

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira Bejaïa

Faculté de Technologie
Département de Génie Electrique



Mémoire de fin d'études

En vue de l'obtention du diplôme de Master en Automatique et Systèmes

Thème :

Planification de trajectoire sans collision d'un robot manipulateur en utilisant l'approche de champs de potentiels artificiels

Préparé et présenté par :

SMAOUN Ramdhane

BELATRECHE Zakaria

Membres du jury :

M. LEHOUCHE Hocine, président

M. HADDAR Hocine, directeur

M. MENDIL Boubekour, examinateur

Année universitaire : 2018/2019

Remerciements

Comme tout travail de recherche est le résultat de la contribution de plusieurs personnes nous tenons par conséquent à exprimer nos remerciements à tous ceux qui nous ont aidés peu ou prou dans la réalisation de ce mémoire.

Nous tenons tout d'abord à remercier Dieu de nous avoir donné le courage, la force et l'esprit d'abnégation.

Nous remercions, les membres du jury qui ont accepté de lire et d'évaluer notre modeste travail : ainsi nous remercions notre directeur de mémoire M. HADDAR Hocine qui nous a accompagnés tout au long de cette année et grâce à qui ce travail a pu voir le jour, nous remercions notre examinateur M. MENDIL Boubekeur, et M. LEHOUCHE Hocine, président du jury.

De même, nos remerciements sont adressés à tous nos enseignants des cycles primaire, moyen, et secondaire ainsi que ceux du département du Génie-Electrique et notamment le département d'Automatique.

Nous tenons également à remercier M. LATOUI Farid qui nous a aidés et encouragés et M. BERRAH pour ses conseils.

Dédicace

Je dédie ce modeste travail :

A mes très chers parents,

A mes frères,

A ma très chère grand-mère

A la mémoire de mon défunt frère aîné IDIR,

A Toute la famille Smaoun,

A tous ceux qui ont contribué à la réalisation de ce travail,

Aux membres du jury,

A mes enseignants d'Automatique

A tous mes enseignants

Et à tous ceux que je connais.

SMAOUN RAMDHANE

Dédicace

*Je dédie ce travail à mes très chers parents, mon père Mouloud
et ma mère Noudjoud,*

A mon très cher frère Oussama,

*A la mémoire de ceux qui nous ont quitté ce monde, à mon cher
grand-père Arezki et mon proche ami Mohamed Lamine et mon
cher cousin Djoudi,*

A toute la famille Benlatreche et Kheniche,

A ceux qui étaient à mes cotés et à ceux qui ne l'étaient pas

Et à mes amis.

Zakaria Benlatreche

Tables des matières

Introduction générale.....	1
Chapitre I : La robotique.....	3
I.1. Introduction.....	4
I.2. Historique.....	4
I.3. Classification des robots.....	4
I.4. Structure générale d'un robot manipulateur.....	5
I.4.1. Système mécanique articulé.....	6
I.4.2. Actionneurs ou organes de motorisation.....	7
I.4.3. Effecteur.....	7
I.4.4. Capteurs ou organes de perception.....	7
I.5. Définitions.....	8
I.5.1. Degré de liberté et redondance.....	8
I.5.2. Configuration singulière.....	8
I.5.3. Espace opérationnel.....	8
I.5.4. Système de traitement.....	9
I.5.5. L'interface Homme-Machine.....	9
I.5.6. La commande.....	9
I.6. Matrices et transformations homogènes.....	9
I.6.1. Coordonnées homogènes.....	9
I.6.2. Transformations homogènes.....	10
I.7. Modélisation.....	11
I.7.1. Modélisation géométrique.....	12
I.7.1.1. Modèle géométrique direct.....	14
I.7.1.2. Modèle géométrique inverse.....	14
I.7.2. Modélisation cinématique.....	16
I.7.2.1. Modèle cinématique direct.....	16
I.7.2.2. Modèle cinématique inverse.....	20
I.7.3. Modélisation dynamique.....	20
I.7.3.1. Formalisme de Lagrange.....	21
I.7.3.2. Vitesse des articulations.....	22
I.7.3.3. Energie cinétique.....	22
I.7.3.4. Energie potentielle.....	23
I.7.3.5. Équation du mouvement d'un robot manipulateur.....	23

I.8.Conclusion.....	24
Chapitre II : Planification de trajectoire.....	25
II.1. Introduction.....	26
II.2. Formes de planification.....	26
II.3. Planification de trajectoire	27
II.4. Résolution du problème de planification.....	29
II.5. Classification des méthodes de planification.....	29
II.6. Planification de trajectoire sous contraintes.....	34
II.7. Les algorithmes d'évitement d'obstacle.....	35
II.8. Conclusion.....	40
Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique.....	41
III.1.Introduction	42
III.2. Bitmap de l'espace de travail et bitmap l'espace de configuration.....	42
III. 3.Calcul du champ de potentiel artificiel.....	43
III. 3.1.Calcul du champ de potentiel de l'espace de travail WSAPF.....	44
III.3. 2. Calcul du champ de potentiel de l'espace de configuration CSAPF.....	50
III.4.Planification en utilisant le champ de potentiel artificiel.....	51
III.5.Résultats de simulation.....	52
III.5.1. Cas d'un robot ponctuel.....	54
III.5.2. Cas d'un robot rectangulaire dans la scène1.....	56
III.5.3. Cas d'un robot rectangulaire dans la scène2.....	58
III.5.4. Cas d'une chaîne articule de 3 ddls.....	59
III.5.5. Cas d'une chaîne articule de 4 ddls.....	62
III.6. Conclusion.....	63
Conclusion générale.....	64
Références bibliographiques.....	66

Liste des figures

Figure I.1. Structure générale d'un robot manipulateur.....	5
Figure I.2. Représentation d'une articulation rotoïde.....	6
Figure I.3. Représentation d'une articulation prismatique.....	6
Figure I.4. Type d'articulations.....	7
Figure I.5. Représentation d'un point dans l'espace.....	10
Figure I.6. Association des repères selon la convention D-H.....	13
Figure I.7. Transformations entre l'organe terminal et le repère atelier.....	14
Figure I.8. Bras à deux articulation.....	14
Figure I.9. Modèle géométrique direct et inverse.....	15
Figure I.10. robot planaire RRR dont les repères ont été placés selon la convention D-H.....	17
Figure II.1. Planification et génération de trajectoires.....	27
Figure II.2. suivi d'une trajectoire.....	28
Figure II.3. Planification dans l'espace articulaire.....	28
Figure II.4. Planification dans l'espace cartésien.....	29
Figure II.5. Planification dans l'espace image.....	29
Figure II.6. Diagramme de Voronoi.....	32
Figure II.7. Graphe de visibilité.....	32
Figure II.8. Décomposition cellulaire.....	33
Figure II.9. Réseaux probabilistes.....	33
Figure II.10. Algorithme BUG 1.....	37
Figure II.11. Algorithme BUG 2.....	38
Figure II.12. Exploration de la même région plusieurs fois par un robot qui utilise BUG 2.....	39
Figure II.13. Un cas où un robot fait un tour complet sur l'obstacle sous BUG 2.....	39
Figure III.1. Algorithme de calcul de WSAPF.....	45
Figure III.2. Les iso-contours du WSAPF pour une grille de (513x513).....	46
Figure III.3. Champs de potentiel dans l'espace de travail WSAPF simple dans la grille 33x33.....	46
Figure III.4. Algorithme de calcul de la distance L_1 des obstacles.....	47
Figure III.5. Squelette de l'espace de travail obtenu pour une grille de 513x513.....	48
Figure III.6. Algorithme de calcul du WSAPF amélioré.....	49
Figure III.7. Les iso-contours du WSAPF amélioré pour une grille de (513x513).....	50

Figure III.8. Spécification des coordonnées des vertex des différents obstacles.....	53
Figure III.9. Scène considérée pour le cas d'un robot ponctuel et les obstacles spécifiés : III.6.....	53
Figure III.10. Le bitmap de l'espace de travail sous une résolution de 33x33.	54
Figure III.11. Chemin obtenu pour un robot ponctuel en utilisant le champ de potentiel artificiel simple dans l'espace de travail.....	55
Figure III.12. Chemins obtenus pour un robot ponctuel avec une résolution de 513x513 avec 1. WSAPF simple ; 2. WSAPF amélioré.....	56
Figure III.13. Chemin obtenu pour un robot polygonal avec une grille de (65x65) et : 1. WSAPF simple 2. WSAPF amélioré.....	57
Figure III .14. Chemin obtenu pour un robot polygonal avec une grille de (65x65) : 1. WSAPF simple ; 2. WSAPF amélioré. $q_{init}=[0.075 \ 0.9 \ \pi/2]$; $q_{goal}=[0.9 \ 0.03 \ 0]$	58
Figure III.15. Spécification du robot manipulateur et des obstacles de la scène considérée.....	59
Figure III.16. Scène et robot considérés pour la quatrième application.....	59
Figure III.17. Planification du chemin pour un manipulateur plan de 3 ddls en utilisant WSAPF amélioré.....	61
Figure III.18. Configurations initiale et finale de la chaîne articulée de 4 ddls.....	62
Figure III.19. Représentation de certaines configurations obtenues pour la chaîne de 4ddls.....	63

