeur de code pour le microcontrôleur sur le RP6. Il ne faut pas l'installer sur le PC car il est écrit en Java. Le programme comprend un terminal pour contrôler les données reçues par le robot. Il dispose d'une option qui permet à l'utilisateur de vérifier l'état de la mémoire et la valeur de chaque octet dans la mémoire morte.

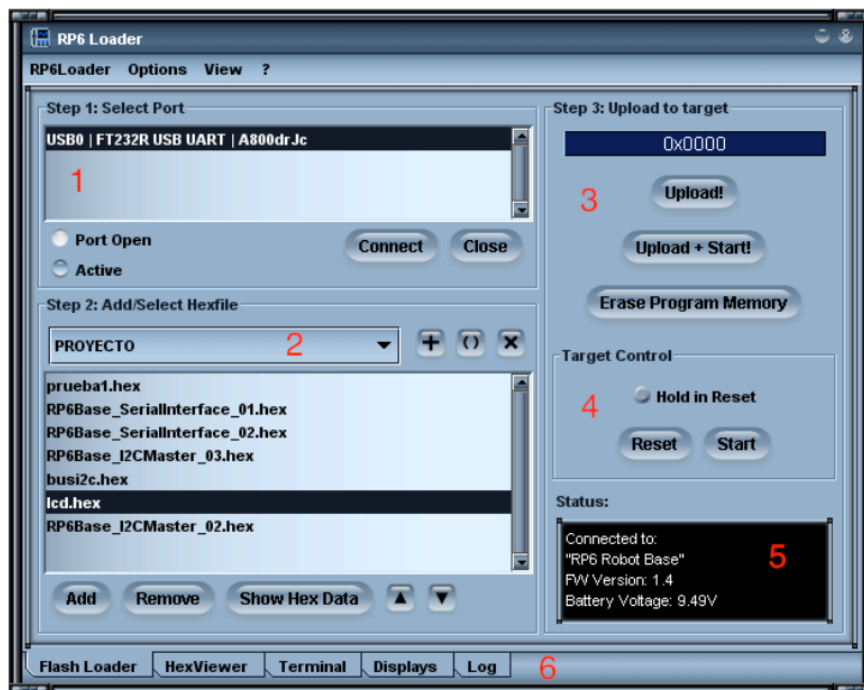


FIGURE 10 : RP6 LOADER SCREENSHOT

La figure 9 montre une capture d'écran du RP6 Loader. Les parties principales du programme sont:

- 1- boîte de sélection pour choisir la méthode de connexion au robot
- 2- Sélection du fichier .hex qui va être envoyé au robot.
- 3- Ajoutez le fichier sélectionné.
- 4- Il est possible de démarrer et d'arrêter le robot à distance.
- 5- Cette boîte indique l'état du robot. Il comprend la version du micro logiciel et de la tension de la batterie mise à jour.
- 6- Tabs pour sélectionner le terminal et le visualiser de mémoire (HexViewer).

Chapitre 7

7. Système de Commande

7.1 Le système d'évitement d'objets par ultrasons

Il existe de nombreuses idées et stratégies documentées pour éviter les objets et les walls²⁶. La plupart des robots contemporains avec qui emploient objectif au moins 3 capteurs différents afin d'obtenir des données de tous les côtés du robot et de décider ce qu'il faut faire. Il existe également des projets qui montent un capteur ultrasonique unique mais elles comprennent un servomoteur pour faire fonctionner le capteur d'un système radar. Le budget de ce projet est limité et l'installation de plus d'un capteur ou servomoteurs dépasserait le budget. La solution adoptée consiste à utiliser un capteur à ultrasons de haute précision fixe avec un motif de faisceau large qui prend des mesures dans différentes positions afin de calculer le bon chemin à suivre.

