

# EVITEMENT D'OBSTACLES DYNAMIQUES PAR UN ROBOT MOBILE

CHERIBET MOHAMED\*, LASKRI MOHAMED TAYEB\*\*

LRI/GRIA Département d'Informatique  
Université Badji Mokhtar Annaba, Algérie  
\*cheribet\_med@hotmail.com,\*\*laskri@univ-annaba.org

## ABSTRACT

Pour un robot évoluant dans un environnement donné, le problème général de planification consiste à déterminer pour le robot un mouvement lui permettant de se déplacer entre deux configurations données tout en respectant un certain nombre de contraintes et de critères. Ceux-ci découlent de plusieurs facteurs de natures diverses et dépendent généralement des caractéristiques du robot, de l'environnement et du type de tâches à exécuter. Les approches classiques de planifications de trajectoires ne fonctionnent pas en environnement à la fois dynamique et incertain: les calculs sont trop nombreux et complexes pour être exécutés en temps réel. Seules les méthodes purement réactives d'évitement d'obstacles sont applicables. Ces dernières sont cependant mal adaptées à la prise en compte de contrainte de tâches comme atteindre un but, Des approches récentes tendent à les combiner dans des méthodes hybrides appelées méthodes itératives.

**MOTS CLES :** Environnement dynamique, évitement d'obstacles, planification, robot mobile.

## 1 INTRODUCTION

De plus en plus la recherche dans le domaine robotique à intérêts industriels se concentre sur l'amélioration du degré d'autonomie du système robotique. Les robots sont conçus pour opérer dans différentes conditions, en prenant en compte des missions à périodes de temps plus long sans intervention humaine. Il s'agit notamment de tâches spécifiques dans un environnement dangereux ou hostile. Les systèmes de navigation autonome sont utilisés dans des applications comme les robots de service, de surveillance ou d'exploration où le robot se déplace et effectue la tâche principale à la fois. Un des aspects clés de ces robots est la mobilité, car elle constitue la base sur laquelle on peut intégrer plusieurs sous- systèmes avec des fonctionnalités différentes. Cependant, la performance du système de mouvement affecte fortement les performances des tâches. Des problèmes spécifiques se posent dans les applications qui peuvent avoir des conséquences fatales (par exemple, des robots qui transportent un matériel dangereux). La mobilité est étroitement liée à la nature de l'environnement. Dans de nombreuses applications, l'environnement ne peut être spécifié à priori avec une carte et peut être dynamique. Dans ces circonstances, des capteurs peuvent recueillir des informations sur l'environnement et adaptent les mouvements du robot à

toute nouvelle situation imprévue ou un événement. Les systèmes de mouvement à base de capteurs semblent être le choix naturel, mais la plupart d'entre eux ne permettent pas une navigation robuste et fiable dans des environnements très compliqués. Cela concerne généralement des endroits avec peu d'espace pour manœuvrer, des environnements hautement dynamiques ou qui conduisent à des situations de blocage. Un exemple est un bureau, où le robot se déplace parmi les chaises, les tables et étagères (avec des positions inconnues) et les humains (ce qui rend l'environnement hautement dynamique). La manœuvre est également extrêmement difficile dans certaines circonstances quand il y a peu d'espace pour manœuvrer. Notre travail s'intéresse au contrôle du mouvement d'un robot en vertu de ces conditions. Nous avons conçu un système à base d'un capteur laser avec trois modules qui fonctionnent ensemble pour mener à bien la tâche de mouvement. Notre conception est basée sur certaines exigences que nous avons identifiées pour conduire le robot dans des environnements dynamiques et gênants. Le système diffère des travaux antérieurs dans le choix et l'implémentation des modules et dans l'architecture d'intégration. Notre contribution concerne les aspects fonctionnels et informatiques de la conception des modules et de leur intégration, et la validation expérimentale (par

simulation) dans des environnements dynamiques. Notre système à base d'un capteur laser a été développé pour déplacer le robot aux positions souhaitées sans collisions. Cette fonctionnalité est uniquement un sous-ensemble du problème de navigation. D'autres aspects impliquent la perception, la modélisation et le contrôle. Ils ne seront pas abordés dans cet article, mais ils sont essentiels pour construire un système complet de mouvement autonome. Ce document est organisé comme suit: d'abord nous discuterons des travaux liés (section II), et les exigences d'un système de contrôle de mouvement à base de capteurs (Section III). Ensuite, nous donnerons un aperçu du système (section IV). Nous présentons les résultats de la simulation (section. V) et on termine par des conclusions (section. VI).