Liste des abréviations

m	Nombre de degrés de liberté
R	Articulation rotoïde
P	Articulation prismatique
(P_x, P_y, P_z)	Coordonnées cartésiennes
P_t	Facteur d'échelle
${}^i T_j$	Matrice de transformation
Trans (α, b, c)	Matrice de translation pure
Rot (x, y)	Matrice de rotation pure
(x, y)	Variables opérationnelles
J	Matrice jacobéenne
F	Force
M	Masse
W	Vitesse angulaire
C_i	Tenseur d'inertie
L	Lagrange
T	Énergie cinétique totale
V	Énergie potentielle totale
Γ	Couple
$(q \text{ ou } \theta)$	Position articulaire
$(\dot{q} \text{ ou } \dot{\theta})$	Vitesse articulaire
$(\ddot{q} \text{ ou } \ddot{\theta})$	Accélération articulaire
P_0	Position initial
P_f	Position finale
T	Temps
P_j	Vecteur de points de contrôle
T	Le couple
S	Position articulaire
\dot{s}	Vitesse articulaire
\ddot{s}	Accélération articulaire
R	Vecteur position
\dot{r}	Vecteur de vitesse
t_f	Temps final
r_c	Position du centre de masse
\dot{r}_c	Vitesse du centre de masse
$\tau_{\min}(\theta, \dot{\theta})$	Le couple minimale
$\tau_{\max}(\theta, \dot{\theta})$	Le couple maximale
H_i	point de contact
L_i	point de départ
Q_m	coordonnées d'un point
WS	Espace de travail
WC	Espace de configuration
BM	Bitmap
GW	Grille de l'espace de travail
GC	Grille de l'espace de configuration

WSAPF	champ potentiel artificiel de l'espace de travail
CSAPF	champ potentiel artificiel de l'espace de configuration
δ	La distance entre deux points adjacents dans une grille
p	Point de control du robot
V_p	Champ potentiel de l'espace de travail
$U(q)$	Champ potentiel de l'espace de configuration
q	Position actuelle
S	Squelette de l'espace de travail
L	La distance aux obstacles
OB	obstacles

Introduction générale

La technique est un mot français qui provient du mot grec « *techné* », qui signifie ensemble de procédés, et de moyens pratiques propres à une discipline, à un domaine particulier [1]. C'est avec les techniques que les humains, à travers l'histoire, ont pu mettre en place des automatismes allant des plus simples aux complexes. Ces derniers font l'objet d'étude de l'automatique qui est un ensemble de « sciences et techniques de l'automatisation, du travail sans intervention humaine » [2]. La robotique est l'une des sciences de l'Automatique, une science et une technique permettant la conception et la construction de machines automatiques et de robots. Le robot est un appareil automatique capable de manipuler des objets ou d'exécuter des opérations pénibles pour l'homme selon un programme fixe, modifiable ou adaptable [3].

Le thème sur lequel nous avons travaillé est la planification de trajectoire sans collision pour un robot manipulateur en utilisant l'approche de champs de potentiels artificiels. A travers ce modeste travail nous allons essayer de donner des réponses aux questions suivantes :

Comment planifie-t-on les trajectoires pour robot lui permettant d'éviter les collisions contre les obstacles sur une scène de travail déterminée? Quelles sont les étapes qui caractérisent la méthode du champ potentiel ? Comment utilise-t-on la méthode du champ de potentiels pour atteindre la destination finale du robot manipulateur ? Quel est l'algorithme qui permet de sortir le robot des minimums locaux ?

Pour répondre à ces questions nous avons proposé la solution ci-après :

La méthode du champ potentiel artificiel serait la méthode la plus simple et efficace pour la planification de trajectoire pour robot manipulateur.

Notre mémoire comporte trois chapitres. Le premier chapitre traite du thème de la robotique en général. Le deuxième chapitre est consacré à la planification de trajectoires pour robots. Le chapitre 3 traite les résultats de simulations de la planification du chemin du robot en utilisant la méthode du champ de potentiel artificiel.

Chapitre I :
La robotique

I.1.Introduction

Le robot est une machine poly-articulée, dotée de possibilité d'auto-adaptabilité et qui peut agir d'une manière plus au moins autonome dans son environnement [4]. Pour la conception, la simulation ou la commande des robots, il est nécessaire de disposer d'un modèle de mécanisme. Dans ce chapitre on va présenter quelques types robots et les différentes modélisations.

I.2.Historique

L'histoire de l'humanité a connu des étapes et des époques qui ont marqué son évolution, allant de la préhistoire jusqu'au 21^{ème} siècle. Voici quelques inventions qui ont marqué cette évolution.

La roue : est le premier mécanisme apparue vers 3500 avant J-C, elle est considérée comme un premier exploit.

L'horlogerie, les clepsydres : appelée aussi horloges à eau, elle est inventée vers 1600 avant J-C. en Égypte. Elles étaient constituées de simples bols percés dont l'écoulement régulier permettait une mesure rudimentaire du temps.

Les automates : l'un des automates les plus surprenants est certainement le canard digérateur de Jacques de Vaucanson (1738), il était capable de se comporter comme un véritable animal.

L'intelligence artificielle : en 1950 Alan Turing a défini une méthode afin de tester la présence d'une conscience ou d'une intelligence au sein d'une machine.

Les animâts : ce sont des animaux artificiels inspirés des vivants, élaborés en 1985 par S. Wilson.

Asimo de Honda : robot développé par Honda en 2000, Asimo est capable de courir, sautiller sur un pied, courir à 9km/h et manipuler des objets [5].

Les drones : c'est un aéronef télécommandé. Le drone Metafly est comme un vrai oiseau. Il était accessible aux premiers acheteurs en septembre 2019 [6].

I.3. Classification des robots

Les robots peuvent être classés comme suit.

Les robots manipulateurs: Les manipulateurs sont des systèmes mécaniques destinés à exécuter de manière autonome des tâches répétitives dans un environnement fixe et

ordonné et dont les mouvements enregistrés dans une mémoire se répètent de manière cyclique. Les domaines d'application des manipulateurs sont : la peinture, la soudure par position, la manipulation des objets ...etc.

Les robots didactiques : Ils sont de version ou de format réduits. Ils assurent un rôle de formation et d'enseignement, ils peuvent être aussi utilisés pour exécuter des tests de faisabilité d'un poste robotisé.

Les robots mobiles autonomes: Ce sont des robots utilisés en zone dangereuse (nucléaire, incendie, sécurité civile, déminage... etc.), inaccessible (océanographique, spatiale) dotés de capteurs et de logiciels sophistiqués. On peut distinguer deux types de locomotions : les robots marcheurs qui imitent la démarche humaine, et les robots mobiles qui ressemblent à des véhicules.

Les télémanipulateurs : Le télémanipulateur est un système mécanique poly-articulé et multifonctionnel capable d'assister l'homme dans des opérations exécutées en milieu hostile comme il peut suppléer un handicap. Il est constitué de deux entités : un bras maître et un bras esclave. Le bras esclave reproduit instantanément le mouvement imposé par un opérateur à un bras maître. L'application des télémanipulateurs est répandue dans les domaines nucléaires, spatial, sous-marin (forage), militaire (déminage) et médical (prothèse).

I.4. Structure générale d'un robot manipulateur

Un robot manipulateur est constitué d'un ensemble de composants, ayant chacun un rôle bien spécifique, lesquels nous allons détailler ci-dessous :

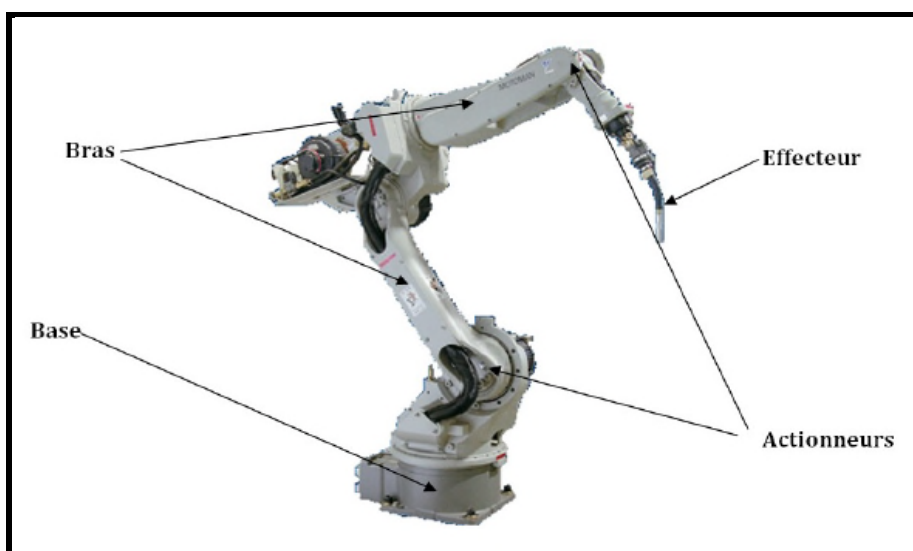


Figure I.1 : Structure générale d'un robot manipulateur

I.4.1. Système mécanique articulé

Un système mécanique articulé (SMA) est un ensemble d'éléments solides reliés entre eux par des liaisons animées avec des joints mécaniques.

Une articulation: c'est un système de motricité qui permet à un corps de se positionner selon les coordonnées attribuées.

- a. Articulation rotoïde :** Il s'agit d'une articulation de type pivot, notée R , réduisant le mouvement entre deux corps à une rotation autour d'un axe commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe.

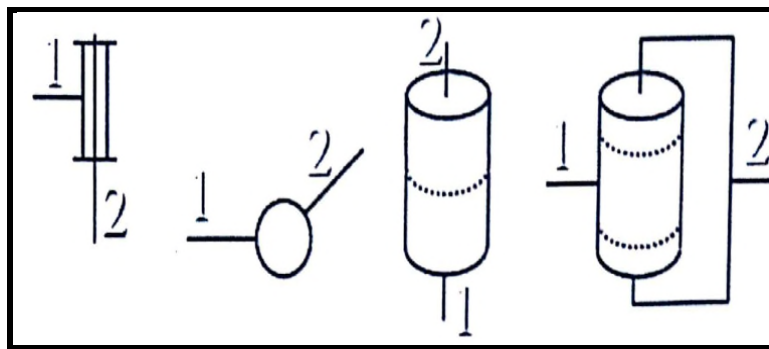


Figure I.2: Représentation d'une articulation rotoïde

- b. Articulation prismatique :** Il s'agit d'une articulation de type glissière, notée P , réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe.