7.2. Les recherches actuelles sur les stratégies d'évitement objet

Dans les dernières années, l'une des techniques les plus populaires utilisés dans les systèmes objet d'évitement est appelée méthode de champ de potentiel ou PFM que les modèles du robot et son comportement de l'environnement à l'aide des flèches sur le terrain. Il est une technique qui a été développée dans les années quatre-vingt, mais il est encore en usage bien qu'il y ait quelques recherches qui exposent sa limitations. Une autre technique populaire travaille avec la logique floue, qui consiste à considérer plus de deux états possibles, l'introduction de valeurs entre la logique 0 et 1. Dans cet exemple précis pris dans le document référencé dans [8], le robot utilise un anneau de 24 capteurs à ultrasons pour prendre des mesures. La méthode est basée sur ayant une période d'apprentissage dans lequel analyse le robot et magasins divers facteurs ENVIRONNEMENTAL (Figure 10), qui permet au robot de trouver facilement et rapidement son chemin vers un objectif prédéfini (figure 11)

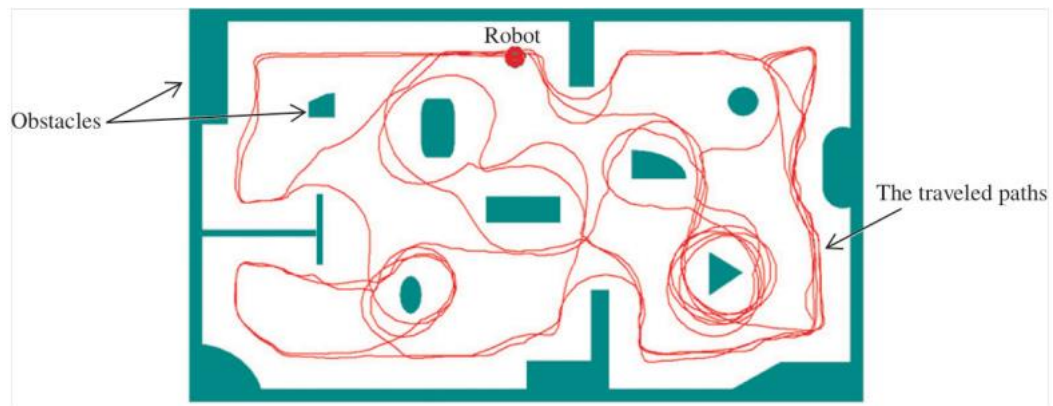


FIGURE 11 : LEARNING PHASE

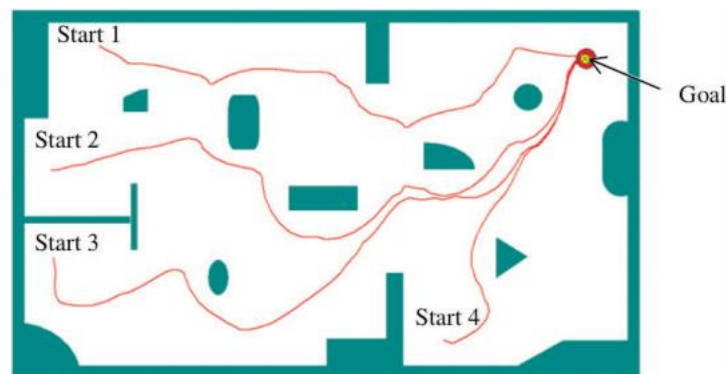


FIGURE 12 : GOAL SEEKING

7.3. Time of flight

Les valeurs générées par le capteur à ultrasons sont obtenus en utilisant une technique appelée temps de vol (TOF).

Cette technique est basée sur la formule élémentaire de la physique qui stipule que la distance est égale à la vitesse multipliée par le temps (**distance = vitesse x temps**).

In order to calculate the time, the system has an emitter that sends an energy wave. When this wave hits a surface it is reflected and comes back to the receiver. The time the wave spends in this process is measured.