## 2 TRAVAUX LIÉS

Le problème de planification de mouvement dans des environnements dynamiques englobe des questions telles que la représentation de l'environnement (Modélisation), la délibération globale et la réactivité. La planification sans tenir compte de l'exécution est une restriction à un petit domaine du problème. C'est parce qu'il devient difficile de considérer tous les imprévus et ce n'est pas pratique de formuler des plans qui ne reflètent pas un environnement changeant. D'autre part, les systèmes réactifs limitent leur application à la perception-action, la flexibilité et la robustesse de mouvement. Le problème global ne peut être résolu individuellement par ces systèmes puisqu'ils ont besoin de modèles plus vastes de connaissances et d'une certaine manière à intégrer la mémoire. L'intérêt est porté sur la synthèse d'un mode de contrôle qui intègre les deux méthodes à la fois et non pas sur l'extension de deux environnements séparés, [1]. Les systèmes hybrides tentent de combiner les deux paradigmes en utilisant l'intelligence artificielle pour représenter et utiliser les connaissances, avec la meilleure réactivité, robustesse, adaptation et flexibilité. Fondamentalement, ces systèmes doivent combiner un planificateur (délibération) et un reacteur (exécution). Les principales différences entre eux sont les suivantes: (I) L'interaction entre le planificateur le reacteur (i.e. comment la méthode réactive utilise les informations disponibles du planificateur). (II) Les techniques utilisées pour mettre en œuvre chaque module. Une manière de préciser l'interaction entre la délibération et la réaction est de considérer la planification comme un composant qui fait la composition des différents comportements lors de l'exécution, [2]. Par exemple, ces comportements peuvent être mis en œuvre avec des champs de potentiel, [3], de sorte que la modification de leur poids change le comportement global du système. Une autre possibilité est d'utiliser la planification pour orienter le contrôle réactif, [4], ou comme un système qui adapte les paramètres de la

composante réactive à base de l'évolution de l'environnement, [5]. Dans les deux cas, la planification joue un rôle tactique, alors que le reacteur dispose d'un degré de liberté d'exécution. L'avantage de cette configuration planner- reactor est qu'elle combine la composante délibérative [le plan est toujours disponible dans l'exécution et s'améliore dans le temps] et la composante réactive (exécuteur du mouvement). Une perspective sur les architectures hybrides est donnée dans [6]. Dans le contexte de mouvement, une stratégie commune est de calculer une trajectoire de l'utiliser pour orienter le module réactif, [7]. Ces techniques nécessitent de grandes ressources de calcul, à cause de l'utilisation de planificateurs globaux (ils trouvent toujours un chemin s'il existe). D'autres techniques calculent un chemin qui se déforme en exécution en se basant sur l'évolution de l'environnement [dans l'espace de travail, [8], ou dans l'espace de configuration, [9]]. Néanmoins, ces méthodes doivent faire une replanification lorsque le chemin est invalidé ou quand le robot se déplace très loin de la configuration initiale en raison des obstacles inattendus. Alternativement, Ulrich et Borenstein (1998) [10]

Présente une stratégie de création d'arbres de chemins obtenus par l'exécution de l'algorithme réactive à quelques longueurs d'avance sur l'exécution. Ce système obtient de bons résultats en plateformes ayant peu de ressources de calcul, mais ne garantit pas une convergence vers la cible. Une autre possibilité consiste à calculer un canal d'espace libre qui contient des ensembles de chemins, en laissant le choix à l'exécution, [11]. Ces questions sont étroitement liées au choix et à l'implémentation des techniques pour chaque module. Toutes les stratégies citées antérieurement, utilisent le planificateur global pour obtenir un chemin afin d'orienter le contrôle réactif. Le planificateur est habituellement une technique numérique efficace exécutée en temps réel. Un autre module clé est la méthode réactive elle-même. Certaines techniques sont basées sur des méthodes potentielles, mais elles ont des limites. D'autres techniques calculent un ensemble de commandes de mouvement intermédiaires et de choisir une qui sera la prochaine à exécuter. Les commandes sont des directions du mouvement, ensembles de vitesses, ou ensembles de trajectoires. Toutefois, ces méthodes réactives sont d'une utilité limitée quand le scénario rend difficile la manœuvre du robot (généralement avec une densité d'obstacles élevée). Le travail de Minguez et Montano (2004)

[12] a identifié certaines conséquences comme les situations locales de blocage, les mouvements irréguliers et oscillants, ou l'impossibilité de conduire le robot dans des zones à forte densité d'obstacles ou lorsque il est loin de la direction du but. Ces comportements acquièrent une

plus grande pertinence dans le développement d'applications robustes pour to navigué indépendamment de la difficulté de l'environnement. Les systèmes de navigation qui reposent sur des méthodes réactives héritent aussi de ces inconvénients, limitant leur utilisation dans des applications pratiques. Enfin, dans ce type d'architecture, les modèles sont généralement construits pour servir de base pour le planificateur et fournir une mémoire de courte durée pour le comportement réactif. Dans les environnements d'intérieur, les grilles d'occupation sont généralement utilisées avec des ultrasons, le laser. Dans ce document, nous décrivons un système hybride avec une configuration Planner – Reactor synchrone et hétérogène, où les deux modules utilisent le modèle construit dans l'exécution et de mener à bien la tâche de mouvement. Notre principale contribution est le choix et la conception des modules et leur intégration. En conséquence, le nouveau système peut déplacer le robot dans des environnements très difficiles, là où d'autres systèmes peuvent trouver des difficultés.