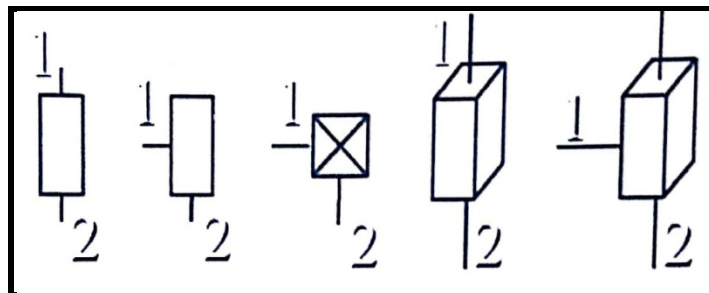


Figure I.3: Représentation d'une articulation prismatique

- c. Articulation rotule :** il s'agit d'une articulation de type sphérique notée S .

- d. Articulation cardan :** il s'agit d'une articulation de type joint universel notée U .

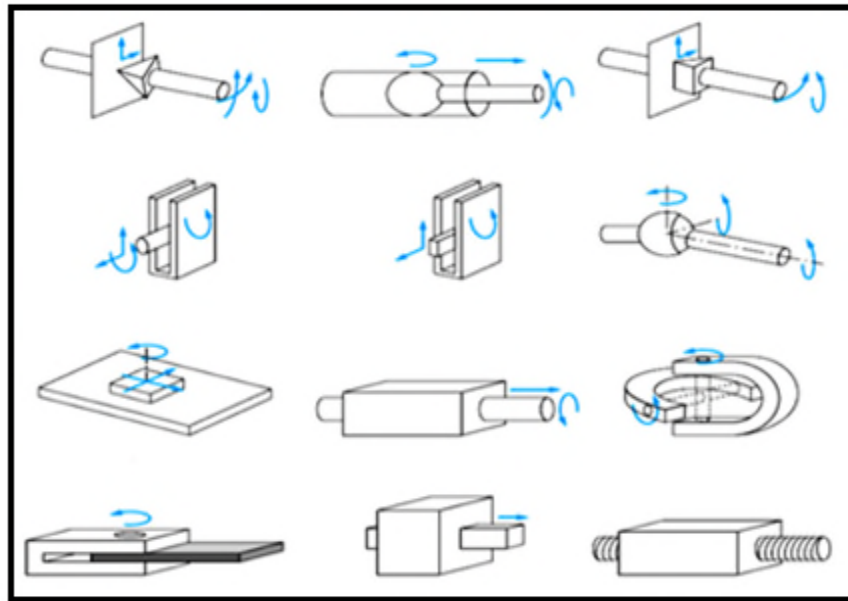


Figure I.4: Type d'articulations

I.4.2. Actionneurs ou organes de motorisation

Le terme actionneur désigne tout dispositif générateur d'effort à vitesse variable qui permet de modifier la configuration d'un robot manipulateur. Il est possible de les classer par :

- **Type de mouvement généré** : On trouve les actionneurs linéaires qui développent une force et génèrent un mouvement de translation parallèlement à cette force, et les actionneurs rotatifs qui développent un couple et génèrent un mouvement de rotation autour de l'axe du couple.
- **Nature de la source d'énergie** : On dispose d'actionneurs pneumatiques qui utilisent l'air comprimé comme source d'énergie, d'actionneurs hydrauliques sous pression, et d'actionneurs électriques qui utilisent l'énergie électrique.

I.4.3. Effecteur

L'effecteur est l'organe terminal du robot, il est fixé au poignet de celui-ci. Ce poignet se termine généralement par une plaque. En fait, la plupart des machines de production exigent des outils et des fixations spécialement conçus pour une application particulière.

I.4.4. Capteurs ou organes de perception

Un capteur est un organe de prélèvement d'information qui élabore à partir d'une grandeur physique une autre grandeur physique de nature différente très souvent électrique. Cette grandeur représentative de la grandeur prélevée est utilisable à des fins de mesures ou de commande. On distingue deux types de capteurs :

- **Les capteurs proprioceptifs** : Ils fournissent l'information sur la configuration du manipulateur tel que la position, la vitesse, l'accélération . . . etc., Pour assurer au robot manipulateur le contrôle de sa structure mécanique articulée, ils permettent à l'unité de commande de prendre la décision adéquate.
- **Les capteurs extéroceptifs** : ils interviennent lorsque l'espace de travail est mal connu, afin de donner les informations sur l'environnement extérieur du manipulateur comme la température, l'image . . .etc., Ils permettent de modifier le comportement du manipulateur pour s'adapter aux contraintes imposées.

I.5.Définitions

Nous allons donner quelques définitions.

I.5.1.Degré de liberté et redondance

Le positionnement complet d'une pièce dans l'espace nécessite six paramètres indépendants, appelés degré de liberté. Parmi les différents choix possibles de ces paramètres, on trouve les trois coordonnées cartésiennes pour positionner un point de la pièce et les trois angles pour l'orienter. Lorsque le nombre de degré de liberté de l'organe terminal est inférieur au nombre de degrés de liberté de l'espace articulaire, on dit qu'il y a une redondance.

I.5.2.Configuration singulière

La configuration singulière ou singularité se traduit physiquement par la nullité du déterminant de la matrice Jacobéenne. Ce qui implique :

- ✓ Les axes de deux articulations prismatiques deviennent parallèles.
- ✓ Les axes de deux articulations rotoïdes se trouvent sur la même droite.
- ✓ L'origine de l'organe terminal se trouve sur une ligne qui intersecte tous les axes des articulations.

I.5.3.Espace opérationnel

L'espace opérationnel est celui dans lequel la situation de l'organe terminal est représentée. La solution la plus simple consiste à utiliser les coordonnées cartésiennes. Soit R_m cet espace, la valeur m constitue donc le nombre de degré de liberté maximum que peut avoir l'organe terminal, et qui est égal au nombre de paramètres indépendants nécessaires pour décrire la situation de l'organe terminal dans l'espace cartésien.

I.5.4.Système de traitement

Le système de traitement gère l'ensemble des tâches. Il assure trois rôles essentiels :

1. Le rôle de l'information qui consiste à collecter l'information venant des capteurs.
2. Le rôle de la décision : en partant d'une tâche définie et en tenant compte des données du système et de l'environnement, il établit les actions adéquates
3. Le rôle de la communication

I.5.5.L'interface Homme-Machine

Une interface homme machine (IHM) est une interface utilisateurs permettant de connecter une personne à une machine, à un système ou à un appareil.[7]

I.5.6.La commande

Elle est constituée d'un ensemble d'ordres donnés par un programmeur pour le pilotage des actionneurs pour accomplir des tâches souhaitées, il faut générer une commande en tenant compte des paramètres suivant :

- Les valeurs de tension appliquées aux différents actionneurs.
- Les forces en sortie d'actionneurs.
- Les variations articulaires.
- Les variations opérationnelles (tâches à réaliser).

I.6.Matrices et transformations homogènes

Dans cette partie, nous allons parler des coordonnées homogènes et des transformations homogènes:

I.6.1. Coordonnées homogènes

Un point est représenté par P_x, P_y et P_z coordonnées cartésiennes :

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \quad (I.1)$$

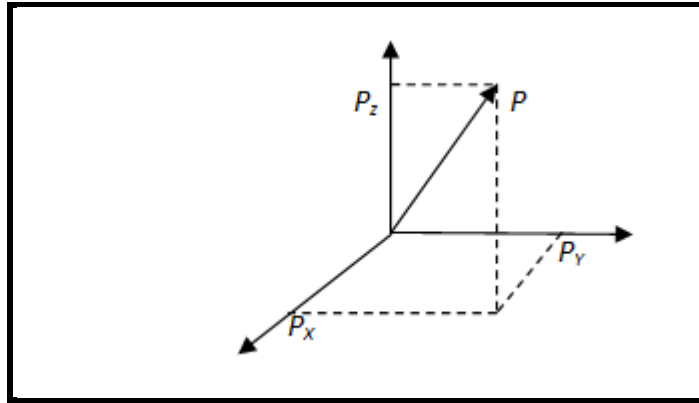


Figure I .5:Représentation d'un point dans l'espace

I.6.2. Transformations homogènes [8]

Un vecteur des coordonnées d'un point (c) dans un repère orthonormé $oc=(a,b,c)^T$ peut être écrit : $oc=(x,y,z,w)^T$ avec $x=w.a$, $y=w.b$ et $z=w.c$, w :facteur d'échelle (égale=1)ce qui permet de travailler avec des matrices $4X4$ incluant à la fois la rotation et la position.

$$T=^A T_B = \begin{bmatrix} R_{3X3} & P_{3X1} \\ N_{1X3} & 1 \end{bmatrix} = \begin{bmatrix} s_x & n_x & \alpha_x & P_x \\ s_y & n_y & \alpha_y & P_y \\ s_z & n_z & \alpha_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.2)$$

Avec :

$N=(0,0,0)$:représente la transformation de perspective.

$P_{(3X1)} = P_A^B$: est le vecteur de position de l'origine du repère mobile par rapport à (A).

R_{3X3} :est la matrice de rotation.

1 : est le facteur d'échelle.