L'autre facteur dans l'équation est la vitesse. Dans ce cas particulier, il est la vitesse de propagation de l'onde que l'émetteur envoie. Cette valeur dépend du support de transmission, qui dans ce cas est l'air. Après avoir obtenu ces valeurs calcul de la distance est facile.

Systèmes de TOF peuvent avoir les sources d'erreur suivantes:

- Les variations de la vitesse de propagation: la vitesse des ondes acoustiques est influencée par les variations de la température et de l'humidité.
- Les incertitudes sur le calcul du temps de Voyage de l'onde: peut-être causée par la réflectivité différente des surfaces mesurées ou par l'atténuation du signal à l distance.
- Imprécisions dans le circuit qui mesure le temps de vol: le circuit doit être plus rapide que le temps de la vague prend à revenir, donc le plus près de la cible est plus le circuit doit être.
- L'interaction de surface: si l'angle d'incidence des ondes envoyées à la surface dépasse une valeur déterminée, la réflexion de l'onde ne peut pas revenir à la position où le récepteur est.

7.4. Le choix des capteurs à ultrasons

Le SRF05 est un module ultrason fabriqué par Devantech. Extrêmement simple d'utilisation, il peut être mis en place très rapidement dans un robot. Contrairement aux SRF02, celui-ci communique avec un protocole TTL, rendant impossible un mise en série de ces modules.

Il peut être utilisé pour la détection d'obstacles autour du robot, ainsi que pour déterminer à quelle distance ces derniers se trouvent.

Le SRF-05 est une évolution du SRF-04. Ainsi la portée passe de 3 à 4 mètres. Un nouveau mode de fonctionnement a été également introduit sur le SRF-05 : l'envoi de l'ordre (trigger) et la réponse (echo) peuvent être véhiculés sur un seul fil. Dans le mode de fonctionnement classique (echo et trigger séparés), le SRF-05 est compatible avec le SRF-04.

Selon la FAQ du fabricant, les deux capteurs ont des motifs de faisceau similaires, mais le faisceau SRF05 est plus large que le SRF02 ce qui est un facteur déterminant puisque le robot utilise un seul capteur et le plus large du faisceau est plus il est d'accomplir la objectif.

Cette différence de faisceau large entre les deux capteurs peut être vue dans les images ci-dessous.

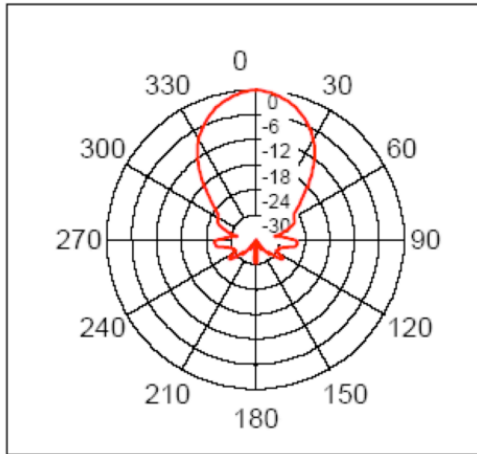


FIGURE 13 : SRF02 BEAM PATTERN

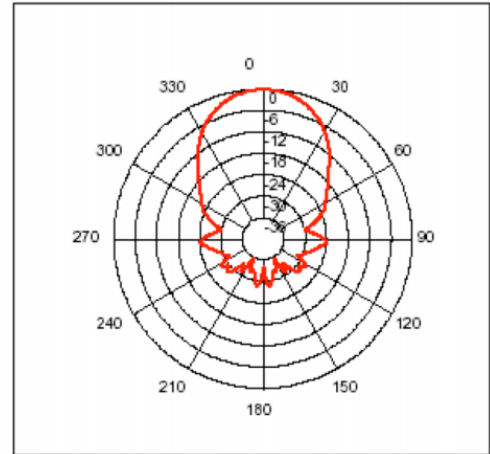


FIGURE 14 : SRF05 BEAM PATTERN

According to the results of the comparison the ultrasonic sensor chosen was the SRF05.