### 3 EXIGENCES POUR UN SYSTEME DE CONTROLE A BASE DE CAPTEUR

La version de base de ces systèmes déplace le robot entre positions sans collision. L'opération est régie par un processus *perception - action* répété à une fréquence élevée. Les capteurs recueillent des informations sur l'environnement (obstacles) et le robot, qui servent pour calculer le mouvement. Le robot exécute le mouvement et le processus redémarre. Le résultat est une séquence de mouvements en ligne qui conduit le robot jusqu'au but sans collisions. Dans cette Section nous précisons certaines exigences générales relatives à ces systèmes [13]:

#### 3.1 L'intégration de l'information

Les mesures successives sensorielles doivent être stockées ou intégrées pour construire une représentation de l'environnement. Cela est nécessaire pour éviter les obstacles qui ne sont pas perçus au moment (contraintes de visibilité du capteur), et d'accroître la portée de l'information utilisée (augmentation de la domination spatiale). En outre, les changements dans les scénarios dynamiques doivent être rapidement intégrés dans le modèle. Sinon, le robot va éviter les zones perçues comme un espace libre ou il ne va pas éviter des obstacles déjà perçus, puisque dans les deux cas, l'information n'a pas pu être stockée dans le modèle.

#### 3.2 L'évitement des situations de piège et des comportements cycliques

Le système doit être muni d'une stratégie visant à éviter ces situations. Plusieurs configurations différentes d'obstacles peuvent piéger le robot (l'exemple le plus typique est un obstacle de la forme U-shape ou les end-zones) ou créer un mouvement cyclique (par exemple, Les distributions symétriques d'obstacles). Le robot ne pourra jamais atteindre l'emplacement final dans ces circonstances.

#### 3.3 Génération d'un mouvement robuste

Le mouvement final doit être calculé par une méthode réactive robuste. Cet algorithme doit être capable d'éviter les collisions indépendamment de la difficulté de l'environnement. Comme règle générale, les scénarios qui posent plus de problèmes ont une large densité d'obstacles où la manœuvre est difficile et critique.

#### 3.4 L'intégration de fonctionnalités

Toutes les fonctionnalités doivent être intégrées au sein d'une architecture pour la spécification, la coordination et la non détection et la récupération. L'intégration doit être munie d'un cycle perception – action à une fréquence élevée, ce cycle fixe la réactivité du système pour les changements imprévisibles (détectés par les capteurs). En outre, cela favorise la portabilité entre les différentes plates-formes et capteurs (les interactions entre les modules ne sont pas conçus à partir de zéro lorsque les modules sont remplacés). Dans notre travail, nous nous efforçons de rendre notre système capable de remplir ces exigences.

### 4 SYSTEME HYBRIDE

Nous fournissons une vue générale de notre système hybride qui comporte trois modules et une architecture pour la supervision. Les fonctionnalités des modules sont la modélisation, la planification et la navigation réactive:

#### 4.1 Module de modélisation

Il construit un modèle de l'environnement (intégration de l'information sensorielle).

Nous avons utilisé une grille d'occupation binaire dont les cellules sont mises à jour chaque fois qu'une mesure sensorielle est disponible. La grille a une taille limitée (représentant une partie fixe de l'espace), et dont l'emplacement

est recalculée en permanence pour inclure la position du robot (Figure 1).