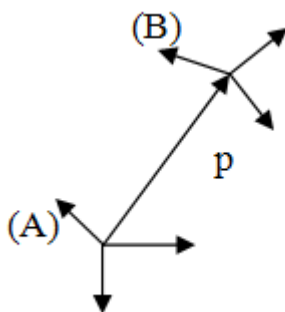
n_{3X1} : coordonnées de e_2 dans (A)

s_{3X1} : coordonnées de e_1 dans (A)

a_{3X1} coordonnées de e_3 dans (A)

(e_1, e_2, e_3) : vecteur unité du repère mobile(B)

Alors (n,s, α) représentent les axes de (B) dans (A) et P représente la position de l'origine de (B) par rapport à (A)



I.6.2.1. La rotation

La matrice de rotation par rapport à x d'un angle α est donnée par :

$$T_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.3)$$

La matrice de rotation par rapport à y d'un angle \emptyset est donnée par :

$$T_{y,\emptyset} = \begin{bmatrix} \cos \emptyset & 1 & 0 & \sin \emptyset & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -\sin \emptyset & 0 & \cos \emptyset & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.4)$$

La matrice de rotation par rapport à z d'un angle θ est donnée par :

$$T_{z,\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.5)$$

I.6.2.2. La translation

La matrice de translation d'un vecteur (a,b,c) le long des différentes axes d'un repère est donnée par :

$$T_{tr} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.6)$$

I.7. Modélisation

La conception, la simulation et la commande d'un robot exigent le calcul de certains modèles mathématiques [9].

1. **Les modèles de transformation** entre l'espace articulaire et l'espace opérationnel sont très importants puisque les robots sont commandés dans l'espace articulaire, tandis que les tâches sont définies dans l'espace opérationnel.

On peut y distinguer deux types de modèles :

- ✓ **Les modèles géométriques directs et inverses** qui donnent l'endroit de l'effecteur en fonction des variables articulaires du mécanisme et vice versa.

- ✓ **Les modèles cinématiques directs et inverses** qui donnent la vitesse de l'effecteur en fonction des vitesses d'articulation et vice versa.

2. **Les modèles dynamiques** donnant les relations entre les couples d'entrées en l'occurrence les forces des actionneurs et les positions, ou les vitesses et les accélérations des articulations.

I.7.1. Modélisation géométrique [8]

Il s'agit de deux modèles, en l'occurrence, le modèle géométrique direct et le modèle géométrique inverse :

I.7.1.1. Modèle géométrique direct

Pour pouvoir effectuer le calcul de l'équation géométrique directe d'un bras manipulateur à corps rigides et sans jeux au niveau des articulations nous devons recourir à une méthode systématique et générale qui doit être appliquée afin de définir la position et l'orientation relatives de deux corps consécutifs, la méthode de Denavit-Hartenberg est la plus répandue.

Convention DH

C'est une méthode matricielle permettant d'établir systématiquement un repère pour chaque corps d'une chaîne articulée, on peut représenter le repère associé à chaque corps au niveau de l'articulation par rapport au repère précédent. On peut aussi situer le poignet par rapport au repère de base R_0 .

Afin d'établir les paramètres DH , chacun des repère est établi en se basant sur trois règles :

-L'axe Z_{i-1} est selon l'axe de mouvement de la $i^{\text{ème}}$ articulation.

-L'axe X_{i-1} est aligné entre les articulations i et $(i-1)$, qui est perpendiculaire à Z_{i-1} et Z_i (si $a_i=0$).

-L'axe Y_{i-1} complète le repère i pour former un repère « main droite ».

Pour passer de R_{i-1} à R_i on peut suivre le chemin suivant :

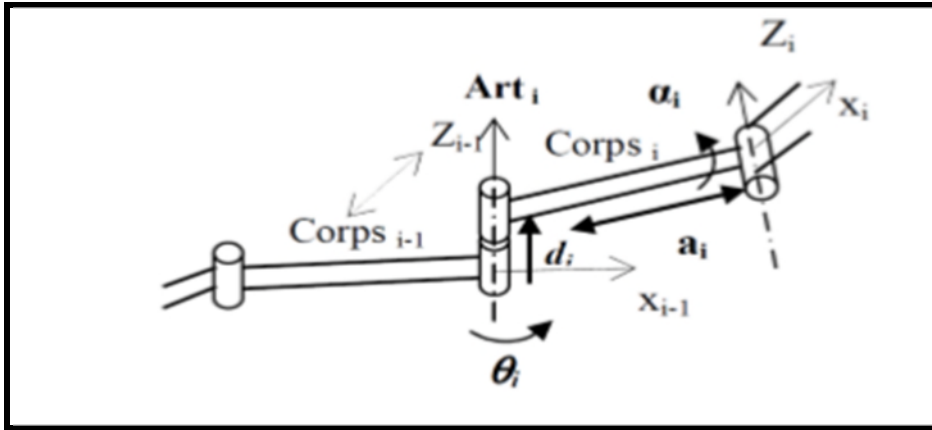


Figure I .6: Association des repères selon la convention D-H [8]

a_i : La distance entre l'axe z_{i-1} et l'axe z_i le long de x_i

α_i : L'angle entre l'axe z_{i-1} et l'axe z_i autour de x_i

θ_i : L'angle entre les axes x_{i-1} et x_i autour de z_i

d_i : La distance entre les axes x_{i-1} et x_i le long de l'axe z_i

Le repère R_{i-1} est placé à la fin du corps ($i-1$).

La matrice de passage d'une articulation à l'articulation suivante est :

$$T = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.7)$$

Le MGD d'un bras de robot est obtenu en utilisant le produit $T_{MG} = T_0^n = T_0^1 \cdot T_1^2 \dots T_{n-1}^n$

Convention DH modifiée (MDH)

C'est une configuration dont le repère est associé au début du corps.

a_i : La distance entre l'axe z_i et l'axe z_{i+1} , le long de x_i .

α_i : L'angle entre l'axe z_i et l'axe z_{i+1} , autour de x_i .

θ_i : L'angle entre les axes x_{i-1} et x_i , autour de z_i .

d_i : La distance entre les axes x_{i-1} et x_i le long de l'axe z_i .

La matrice de passage d'une articulation à l'articulation suivante est :

$$T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \cos\alpha_{i-1} \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_{i-1} & -d_i \sin\alpha_{i-1} \\ \sin\alpha_i \sin\theta_i & \sin\alpha_i \cos\theta_i & \cos\alpha_{i-1} & d_i \cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.8)$$

I.7.1.2. Modèle géométrique inverse

Le modèle géométrique inverse *MGI* consiste à définir les variables articulaires qui correspondent à une situation spécifique de l'organe terminal, lorsqu'elles existent.

Trois méthodes de calcul de *MGI* sont connues [9] :

- 1 La méthode de Paul, qui convient pour la plupart des robots industriels.
- 2 La méthode de Pieper, qui permet de résoudre le problème pour les robots à six degrés de liberté avec trois articulations rotoïdes ou trois articulation prismatiques.

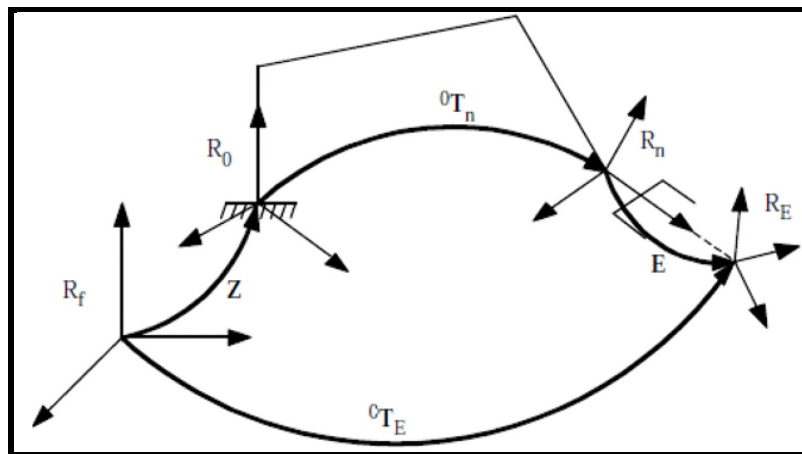


Figure 1.7: Transformations entre l'organe terminal et le repère atelier

La méthode de Paul

Cette méthode est basée sur l'identification des variables articulaires, en connaissant la matrice W_0 décrivant la situation du poignet, alors :

Exemple :

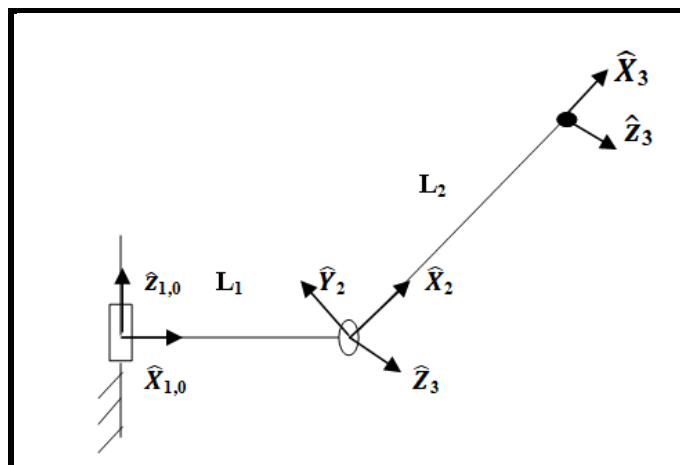


Figure I.8 : Bras à deux articulations

Tableau I.1. Convention MDH.

Articulation	α	a	d	θ
$R_0 \longrightarrow R_1$	0	0	0	θ_1
$R_1 \longrightarrow R_2$	90	L_1	0	θ_2
$R_2 \longrightarrow R_3$	0	L_2	0	0

$$W_0 = T_0^1 T_1^2 T_2^3 \text{ et } T_1^0 \cdot W_0 = T_1^2 T_2^3$$

Le 1^{er} terme ne dépend que de θ_1 et le 2^{ème} est indépendant de θ_1 . Après l'isolation, on identifie θ_1 .

$T_2^1 W_1 = T_2^3$ d'où θ_2 de la même façon on calcul les autres angles.

Méthode de Pieper :

C'est une méthode qui sert à la définition de l'équation de position et d'orientation.