FIGURE 15 : CAPTURES A ULTRASOUNDS

7.5. Mode Trigger et Echo sur un même pin

Dans ce mode, la demande et la réponse se fera sur un même pin. Cela permet, entre autres, d'économiser le nombre de pattes utilisées sur le microcontrôleur. Pour activer ce mode de fonctionnement, le pin Mode doit être connecté à la masse.

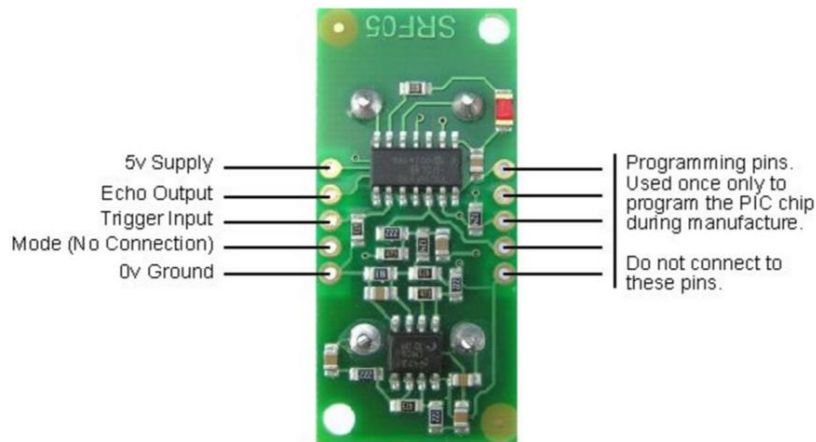


FIGURE 16 : SCHEMA SRF05

7.6. Stratégie pour la prévention de l'objet

L'exigence de base est que le robot doit utiliser un seul capteur à ultrasons. La position choisie pour le capteur dans la structure du robot se trouve au milieu de la face avant du robot.

La stratégie repose sur deux mesures faisant en différents points et ensuite calculer l'angle d'échappement en utilisant les valeurs obtenues.

Le capteur à ultrasons prend des mesures tout le temps et lorsque la distance obtenue (Segment OB) est inférieure à la distance d'arrêt sélectionné, le robot arrête (point n ° 1 de la figure 16). Une fois que le robot est arrêté (point 2), il tourne 45° à sa droite et prend la distance à la paroi (mesure droite, le segment OC). Il revient à sa position initiale, se tourne 45° à gauche et prend à nouveau la distance à la paroi (mesure de gauche, le segment OA). Ensuite, le microcontrôleur calcule l'angle d'échappement et le robot fait pivoter la valeur de l'angle obtenu et continue vers l'avant dans une voie parallèle à la paroi (point 3).