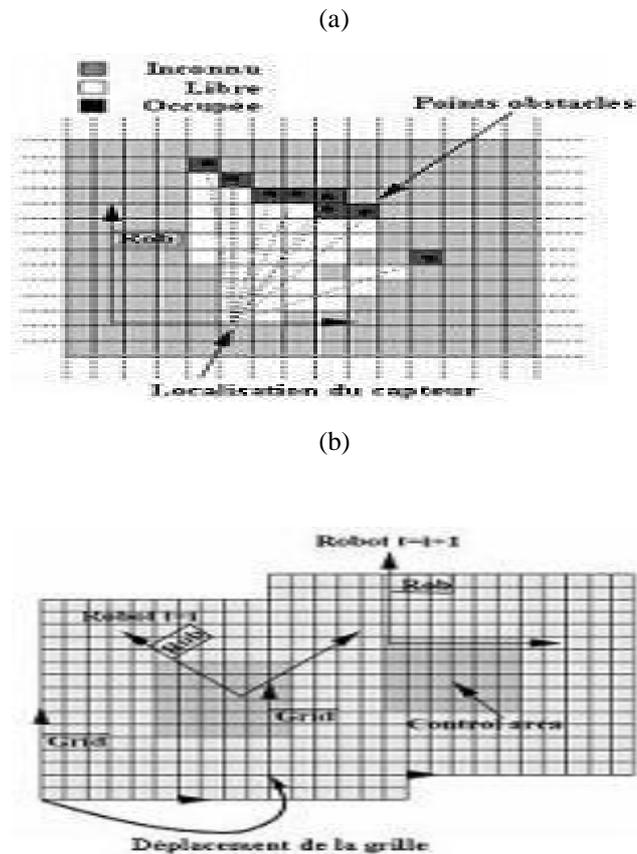


Figure 1

Ces figures montrent comment les mesures laser sont intégrés dans la grille et comment la position de la grille est recalculée pour laisser le robot dans la zone centrale. (a) Dans le scan laser, les cellules sont occupées pour les points obstacles, et les cellules dans les lignes qui joignent les points obstacles et le capteur sont libres. (b) À l'instant  $t = i$  le robot est dans la zone de contrôle de la grille. Ensuite, à l'instant  $t = i + 1$ , le robot a quitté la zone de contrôle et l'emplacement de la grille est recalculé (en multiples de taille de cellule et sans rotation) afin que la nouvelle position du robot soit dans cette zone.

#### 4.2 Module de planification

Ce module utilise un planificateur de mouvement pour obtenir des informations afin d'éviter les situations de piège et les mouvements cycliques (pas pour contrôler le robot). Le planificateur utilise une fonction de navigation (NF1) sur la grille du module précédent, et calcule alors un chemin vers la destination en utilisant une stratégie de

descente de gradient. Nous avons choisi ce planificateur parce que la fonction de navigation n'a pas de minima locaux (elle trouve un chemin s'il existe), et c'est une fonction numérique qui fonctionne d'une manière efficace sur une grille (i.e., la représentation existante).

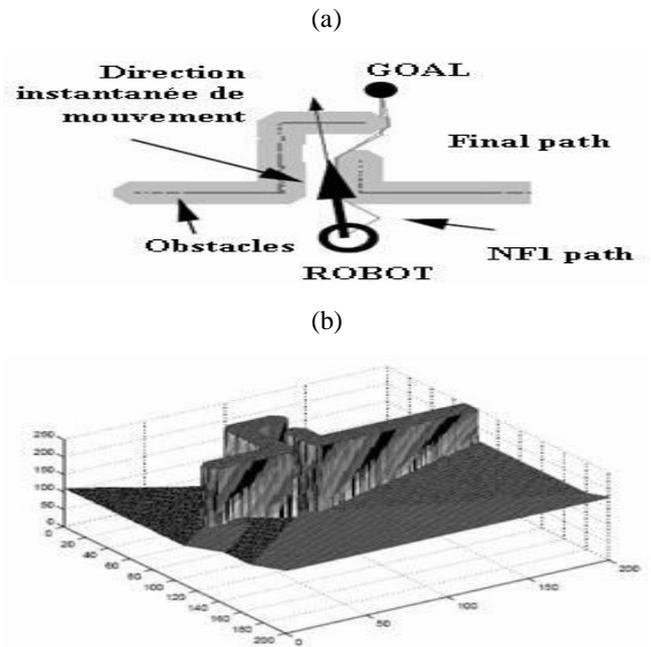


Figure 2

Cette figure montre le fonctionnement du planificateur. Premièrement, les obstacles sont agrandis au rayon du robot (a). Ensuite, la NF1 est calculée en propageant une vague à partir de la position du but, où chaque cellule est étiquetée avec le coût cumulé de la vague (b). Sur cette fonction un chemin est calculé grâce à une stratégie de recherche basée sur la descente du gradient, le chemin NF1 dans la figure (a). Enfin, le chemin est étiré (pour être optimisé) afin d'obtenir la direction instantanée du mouvement à partir de la première partie du chemin.

Le planificateur utilise un processus à deux étapes pour calculer un chemin de la position du robot à la destination (Figure 2). Premièrement la fonction de navigation est calculée. Chaque obstacle est élargi au rayon du robot, puis la NF1 est construite par propagation d'une vague à partir de la destination sur les cellules libres (chaque cellule est marquée par la distance mesurée en nombre de cellules jusqu'au but) (Figure 3). Deuxièmement, un chemin est calculé en utilisant une stratégie de recherche basée sur la descente du gradient.