Équation de position :

L'effecteur dépend de $(\theta_1, \theta_2, \theta_3)$. En comparant le vecteur de position de poignet avec la dernière colonne de la matrice du MGD, on peut calculer $(\theta_1, \theta_2, \theta_3)$.

Équation d'orientation :

En utilisant les valeurs calculées de $(\theta_1, \theta_2, \theta_3)$, on peut calculer les termes restant (à s'avoir le nombre de d_{ll}).

Soit R_s est la matrice d'orientation actuelle du poignet on écrit :

$$R_s = R_0^1 R_1^2 R_2^3 = R_0^2 R_2^3 \text{ alors } R_2^0 R_s = R_2^3$$

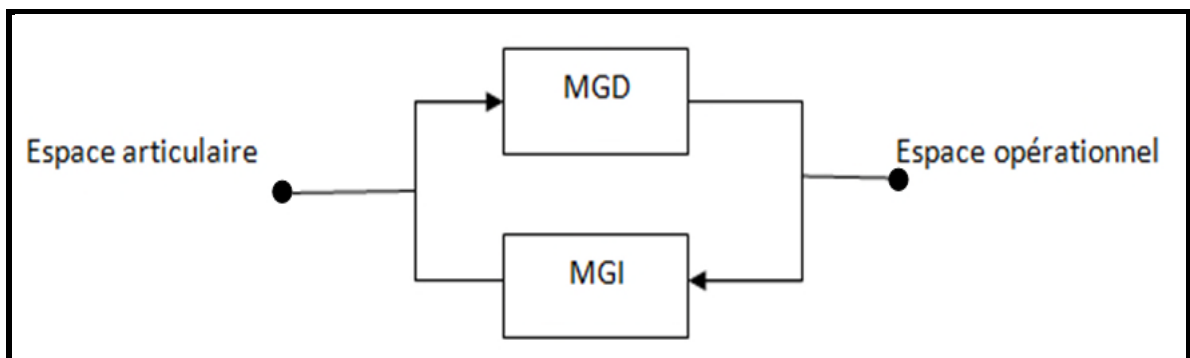


Figure I.9 : Modèle géométrique direct et inverse

I.7.2. Modélisation cinématique[8]

Il s'agit de deux modèles, en l'occurrence, le modèle cinématique direct et le modèle cinématique inverse :

I.7.2.1. Modèle cinématique direct

Le modèle cinématique direct *MCD* d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires [4]. Il s'écrit :

$$\dot{X} = J(q)\dot{q} \quad (I.9)$$

Où, $J(q)$ désigne la matrice Jacobéenne de dimension $m \times n$ du mécanisme et qui est égale

$$\frac{\partial X}{\partial \dot{q}}$$

\dot{X} : vitesse opérationnelle.

\dot{q} : vitesse articulaire.

I.7.2.1.1. Matrice de transformation différentielle

La translation et la rotation différentielles par rapport au repère de base est :

$$T+dT=Trans(d_x, d_y, d_z).Rot(k, d\theta).T \text{ Alors } T=[Trans(d_x, d_y, d_z).Rot(k, d\theta)-I]. T=\Delta.T$$

$Trans(d_x, d_y, d_z)$: translation différentielle par rapport aux axes du repère de base.

$Rot(k, d\theta)$: est la matrice de rotation différentielle autour de l'axe k du repère de base.

$dT=\Delta.T$: variation différentielle de matrice T .

Quand $\theta \rightarrow 0$ alors $\sin\theta \approx \theta$ et $\cos\theta \approx 1$, les matrices de rotation différentielles données par :

$$T_x \delta_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\delta_x & 0 \\ 0 & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.10)$$

$$T_y \delta_y = \begin{bmatrix} 1 & 0 & \delta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\delta_y & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.11)$$

$$T_z \delta_z = \begin{bmatrix} 1 & -\delta_z & 0 & 0 \\ \delta_z & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.12)$$

La matrice de rotation différentielle globale pour des rotations $(\delta_x, \delta_y, \delta_z)$, réalisées dans n'importe quel ordre autour des axes (X, Y, Z) du repère de base, en négligeant les termes produits :

$$T_{rot}(\delta_x, \delta_y, \delta_z) = \begin{bmatrix} 1 & -\delta_z & \delta_y & 0 \\ \delta_z & 1 & -\delta_x & 0 \\ -\delta_y & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.13)$$

La matrice globale :

$$\Delta = [Trans(d_x, d_y, d_z) \cdot Rot(k, d_\theta) - I] = \begin{bmatrix} 0 & -\delta_z & \delta_y & d_x \\ \delta_z & 0 & -\delta_x & d_y \\ -\delta_y & \delta_x & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (I.14)$$

La transformation Δ est construite des deux vecteur $dP_n = (d_x, d_y, d_z)^T$ et $dr_n = (\delta_x, \delta_y, \delta_z)^T$

Le vecteur $D = (d_x, d_y, d_z, \delta_x, \delta_y, \delta_z)^T$.

Exemple : Soit le robot planaire, sa matrice de transformation selon la convention DH est donnée par :

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_i \cdot \cos\theta_i \\ \sin\theta_i & \cos\theta_i & 0 & a_i \cdot \sin\theta_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.15)$$

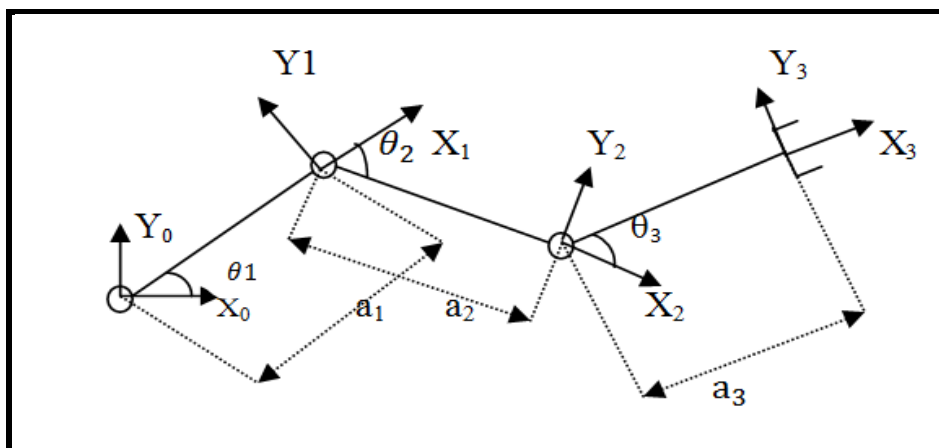


Figure I.10 robot planaire RRR dont les repères ont été placés selon la convention D-H

Le MGD :

$$T_{0^3} = \begin{bmatrix} c123 & -s123 & 0 & a3.c123 + a2.c12 + a1.c1 \\ s123 & c123 & 0 & a3.s123 + a2.s12 + a1.s1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Le changement de position pour une variation $\delta_z = \delta\theta_1$:

$$dP = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} -a3.s123 - a2.s12 - a1.s1 \\ a3.c123 + a2.c12 + a1.c1 \\ 0 \end{bmatrix} \delta\theta_1$$

La matrice Δ avec la variation $\delta_z = \delta\theta_1$ est:

$$\Delta = \begin{bmatrix} 0 & -\delta_z & \delta y & d_x \\ \delta_z & 0 & -\delta x & d_y \\ -\delta y & \delta x & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & -a3.s123 - a2.s12 - a1.s1 \\ 1 & 0 & 0 & a3.c123 + a2.c12 + a1.c1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \delta\theta_1$$

Le déplacement différentiel :

$$D_{\theta_1} = (-a3.s123 - a2.s12 - a1.s1, a3.c123 + a2.c12 + a1.c1, 0, 0, 0, 1)^T \cdot \delta\theta_1$$

De la même façon, on peut calculer le déplacement différentiel causé par $\delta\theta_2$ et $\delta\theta_3$

$$D_{\theta_2} = (-a3.s123 - a2.s12, a3.c123 + a2.c12, 0, 0, 0, 1) \delta\theta_2$$

$$D_{\theta_3} = (-a3.s123, a3.c123, 0, 0, 0, 1) \delta\theta_3$$

Le mouvement différentiel total :

$$\begin{bmatrix} d_x \\ d_y \\ d_z \\ \delta_x \\ \delta_y \\ \delta_z \end{bmatrix} = \begin{bmatrix} -a3.s123 - a2.s12 - a1.s1 & -a3.s123 - a2.s12 & -a3.s123 \\ a3.c123 + a2.c12 + a1.c1 & a3.c123 + a2.c12 & a3.c123 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_{\theta 1} \\ \delta_{\theta 2} \\ \delta_{\theta 3} \end{bmatrix}$$

$$= J \cdot \begin{bmatrix} \delta_{\theta 1} \\ \delta_{\theta 2} \\ \delta_{\theta 3} \end{bmatrix} \quad (I.16)$$

Le calcul des vitesses articulaires en fonction de la vitesse dans l'espace de travail se fait en utilisant le modèle inverse.

I.7.2.1.2.Méthode de propagation des vitesses :

Nous allons traiter de la vitesse angulaire et de la vitesse linéaire :

➤ Vitesses angulaires :

La vitesse angulaire de corps C_{i+1} est égale à celle du corps C_i plus la vitesse de rotation au niveau de l'articulation $(i+1)$.

Donnée par : (dans le repère R_i)

$$W_i^{i+1} = W_i^i + R_i^{i+1} \cdot \dot{\theta}_{i+1} \cdot Z_{i+1}^{i+1} \quad (I.17)$$

Et donnée par : (dans le repère R_{i+1})

$$W_{i+1}^{i+1} = R_{i+1}^i \cdot W_i^i + \dot{\theta}_{i+1} \cdot Z_{i+1}^{i+1} \quad (I.18)$$

W_i^{i+1} : Vitesse angulaire du corps C_{i+1} exprimée dans R_i

W_i^i : Vitesse angulaire du corps C_i exprimée dans R_i

R_i^{i+1} : Matrice de rotation du repère R_{i+1} par rapport à R_i

$$\dot{\theta}_{i+1} \cdot Z_{i+1}^{i+1} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix} \quad (I.19)$$

$\dot{\theta}_{i+1}$: un scalaire.