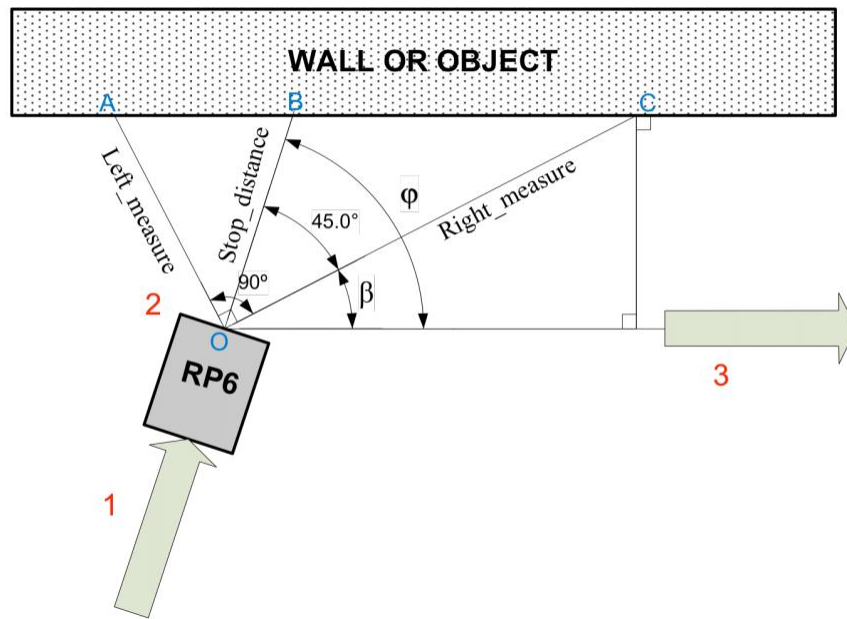


FIGURE 17 : OBJECTS AVOIDANCE STRATEGY

Le calcul de l'angle d'échappement est basé sur la trigonométrie d'un triangle rectangle.

The triangles are shown in (Figure 17) and the angles in (Figure 16)

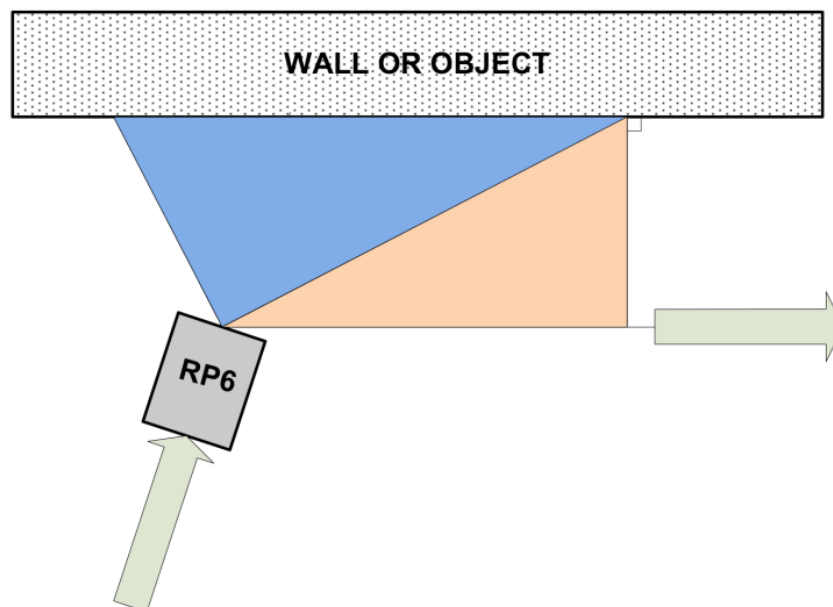


FIGURE 18 : ESCAPE ANGLE CALCULASSIONS

Le premier triangle est formé par la paroi, la mesure à gauche et la bonne mesure. L'angle de α de la figure 18 peut être calculé en appliquant la formule suivante:

$$\tan \alpha = \frac{\text{right_measure}}{\text{left_measure}} \quad [1]$$

Si l'angle est obtenu à partir de [1]:

$$\alpha = \arctan \frac{\text{right_measure}}{\text{left_measure}} \quad [2]$$

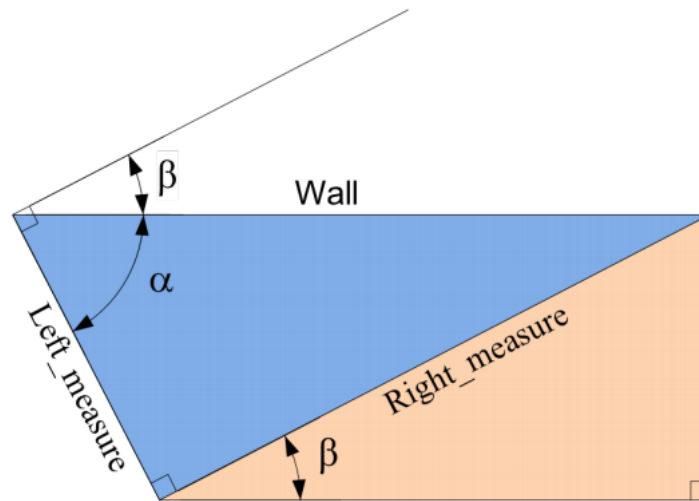


FIGURE 19 : ESCAPE ANGLE CALCULASSIONS

De l'angle droit et la valeur de α dans [figure 18], les β angle peuvent être engagées:

$$\beta = 90^\circ - \alpha \quad [3]$$

Avec les données obtenues dans [3], il est possible d'obtenir la valeur de φ (l'angle du robot doit tourner afin de suivre un chemin relativement parallèle au mur), comme indiqué dans la (Figure 16):

$$\varphi = 45 + \beta \quad [4]$$

Combiner [2], [3] et [4] l'équation complète qui obtient l'angle de fuite est :

$$\varphi = 135 - \arctan \frac{right_measure}{left_measure}$$

Annexe

Exemples de robots actuels:

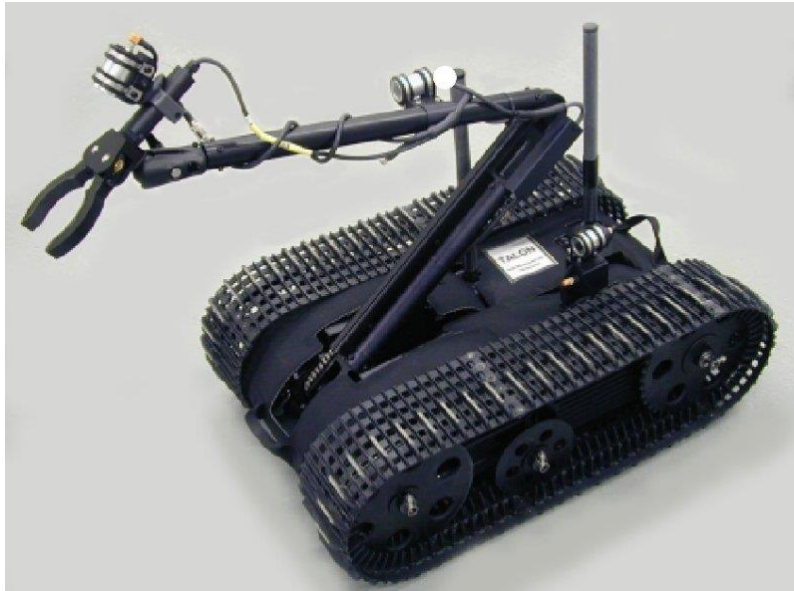


FIGURE 20 : ROBOT MILITAIRE DE RECONNAISSANCE ET DETECTION D'ENGIN EXPLOSIF, IL SE DEPLACE A L'AIDE DE DEUX CHENILLES COMME UN TANK.



FIGURE 21 : ROBOT FUTURISTE CONSTITUE D'UN CORPS SPHERIQUE TOURNANT SUR LUI-MEME POUR AVANCER, IMAGINE POUR STAR WARS.



FIGURE 22 : ROBOT FRANÇAIS SE DEPLACENT L'AIDE DE 3 ROUE, C ROBOT EST DESTINE A INTEGRER LES FOYERS FAMILIAUX IL EST DOTE D'UNE REMARQUABLE INTELLIGENCE ARTIFICIEL.