Le planificateur utilise un processus à deux étapes pour calculer un chemin de la position du robot à la destination (Figure 2). Premièrement la fonction de navigation est calculée. Chaque obstacle est élargi au rayon du robot, puis la NF1 est construite par propagation

d'une vague à partir de la destination sur les cellules libres (chaque cellule est marquée par la distance mesurée en nombre de cellules jusqu'au but) (Figure 3). Deuxièmement, un chemin est calculé en utilisant une stratégie de recherche basée sur la descente du gradient

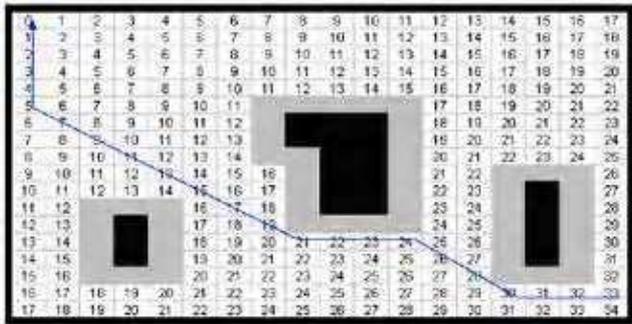


Figure 3: Chemin généré par la fonction de navigation.

### 4.3 Module de navigation réactive

Il calcule un mouvement libre en évitant la collision avec les obstacles. La méthode utilisée est la Vector Field Histogram (VFH en bref, qui est basée sur la sélection d'une situation de navigation à chaque moment et l'application de la loi de mouvement associée. Cette méthode est très efficace et robuste dans les environnements avec peu d'espace pour manœuvrer. Dans leur version d'origine, les VFH [14] consistent à représenter dans un premier temps l'environnement par une grille d'occupation. Chaque cellule contient une grandeur réelle correspondant en quelque sorte à la probabilité de trouver un obstacle à cet emplacement. La grille est ensuite traduite en un histogramme dont chaque « barre » représente une direction du robot et dont la hauteur est proportionnelle à la probabilité de percuter un obstacle en suivant cette direction (Figure 4). Par seuillage, il est ainsi possible d'identifier les directions libres du robot, appelées passages. Une fonction de coût prenant en compte la direction du but, la direction précédemment suivie et l'orientation courante des roues permet de sélectionner un des passages et la direction correspondante. La vitesse est ensuite calculée en fonction de la distance du robot aux obstacles. Une version plus récente (VFH+) [10] prend en compte la largeur du robot et ses contraintes cinématiques (les trajectoires ne sont plus assimilées à des droites mais à des arcs de cercles.

Un des inconvénients majeur de l'approche est la difficulté à régler le seuil déterminant les « passages » et la difficulté à naviguer en milieu contraint (couloirs étroits, passage de portes). De plus, la dynamique du robot n'est pas réellement prise en compte et le traitement purement

local empêche de garantir la prise en compte des contraintes de tâches. Ce dernier point a toutefois été traité dans la dernière version de cette approche sous la dénomination VFH\* [15], où plusieurs déplacements sont testés à l'avance. Ceci a pour effet de limiter les situations de blocage et d'améliorer la convergence vers le but

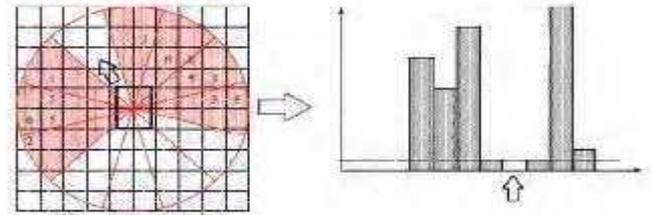


Figure 4: Représentation de la grille par un histogramme polaire

Architecture d'intégration:

Elle intègre les modules suivant une configuration synchrone *Planner - Reactor*, [16]

[18], où les deux parties utilisent le modèle construit pendant l'exécution. La synchronisation entre modules permet d'éviter les problèmes d'incohérences dans le temps. Le système fonctionne comme suit (Figure 5): étant donné un Laser scan et de l'odométrie du robot, le constructeur du modèle intègre cette information dans le modèle existant. Ensuite, l'information sur les obstacles et l'espace libre dans la grille d'occupation est utilisée par le module de planification pour calculer le chemin jusqu'au but. Enfin, le module réactif utilise l'information sur les obstacles dans la grille et l'information du planificateur global pour générer un mouvement (conduire le robot vers sa destination sans collisions). Le mouvement est exécuté par le contrôleur de robot et le processus redémarre. Il est important de souligner que les trois modules travaillent de manière synchrone dans le cycle perception - action. Cela renforce l'importance du choix et les aspects informatiques des techniques utilisées dans chaque module.