Z_{i+1}^{i+1} : axe de rotation.

➤ **Vitesses linéaires**

La vitesse linéaire de l'origine du repère R_{i+1} est la même que celle de l'origine du repère R_i plus une composante produite par la rotation de corps C_i .

Donnée par : (dans R_i)

$$V_i^{i+1} = V_i^i + W_i^i X P_i^{i+1} \quad (I.20)$$

Et donnée par : (dans le repère R_{i+1})

$$V_{i+1}^{i+1} = R_{i+1}^i (V_i^i + W_i^i X P_i^{i+1}) \quad (I.21)$$

P_i^{i+1} : Vecteur aligné le long du corps C_i .

$W_i^i X P_i^{i+1}$: Produit vectoriel qui correspond à la vitesse linéaire et l'origine du repère

R_{i+1} : Induite par la rotation du corps C_i à la vitesse W_i^i .

I.7.2.2. Modèle cinématique inverse

C'est le modèle cinématique inverse *MCI* qui indique les vitesses articulaires \dot{q} qui correspondent à une vitesse désirée \dot{X} de l'organe terminal [4]. On a besoin de calculer J^{-1} .

❖ **Cas de régularité :**

La matrice J est carrée et non singulière, on peut alors calculer J^{-1}

$$\text{On a : } \dot{q} = J^{-1} \cdot \dot{X} \quad (I.22)$$

❖ **Cas de singularité :**

- Cas de singularité permanente ($\det(J)$ est toujours nul) : cas purement théorique.

- Cas de configuration singulière ($\det(J)$ est nul pour quelques valeurs

Dans ce dernier cas, on distingue deux types de singularités :

- Celles liées à la frontière de l'espace de travail : quand le bras est complètement étiré ou plié.

- Celles liées à l'intérieur de l'espace de travail dues de la redondance.

I.7.3. Modélisation dynamique

C'est la relation entre les couples (ou forces) appliquées aux actionneurs et les positions, vitesses et accélérations articulaires [10]. Il est représenté par la relation suivante :

$$\Gamma = f(q, \dot{q}, \ddot{q}, f_e) \quad (I.23)$$

Avec :

- Γ : vecteur des couples des actionneurs, selon que l'articulation est rotoïde ou prismatique.

- q : vecteur des positions articulaires.

$-\dot{q}$: vecteurs des vitesses articulaires.

$-\ddot{q}$: vecteurs des accélérations articulaires.

$-f_e$: vecteurs représentant l'effort extérieur qu'exerce le robot sur l'environnement.

La relation (1.1) est appelée modèle dynamique inverse ou modèle dynamique parce qu'elle définit le système d'entrée en fonction des variables se rattachant aux sorties.

Le modèle dynamique direct décrit les accélérations articulaires en fonction des positions, vitesses et couples. Il est représenté par la relation suivante :

$$\ddot{q} = g(q, \dot{q}, \Gamma, f_e) \quad (I.24)$$

Le modèle dynamique joue un rôle important dans la conception et le fonctionnement du robot. Le modèle dynamique inverse pourrait être utilisé pour le choix des actionneurs, alors que le modèle dynamique direct est utilisé pour faire des simulations, dans le but de tester les performances du robot. En ce qui se rattache aux fonctionnalités du robot, le *MDI* pourrait être utilisé pour calculer les couples d'actionneurs nécessaires pour la réalisation d'un mouvement souhaité. Il sert également pour l'identification des paramètres dynamiques qui sont nécessaires à la fois pour le contrôle et la simulation [9]. Le modèle dynamique des robots peut être obtenu en recourant à plusieurs approches. Les plus utilisées sont le formalisme de Lagrange, et le formalisme de Newton-Euler.

L'approche de Newton-Euler prend en considération les forces et les moments agissants entre les liens. La formulation de Newton-Euler est une approche basée sur l'équilibre des forces, la formulation de Lagrange est basée sur l'énergie. Mais pour le même manipulateur, les deux donnent les mêmes équations concernant le mouvement [11]. Dans notre étude nous utiliserons l'approche formaliste de Lagrange et on considèrerait que le robot à chaîne ouverte simple.

I.7.3.1. Formalisme de Lagrange

On considère un système idéal sans frottement ou élasticité, n'exerçant ni des forces ni des moments sur l'environnement. La formulation de Lagrange décrit le comportement d'un système dynamique en termes d'énergie [12]. Les équations de Lagrange sont généralement écrites sous la forme :

$$\Gamma = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \quad (I.25)$$

Où L est le Lagrangien du système, et qui se définit comme la différence entre l'énergie cinétique K et l'énergie potentielle P :

$$L = K - P \quad (I.26)$$

I.7.3.2. Vitesse des articulations

Soit r_i^i : les coordonnées d'un point fixé sur le corps i , exprimé dans le repère R_i

$$r_i^i = (x_i, y_i, z_i, 1)^T$$

Dans le repère de base R_0 on a :

$$r_0^i = T_0^i \cdot r_i^i$$

$$T_0^i = T_0^1 \cdot T_1^2 \dots \cdot T_{i-1}^i$$

Pour une articulation rotoïde :

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.27)$$

Pour une articulation prismatique

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & 0 \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.28)$$

$$V_0^i = \frac{dr_0^i}{dt} = \frac{d(T_0^i \cdot r_i^i)}{dt} \quad (I.29)$$

I.7.3.3. Énergie cinétique

L'énergie cinétique k_i du i^{ieme} corps est donnée par l'expression :

$$k_i = \frac{1}{2} m_i v^T C_i v C_i + \frac{1}{2} \omega_i^T C_i I_i \omega_i \quad (I.30)$$

Où le premier terme est l'énergie cinétique due à la vitesse linéaire et le deuxième terme est l'énergie cinétique due à la vitesse angulaire. L'énergie cinétique totale du manipulateur est la somme de l'énergie cinétique des différentes articulations c'est-à-dire :

$$K = \sum_{i=1}^n k_i \quad (I.31)$$

Les vitesses ${}^v C_i^T$ et ${}^i \omega_i^T$ sont des fonctions de q et \dot{q} respectivement, donc on voit que l'énergie cinétique d'un manipulateur peut être décrite par une formule en fonction de la position et de la vitesse, $K(q, \dot{q})$. L'énergie cinétique d'un manipulateur est donnée par la formule suivante:

$$K(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (I.32)$$

Où $M(q)$ est une matrice $n \times n$ de l'énergie cinétique appelée matrice d'inertie. Une expression de la forme ci-dessus est appelée la forme quadratique. En effet, l'énergie cinétique doit être toujours positive, pour cela la matrice d'inertie du manipulateur doit être une matrice positive.

I.7.3.4. Énergie potentielle

L'énergie potentielle de la $i^{\text{ème}}$ articulation, s'exprime à travers l'expression suivante :

$$P_i = m_i {}^0\mathbf{g}^T {}^0T_{Ci} + P_{refi} \quad (I.33)$$

Où ${}^0\mathbf{g}$ est le vecteur de gravité 3×1 , 0T_C une transformation homogène localisant le centre de la $i^{\text{ème}}$ articulation, et P_{refi} est une constante choisie de manière à ce que la valeur minimum de P_i soit 0. L'énergie potentielle totale est la somme de l'énergie potentielle dans les différentes articulations, ce qui s'exprime à travers la formule suivante:

$$P = \sum_{i=1}^n p_i \quad (I.34)$$

Puisque 0T_C est en fonction de q_1, q_2, \dots, q_n l'énergie potentielle est en fonction de la position de l'articulation, $P(q)$. Par conséquent, le Lagrangien sera :

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q) \quad (I.35)$$

I.7.3.5. Équation du mouvement d'un robot manipulateur

En se reportant à l'équation (1.35) le Lagrangien a la forme :

$$L(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - P(q) \quad (I.36)$$

Où $M(q)$ est la matrice d'inertie du manipulateur et $P(q)$ est l'énergie potentielle due à la gravité. Il convient d'écrire l'énergie cinétique sous forme d'une somme :

$$L(q, \dot{q}) = \frac{1}{2} \sum_{i,j=1}^n M_{ij}(q) \dot{q}_i \dot{q}_j - P(q) \quad (I.37)$$

En procédant à une substitution dans l'équation de Lagrange, les équations de mouvements sont données par :

$$F_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i}$$

Où F_i sont les forces (couples) agissant sur l'articulation i , en utilisant l'équation. (I.37)

On a ainsi :

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} = \frac{d}{dt} \left(\frac{1}{2} \sum_{j=1}^n M_{ij}(q) \dot{q}_j \right) = \frac{1}{2} \left(\sum_{j=1}^n \dot{M}_{ij}(q) \dot{q}_j + M_{ij}(q) \ddot{q}_j \right) \quad (I.38)$$

$$\frac{\partial L}{\partial \dot{q}_i} = \frac{1}{2} (\sum_{j=1}^n M_{ij}(q) \dot{q}_j - \frac{\partial P(q)}{\partial \dot{q}_i}) \quad (I.39)$$

Le terme $\dot{M}(q)$ peut être augmenté en termes de dérivés partiels.

$$\Gamma_i = \frac{1}{2} \sum_{j=1}^n M_{ij}(q) \ddot{q}_j + \frac{1}{2} \sum_{j,k=1}^n \left(\frac{\partial M_{ij}(q)}{\partial q_k} \dot{q}_j \dot{q}_k - \frac{\partial M_{kj}(q)}{\partial q_i} \dot{q}_k \dot{q}_j \right) + \frac{\partial P(q)}{\partial q_i} \quad (I.40)$$

Afin de mettre les équations du mouvement sous forme d'un vecteur, on définit la matrice $C(q; \dot{q})$, telle que :

$$C_{ij}(q, \dot{q}) = \frac{1}{2} \sum_{k=1}^n \left(\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{kj}}{\partial q_i} \right) \dot{q}_k \quad (I.41)$$

L'équation de mouvement peut être écrite comme :

$$\Gamma = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) \quad (I.42)$$

Où :

Γ : Vecteur des couples appliqués aux articulations.