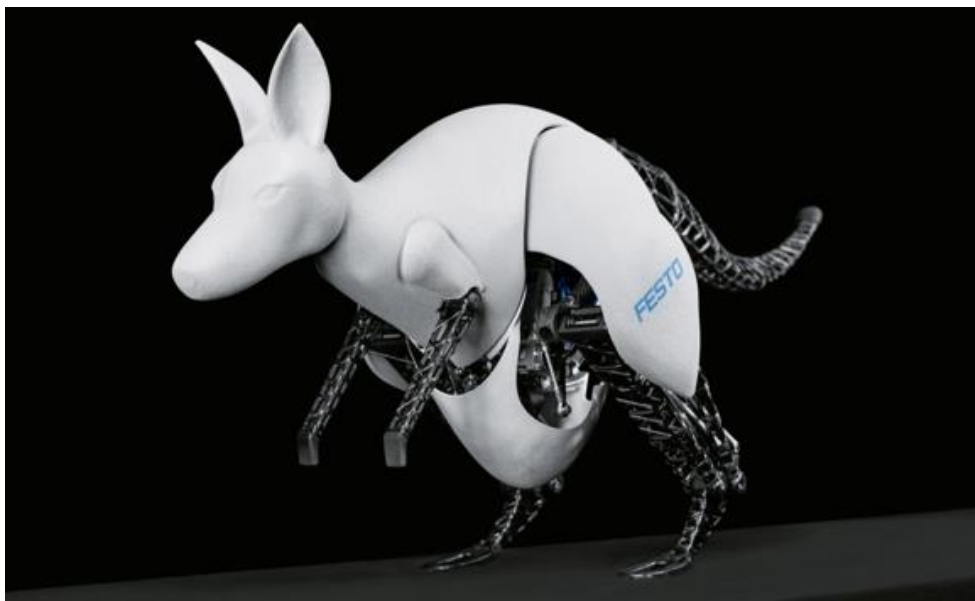


FIGURE 23 : ROBOT IMITANT LE DEPLACEMENT D'UN KANGOUROU. VERITABLE BIJOUX D'INGENIERIE CE ROBOT EST TOTALEMENT AUTONOME ENERGIE CAR IL REUTILISE L'ENERGIE CINETIQUE DU DEPLACEMENT PRECEDENT POUR EFFECTUER LE SAUT SUIVANT.

Conclusion :

Les robots mobiles sont des bijoux de technologie car ils mettent en relation de nombreux principes physiques électroniques tous cela régis par une programmation informatique rigoureuse. Nous avons conscience que c'est techniques ont été développées en plusieurs années. Nous sommes en mesure de travailler sur un algorithme défini et d'implémenter toutes les solutions relatives à l'algorithme.

Les implémentations sont diverses et se font en fonction de l'environnement où le robot va évoluer, par exemple un robot marin aura besoin d'hélice pour se mouvoir alors qu'un robot au sol a besoin de roue ou de jambes, ainsi que des ailes pour un robot volant.

Les domaines d'application des robots mobiles sont très impressionnants et ne cessent de s'accroître. Il existe de nombreux modèles de drones utilisés pour faire du repérage dans le domaine civil et militaire mais aussi dans le cinéma pour offrir des prises de vue uniques. Les robots terrestres ont une multitude d'utilité et ne cessent pas de s'améliorer de la simple voiture téléguidée aux robots complexes explorant Mars la gamme d'utilisation de technique de composant matériel d'heures de travail attribué à la conception en plus des tests sont très variés.

En développant notre robot, le RP6, nous avons pu mettre en pratique ces différentes fonctionnalités liées à la détection d'obstacles ainsi notre robot peut se déplacer librement dans l'espace grâce à ses différents capteurs. Nous avons pu mettre en évidence les points forts ainsi que les limites de ces capteurs. Le développement fut un long travail de recherche afin d'incorporer au mieux le capteur d'ultrasons SRF-05 et d'étudier ses fonctionnalités en passant par l'étude du phénomène des ultrasons.

Les domaines d'application de l'ultrason sont très nombreux nous pourrions permettre aux robots de se représenter son environnement en trois dimensions. La vitesse du son est plus élevée dans l'eau, cette technique pourrait être très utile aux robots marins.