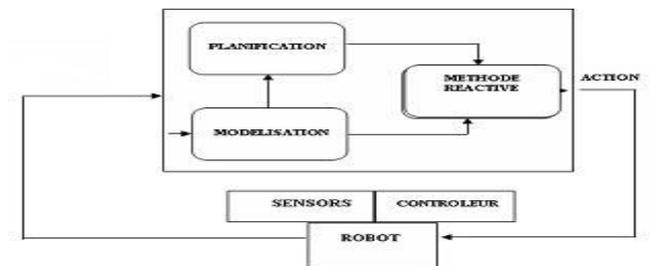


Figure 5

Cette figure illustre le cycle perception - action du système de contrôle de mouvement à base de capteurs. Les

perceptions sont des mesures du capteur Laser et l'odométrie du robot, L'action est le calcul de commandes pour un mouvement sans collision. Le système comporte trois modules qui coopèrent pour mener à bien la tâche de navigation: la modélisation, le planificateur global et la méthode de navigation réactive

## 5 EXPERIMENTATION

Les premiers travaux de notre recherche ont été d'identifier une expérience à réaliser et d'identifier le simulateur à utiliser pour cette expérience. Nous avons étudié plusieurs plateformes [17] de simulation. Après évaluation des capacités de ces plateformes, nous avons choisi Player/Stage/Gazebo dans sa version 2D (Player/Stage) pour sa souplesse d'utilisation dans le contexte multi-robot. Ce simulateur est aujourd'hui un projet GNU, initialement développé par une collaboration internationale des chercheurs en robotique. Il permet d'écrire des programmes et d'exécuter ces derniers en simulation ou sur des plateformes réelles. Ce simulateur nous permet de reproduire des expériences concrètes d'exploration d'un environnement dynamique. Le simulateur nous permet aussi de planifier des trajectoires pour un robot mobile dans un environnement référencé capteurs. Nous considérons comme point de départ un environnement partiellement connu, dans lequel il s'agit de réaliser une mission robotique. La question de la planification robotique peut se résumer à l'affectation d'une cible au robot mobile afin de minimiser le temps global de la mission. Dans ce qui suit nous présentons les expériences réalisées afin de valider notre approche. Nous avons mené ces expériences pour résoudre deux types de problème: (1) *Evitement de situations de pièges et comportements cycliques*. (2) *Evitement d'obstacles dans un environnement dynamique*.

### 5.1 Evitement de situations de pièges et comportements cycliques

Dans cette expérience, le système de navigation résout plusieurs situations de piège et évite les comportements cycliques produits par la méthode réactive.

La figure 6 montre des situations de piège où notre méthode réactive utilisée VFH échoue dans la mission d'atteindre le but et le robot se trouve bloqué infiniment dans un comportement cyclique comme l'indique les figures 6a; piège sous forme de U très célèbre et qui constitue un point faible des méthodes réactives. La figure 6c montre une situation où le robot fait un chemin plus long et sort d'une situation de piège après plusieurs cycles. Un autre point faible de la méthode réactive est montré dans la figure 6e où le robot se trouve bloqué par l'angle de la porte et entre dans un comportement cyclique infini. Le caractère local de la méthode lui confère un inconvénient majeur: les " minima locaux " de la fonction de sélection. En effet, un coût minimal peut être attribué à un déplacement répondant localement aux critères de choix désirés, mais qui, dans le futur, conduira le robot à une situation de blocage ou à une collision. Les figures 6b, 6d, 6f montre comment les situations de piège précédentes ont été évitées par notre approche hybride. La coopération entre les trois modules permet d'éviter le mouvement cyclique et les situations de piège. C'est parce que la direction du mouvement calculé par le planificateur contient les informations nécessaires pour éviter ces situations et les environnements symétriques ne produisent pas de mouvements cycliques puisque la direction du mouvement calculée par le planificateur fait la distinction entre les zones de mouvements possibles.

6e où le robot se trouve bloqué par l'angle de la porte et entre dans un comportement cyclique infini. Le caractère local de la méthode lui confère un inconvénient majeur: les " minima locaux " de la fonction de sélection. En effet, un coût minimal peut être attribué à un déplacement répondant localement aux critères de choix désirés, mais qui, dans le futur, conduira le robot à une situation de blocage ou à une collision. Les figures 6b, 6d, 6f montre comment les situations de piège précédentes ont été évitées par notre approche hybride. La coopération entre les trois modules permet d'éviter le mouvement cyclique et les situations de piège. C'est parce que la direction du mouvement calculé par le planificateur contient les informations nécessaires pour éviter ces situations et les environnements symétriques ne produisent pas de mouvements cycliques puisque la direction du mouvement calculée par le planificateur fait la distinction entre les zones de mouvements possibles.