$M(q)$: Matrice d'inertie du robot.

$C(q, \dot{q}) \dot{q}$: Vecteur des forces de Coriolis et des forces centrifuges.

$G(p)$: Vecteur des forces de gravité.

Pour un système avec frottement ou élasticité, le modèle dynamique est :

$$\Gamma = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F_e \quad (I.43)$$

Il s'agit d'une équation différentielle du second ordre pour le mouvement du manipulateur en fonction des couples articulaires appliqués[13].

I.8. Conclusion

A travers ce chapitre nous avons pu découvrir quelques notions de base du domaine de la robotique. Nous y avons donné quelques définitions générales de base et des notions qui permettent de comprendre la constitution et le fonctionnement d'un robot manipulateur. Une partie de ce chapitre est consacrée pour la modélisation des robots manipulateurs, cette modélisation aide à la conception, elle est utilisée pour la prédiction des mouvements, l'adaptation des actionneurs et la planification des tâches. Le formalisme de Lagrange est utilisé pour l'obtention de l'équation de mouvement du robot manipulateur.

Chapitre II :
Planification de trajectoire

II.1. Introduction

Le problème de planification de trajectoire est toujours associé à l'architecture et à la conception des robots. Ainsi, chaque robot dispose de son propre planificateur de trajectoire selon sa conception et les différentes tâches qui lui sont attribuées. Dans ce chapitre, nous présentons les différents concepts et techniques de planification de trajectoires des robots manipulateurs. Le terme de planification peut prendre plusieurs formes selon le contexte d'utilisation. Dans ce chapitre, nous présentons le problème de planification de trajectoire des robots manipulateurs et ses concepts de base ainsi que la démarche générale de résolution d'un problème de planification. Nous donnons aussi une classification des techniques de planification avec quelques algorithmes d'évitement d'obstacles.

II.2. Formes de planification

Voici quelques formes de planification :

- 1. La planification de mouvement** : elle a pour but de calculer un chemin sans collision en prenant en compte la géométrie et la cinématique.
- 2. La planification sensorielle** : elle vise à trouver les informations nécessaires pour la tâche encours au bon moment, à sélectionner le bon capteur pour la réception de données et à définir une méthodologie de récupération.
- 3. La planification pour la navigation** : son objectif est d'organiser un ensemble de primitives de localisation et de mouvements asservis sur des capteurs en vue d'atteindre un but ou d'explorer un environnement.
- 4. La planification pour la manipulation** : Elle consiste à élaborer une stratégie avec des primitives de prise, de manipulation, de reconnaissance d'objets et d'assemblage en utilisant des retours sensoriels.
- 5. La planification pour la communication** : elle vise à organiser les requêtes et le retour d'interaction (avec l'homme et encas de collaboration ou de coordination multi-robots) pour l'activité encours.
- 6. La planification des tâches** : elle a pour objectif d'organiser l'ensemble des activités du robot dans le temps et leur attribuer des ressources, compte tenu des évolutions prévisibles dans l'environnement.

II.3. Planification de trajectoire

Le but principal de la planification de trajectoire est de trouver une solution optimale en évitant toute collision et donc élaborer une trajectoire afin que le robot puisse accomplir une tâche particulière. En effet, le problème général de la planification de trajectoires est très complexe, car il est déterminé par plusieurs contraintes. La planification de trajectoires est déterminée par l'environnement de travail du robot. Dans le cadre du déplacement du robot dans un environnement connu, ses mouvements peuvent se résumer en une série de tâches bien précises. Cependant, la connaissance de l'environnement ne facilite pas pour autant au robot la connaissance du chemin à emprunter. En conséquence, le problème de planification est souvent décomposé en plusieurs étapes comme on peut le voir dans le schéma ci-dessous :

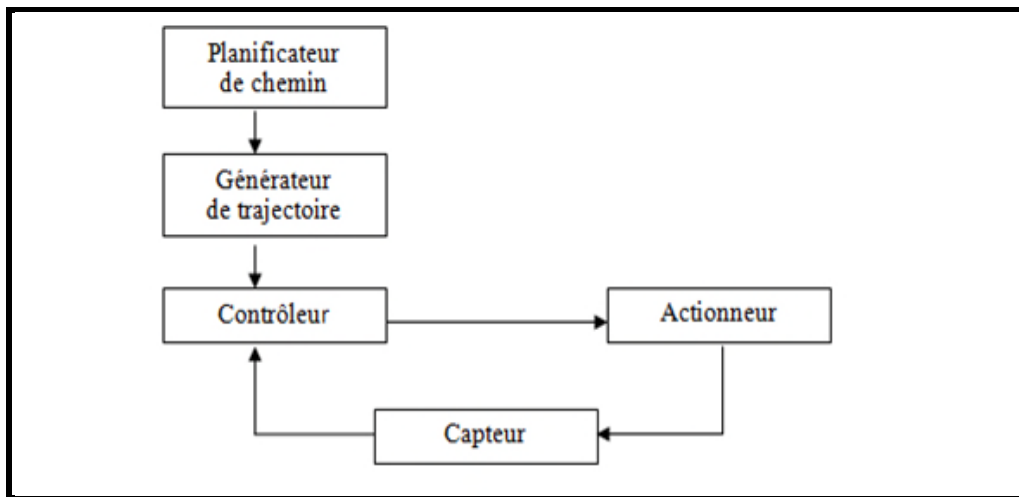


Figure II.1 :Planification et génération de trajectoires

Dans ce schéma, la première étape est relative à la planification d'un chemin traçant l'itinéraire désiré. Dans la majorité des cas, elle consiste à prendre des décisions en fonction de la tâche à réaliser et de l'environnement de travail du robot.

La deuxième étape est relative à la détermination et à la génération d'une trajectoire qui permet le déplacement du robot le long de son chemin. Les données de la planification sont exprimées sous la forme d'une trajectoire définie par une suite de repères constitués par les points de passage correspondant aux situations successives de l'effecteur.

La dernière étape consiste à suivre cette trajectoire à partir de capteurs et d'actionneurs. Elle se résume souvent à une boucle d'asservissement qui permet la régulation de la trajectoire du robot par rapport à la trajectoire idéale [14], comme il est illustré ci-dessous :

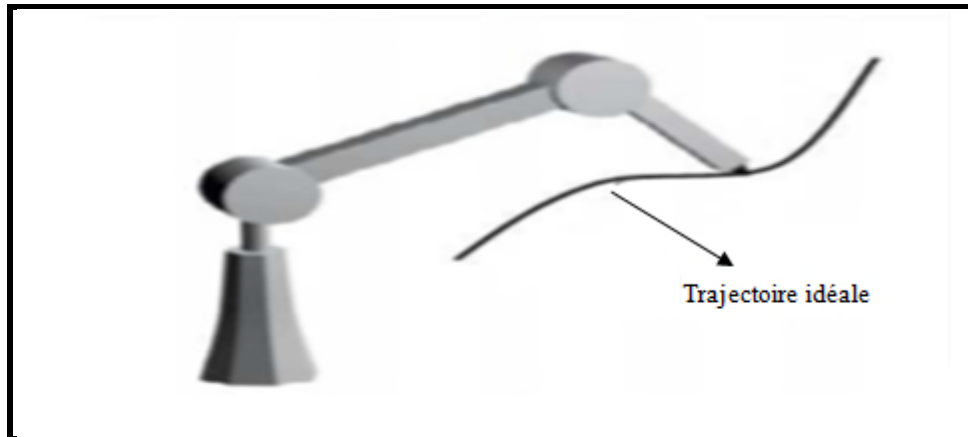


Figure II.2 : suivi d'une trajectoire

II.3.1. Planification dans l'espace articulaire

L'espace articulaire est caractérisé par les variables articulaires. Le modèle cinématique et dynamique du robot est souvent exprimé dans cet espace. La génération et le suivi de trajectoire peuvent se faire directement dans cet espace. Cela se traduit par une séquence de positions articulaires qui constituent les consignes des asservissements comme il est indiqué ci-dessous :

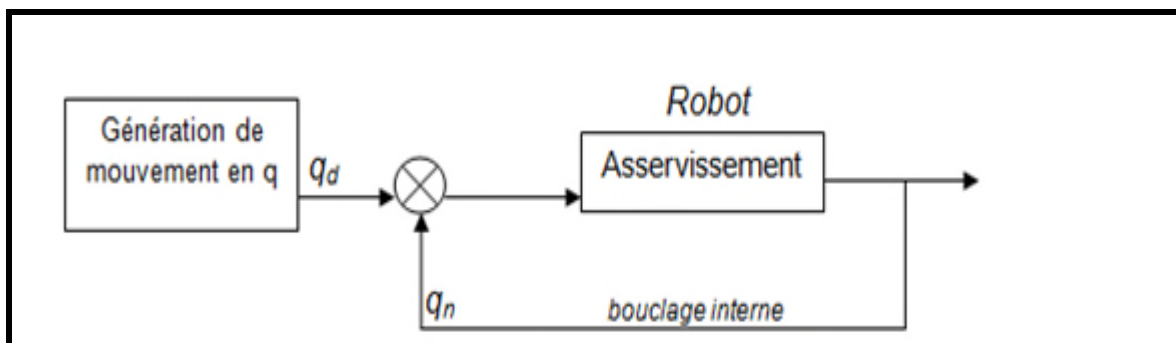


Figure II.3: Planification dans l'espace articulaire

II.3.2. Planification dans l'espace cartésien

Cette approche consiste à travailler dans l'espace cartésien où l'effecteur du robot évolue. Il faudrait donc exprimer la trajectoire dans cet espace. Les coordonnées opérationnelles (cartésiennes) doivent alors être transformées en coordonnées articulaires par le modèle géométrique inverse du robot (MGI). Il est donc facile d'exprimer une tâche robotique en fonction de la géométrie de déplacement du robot. Le temps de calcul des modèles géométriques peut être très prohibitif. C'est ce qu'illustre le schéma suivant :