Bibliographie :

- [Khatib1986] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. International Journal of Robotics Research, vol. 5, n 1, pages 90–98, Spring1986
- [Khatib 1997] M. Khatib, H. Jaouni, R. Chatila & J.P. Laumond. Dynamic path modification for car-like nonholonomic mobile robots. International Conference on Robotics and Automation. Albuquerque, NM, USA, avril 1997. IEEE.
- [Borenstein 1996] J. Borenstein, B. Everett & L. Feng. Navigating mobile robots: Systems and techniques. A. K. Peters, Ltd., Wellesley, 1996.
- [Stentz 1994] A Stentz Optimal and efficient path planning for partially known environments la Proceedings of the IEEE International Conference on Robotics and Automaton May 1994.
- [Fraichard 2004] T., Asama, H.: Inevitable collision state – a step towards safer robots? Advanced robotics1001 -1024 (2004).
- [Borenstein 1991] Borenstein, J. and Koren, Y., "Histogramic In-motion Mapping for Mobile Robot Obstacle Avoidance." IEEE Journal of Robotics and Automation, Vol. 7, No. 4, 1991, pp. 535-539.
- Borenstein, J. and Koren, Y., "Real-time Map-building for Fast Mobile Robot Obstacle Avoidance." SPIE Symposium on Advances in Intelligent Systems, Mobile Robots V, Boston, MA, Nov. 4-9, 1990.
- [Minguez 2000] J. Minguez and L. Montano. Nearness diagram navigation (nd): A new real time collision avoidance approach. In IEEEERSJ Int. Conf. on Intelligent Robots and Systems, 2000.
- [Quinlan 1993] Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993
- [Soueres 1996] P.Soueres, J.P.Laumond - Automatic Control, IEEE Transactions on, 1996
- [Hart 1968] Hart, P., Nilsson, N., and Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Syst. Science and Cybernetics*, SSC-4(2):100-107, 1968.
- [Latombe 1991] Title Robot Motion Planning: Edition en anglais
The Springer International Series in Engineering and Computer Science, Author : Jean-Claude Latombe, Edition: illustrated, Publisher: Springer Science & Business

Media, 1991, Length: 651 pages

- [Choset 1996] Choset, H. (1996). Sensor based motion planning: The hierarchical generalized voronoi graph. Doctoral dissertation, California Institute of Technology
- [Chazzeville et Guibas 1989] *Visibility and intersection problems in plane geometry* (Article, Discrete and Computational Geometry)
- [Shin et McKay 1986] Shin, K.G. and McKay, N.D., "A Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators," IEEE Transactions on Automatic Control, vol. AC-31, No. 6, June 1986.
- Shin, K.G. and McKay, N.D., "Selection of Near-Minimum Time Geometric Paths for Robotic Manipulators," IEEE Transactions on Automatic Control, vol. AC-31, NO. 6, June 1986.
- [Defoort 2007, 2009]
[Lawrence, Zhou et Tits] Experimental motion planning and control for an autonomous nonholonomic mobile robot, M Defoort, J Palos, A Kokosy, T Floquet, W Perruquetti, D Boulinguez, Robotics and Automation, 2007 IEEE International Conference on, 2221-2226p. 713-724. Distributed constraint reasoning applied to multi-robot exploration. A Doniec, N Bouraqadi, M Defoort, VT Le, S Stinckwich Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International
- [Quinlan 1994] J. Ross Quinlan. C4.5: programs for machine Learning. Morgan Kaufmann, 1994
- [Gaillard et al 2009] *Deterministic Kinodynamic Planning* (Workshop of the UK Planning and Scheduling Special Interest Group, Italy)
- [Fox et al. 1997] *The dynamic window approach to collision avoidance* (Article, IEEE Robotics and Automation Magazine)
- [Zadeh 1965] *Fuzzy sets* (Article, Information and Control)