Figure 6: *Evitement de situations de pièges et comportements cycliques*

La figure 6 montre des situations de piège où notre méthode réactive utilisée VFH échoue dans la mission d'atteindre le but et le robot se trouve bloqué infiniment dans un comportement cyclique comme l'indique les figures 6a; piège sous forme de U très célèbre et qui constitue un point faible des méthodes réactives. La figure 6c montre une situation où le robot fait un chemin plus long et sort d'une situation de piège après plusieurs cycles. Un autre point faible de la méthode réactive est montré dans la figure 6e où le robot se trouve bloqué par l'angle de la porte et entre dans un comportement cyclique infini. Le caractère local de la méthode lui confère un inconvénient majeur: les " minima locaux " de la fonction de sélection. En effet, un coût minimal peut être attribué à un déplacement répondant localement aux critères de choix désirés, mais qui, dans le futur, conduira le robot à une situation de blocage ou à une collision. Les figures 6b, 6d, 6f montre comment les situations de piège précédentes ont été évitées par notre approche hybride. La coopération entre les trois modules permet d'éviter le mouvement cyclique et les situations de piège. C'est parce que la direction du mouvement calculé par le planificateur contient les informations nécessaires pour éviter ces situations et les environnements symétriques ne produisent pas de mouvements cycliques puisque la direction du mouvement calculée par le planificateur fait la distinction entre les zones de mouvements possibles.

## 5.2 Évitement d'obstacles dans un environnement dynamique

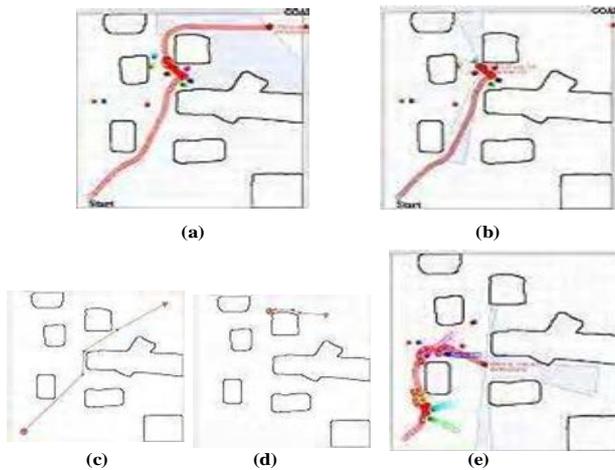


Figure 7

(a) Le robot est entouré de plusieurs robots et reste bloqué  
 (b) Le robot trouve un chemin vers le but après le déplacement du robot le plus haut vers la gauche (c) Le plan global généré par le planificateur (d) Un plan local parmi plusieurs plan générés en cours d'exécution. (e) Évitement d'obstacles mobiles.

Cette expérience met en évidence notre approche hybride dans un environnement difficile (dynamique, complexe et très dense), où le robot navigue le long d'un passage très étroit avec des robots et objets distribués au hasard. (Figure 7). La difficulté réside surtout dans le passage étroit entre les deux robots les plus à droite et compte tenu des distances de sécurité et de la dynamique du robot qui ne permettent pas au robot de suivre la trajectoire calculée par le planificateur (Figure 7c), celui-ci se trouve entouré de plusieurs obstacles où il y a peu d'espace pour faire des manœuvres. Dans certains endroits le robot se trouve à une distance plus petite des obstacles distribués de chaque côté du robot (Figure 7a). Le robot entre dans un comportement cyclique infini. La figure 7b montre comment le robot; après le déplacement du robot le plus haut vers la gauche; a pu atteindre le but en suivant un autre chemin après plusieurs replanifications (Figures 7c, 7d). En effet, le déplacement du robot le plus haut vers la gauche laisse suffisamment d'espace au robot pour sortir de la situation de piège. La figure 7e montre une expérience où le robot évite des obstacles mobiles.

Les principaux aspects de cette expérience sont les performances individuelles et la coopération entre le module de modélisation et le module réactif. Les scans laser dans le module de modélisation ont été rapidement intégrés dans la grille. Ainsi, la méthode réactive évite les nouveaux obstacles dès qu'ils ont été perçus. Cette réactivité est essentielle pour déplacer le robot dans des environnements denses car les retards dans le calcul du mouvement peuvent

conduire à des collisions. Deuxièmement, les dernières mesures sensorielles restées dans la grille ont été utilisées par le module réactif pour la tâche d'évitement. Cela a été important car il arrive que le capteur ne perçoive pas les obstacles les plus proches en raison de contraintes de visibilité. Toutefois, étant donné qu'ils étaient perçus à quelques instants avant, ils sont restés dans la grille et ont été évités (Figure 7).

Cette expérience a été particulièrement difficile du point de vue de la génération du mouvement en raison de l'espace étroit, car la plupart des techniques existantes ont des limitations intrinsèques qui pénalisent le calcul du mouvement en vertu de ces circonstances. Toutefois, la méthode VFH conduisait le robot sans collisions, même au milieu des obstacles très proches. Le mouvement est libre d'oscillations (voir le chemin exécuté dans la figure 7b après déplacement du robot le plus haut vers la gauche) et évite tous les pièges qui peuvent être créés en raison de la densité d'obstacles.

Pour conclure, le système a pu déplacer le robot dans des environnements différents en respectant les exigences pour un système de contrôle à base de capteur (Section III) en ce qui concerne la *génération de mouvements robustes* et *l'intégration de l'information*. *Les mouvements cycliques* et *les situations de piège* ont également été évités.

## 6 CONCLUSION

La stratégie de notre approche consiste à calculer un chemin par le planificateur global basé sur une fonction de navigation (*wavefront*) et l'utiliser pour diriger la méthode réactive. Notre approche a les avantages d'un comportement réactif, tels que l'adaptation rapide aux changements et la flexibilité dans des circonstances imprévisibles. Elle permet d'éviter les limitations typiques d'autres méthodes, comme les situations de piège locales, les oscillations dans des environnements denses, et l'impossibilité d'obtenir une direction de mouvement vers les obstacles ou vers les zones à grande densité d'obstacles. Il en résulte un mouvement sûr et robuste dans des environnements qui restent toujours gênant pour de nombreuses méthodes existantes. En conclusion, notre approche est dotée de meilleures capacités de manœuvrabilité spécialement en environnements dynamiques, denses, complexes et gênants.

Ce travail a ouvert de nouvelles perspectives dans le domaine de la navigation autonome d'un robot mobile. Cependant, il reste encore des points à étudier et développer plus profondément et enrichir l'approche par d'autres méthodes de cartographie et de localisation qui font parties du système complet de navigation.

## REFERENCES

- [1] Arkin, R., 1989. Towards the unification of navigational planning and reactive control. In: Working Notes of the AIII Spring Symposium on Robot Navigation. Stanford University, pp. 1–6.
- [2] Arkin, R., 1987. Motor schema based navigation for a mobile robot: An approach to programming by behavior. In: IEEE International Conference on Robotics and Automation. Robotics and Automation. Raleigh, USA, pp. 264–271.
- [3] O Khatib, Real-Time obstacle avoidance for manipulators and mobile robots, *International Journal Robotics research* 5(1), 90-98, 1986
- [4] Agree P., Chapman, D., 1990. What are the plans for. *Journal for Robotics and Autonomous Systems* 6, 17–34.
- [5] Lyons, D., Hendriks, A., 1992. Planning, reactive. In: Saphiro, S., Wiley, J. (Eds.), *Encyclopedia of Artificial Intelligence*. pp. pp. 1171–1182.
- [6] Arkin, R., 1999. *Behavior-Based Robotics*. The MIT Press.
- [7] Minguez, J., Montano, L., Simeon, N., Alami, R., 2001. Global Nearness Diagram Navigation (GND). In: IEEE International Conf. on Robotics and Automation. Seoul, Korea, pp. 33–39.
- [8] Brock and Khatib, Real-Time Replanning in High-dimensional Configuration Spaces Using Sets of Homotopic Paths, In IEEE int. conf. on Robotics and Automation, San Francisco, USA, 2000
- [9] Quinlan, S., Khatib, O., 1993. Elastic Bands: Connecting Path Planning and Control. In: IEEE Int. Conf. Conf. on Robotics and Automation. Vol. Vol. 2. 2. Atlanta, USA, pp. 802–807. 802-807.
- [10] Ulrich and Borenstein, VFH+ : Reliable Obstacle Avoidance for Fast Mobile Robots, In IEEE int. conf. on Robotics and Automation, Leuven, Belgium, 1998
- [11] Choi, W., Latombe, J., 1991. A reactive architecture for planning and executing robot motion with incomplete knowledge. In: IEEE/RSJ International Workshop on Intelligent Robots and Systems. Osaka, Japon, pp. 24–29.
- [12] Minguez, J., Montano, L., 2004. Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios. *IEEE Transactions on Robotics and Automation* 20 (1), 45–59.
- [13] Minguez, J., & Montano, L. (2005). Sensor-based robot motion generation in unknown, dynamic and troublesome scenarios. *Robotics and Autonomous Systems*, 52(4), 290–311.
- [14] Borenstein and Koren, The vector field histogram – fast obstacle avoidance for mobile robots, *IEEE Transactions on Robotics and Automation* 7(3), 278-288, 1991
- [15] I.Ulrich, J. Borenstein, VFH\* : Local Obstacle Avoidance with Lookahead Verification In IEEE int. conf. on Robotics and Automation, San Francisco, USA, 2000
- [16] Rodney A. Brooks: A robust layered control system for a mobile robot. *Rapport technique*, Cambridge, MA, USA, 1985.
- [17] Kramer, J., Scheutz, M., “Development environments for autonomous mobile robots: A survey”, C \_Springer Science + Business Media, LLC 2006.
- [18] L. Montesano, J. Minguez, L. Montano. Modeling Dynamic Scenarios for Local Sensor-Based Motion Planning. *Autonomous Robots*, Vol. 12, No3, 231-251, 2008