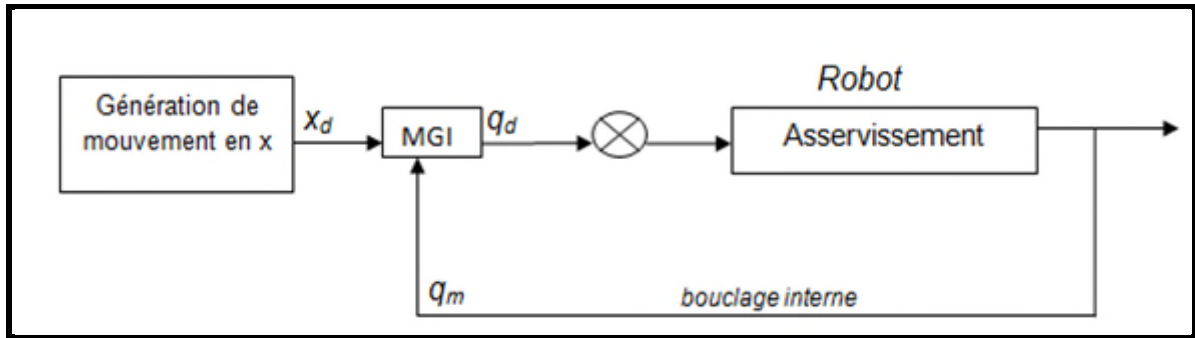


Figure II.4 :Planification dans l'espace cartésien

Un autre espace de travail peut être défini si l'on dispose d'une autre source d'information. En effet, si on se place dans le cas où l'on a de l'information visuelle, on peut travailler dans un nouvel espace qui est plus parlant que les deux précédents. Cet espace est l'espace image dans lequel la trajectoire à suivre est traduite en un motif visuel, comme il est indiqué ci-dessous :

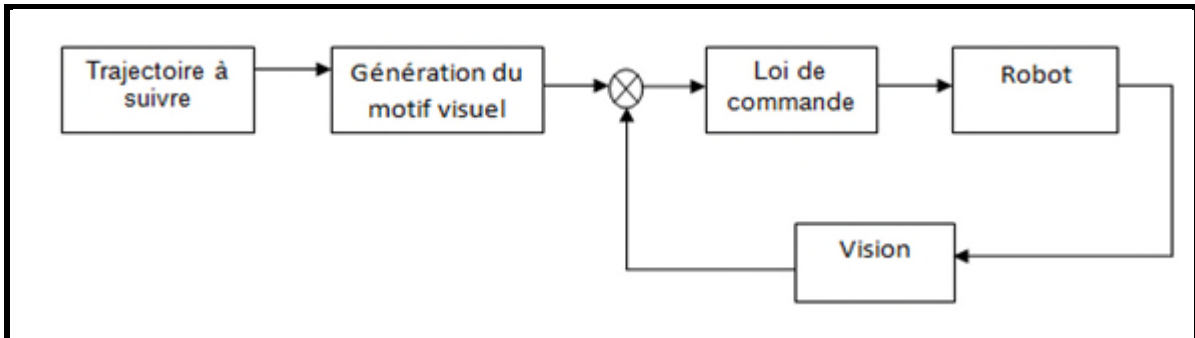


Figure II.5 : Planification dans l'espace image

II.4.Résolution du problème de planification

La démarche générale de la résolution d'un problème de planification peut être faite en quatre étapes :

1. Saisir et comprendre les données de ce problème.
2. Établir une représentation permettant la découverte d'une solution.
3. Calculer une solution dans la nouvelle représentation.
4. Rétablir la solution dans la représentation d'origine où l'on peut vérifier sa validité.

II.5. Classification des méthodes de planification

Le problème de planification de trajectoire est largement traité dans la littérature et plusieurs classifications peuvent être établies [15]. Essentiellement, on peut distinguer deux classifications : la première est relative à l'environnement de travail du robot, c'est-à-dire aux informations prises lors du calcul de la trajectoire. La deuxième classification concerne la technique utilisée pour la résolution du problème.

II.5.1. Classification basée sur l'environnement

Pour avoir des informations sur l'espace de travail du robot manipulateur, on peut soit prendre en compte tout l'espace de travail c'est ce qu'on appelle les méthodes globales ou bien le diviser en plusieurs parties ce qu'on appelle les méthodes locales.

II.5.1.1. Méthodes globales

Avec les méthodes globales, la technique de planification utilisée exploite toutes les informations provenant de l'espace de travail du robot lors du processus de planification. Ainsi, on a une connaissance complète de l'environnement de travail [16]. Cette méthode a l'avantage de ne pas laisser la place à l'imprévu vu qu'on a une connaissance totale sur l'espace de travail. Son inconvénient majeur réside dans le fait que ces méthodes sont très coûteuses en termes de temps de résolution à cause de la grande quantité d'informations utilisées pendant le calcul de trajectoire.

II.5.1.2. Méthodes locales

Le principe des méthodes de planification dites locales est de décomposer l'espace de travail du robot en petites parties. En effet, dans ce genre de méthode, on ne considère pas toutes les informations relatives à l'espace de travail mais on les considère partie par partie. L'avantage avec cette méthode est que le temps de calcul est réduit considérablement. On laisse aussi la place à l'imprévu. Si au cours du déplacement du robot, des changements se produisent dans l'espace de travail, des informations supplémentaires vont être transmises au robot et donc prises en compte lors du calcul des déplacements suivants [17].

II.5.1.3. Méthodes hybrides

Effectuer un bilan des avantages et des inconvénients des différentes méthodes existantes conduit fort logiquement à l'utilisation de méthodes hybrides, combinant des stratégies à la fois globales et locales. [18].

II .5.2. Classification basée sur la méthode de résolution

On peut procéder à une autre classification des méthodes de planification de trajectoires selon la technique de résolution. Par conséquent, et suivant la modélisation faite pour le problème de planification, une technique particulière et appropriée peut être utilisée. On peut les classer en trois catégories : les méthodes basées sur l'optimisation, les méthodes basées sur la commande et les méthodes basées sur une recherche dans un graphe, comme on peut aussi faire la distinction entre les méthodes locales ou globales.

II.5.2.1. Méthodes basées sur l'optimisation

Dans le cadre de ces méthodes, le problème de planification peut être réduit en un problème d'optimisation sous contraintes. Certains travaux proposent une formulation en mono-objectif [19]. Les critères les plus utilisés sont :

- La minimisation de l'erreur de position, nécessaire pour atteindre la configuration finale.
- La minimisation du temps de trajet.
- La minimisation de l'énergie consommée.
- La minimisation des vitesses et des accélérations.

Par rapport à la résolution, et vu la complexité de ce genre de problème, on recourt généralement à l'utilisation de métaheuristiques. Ainsi, on trouve l'utilisation de l'algorithme par essaim particulaire [20], et des algorithmes évolutionnaires.

II.5.2.2 Méthodes basées sur la commande

La deuxième catégorie de méthodes est celle qui se base sur la commande. Tout problème de planification est formulé en un problème de contrôle/commande. Ainsi, pour la résolution, une commande particulière est utilisée.

II.5.2.3. Méthodes basées sur la recherche dans un graphe

La dernière catégorie de méthodes se rattache à la recherche dans un graphe. On a deux étapes:

- 1) La capture de la connectivité de l'espace de travail (la construction d'un graphe).
- 2) La recherche d'une solution dans le graphe construit.

La première étape a pour but de capturer la connectivité de l'espace de travail et dont l'objectif est de construire un graphe qui sera utilisé pour trouver le chemin optimal. Ainsi des techniques ont été développées :

Le diagramme de Voronoi : Le graphe est composé d'un ensemble d'arêtes et d'arcs. Chacun d'entre eux est construit de manière à respecter l'équidistance entre les obstacles. De ce fait, l'évitement d'obstacles est totalement traité. Pour se déplacer d'une configuration à une autre, il suffit de se mettre sur le graphe construit, comme il est indiqué dans le schéma ci-dessous :

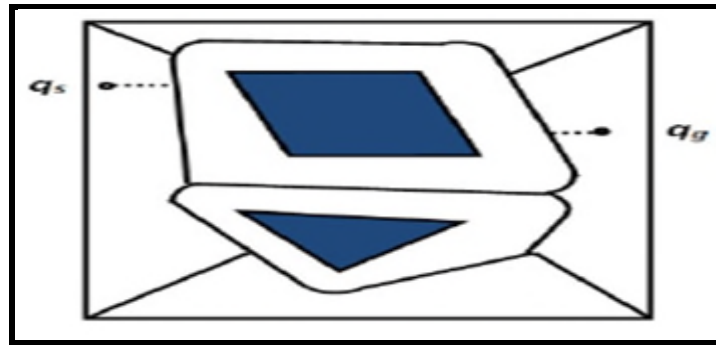


Figure II.6 : Diagramme de Voronoi

Le graphe de visibilité : à partir de chaque sommet de chaque obstacle, on construit toutes les arêtes possibles. Les configurations initiale et finale sont aussi prises en compte. Ainsi, pour se déplacer de la configuration initiale vers la configuration finale, il suffit de trouver le chemin le plus court sur l'ensemble des arêtes construites, comme il est figuré ci-après :

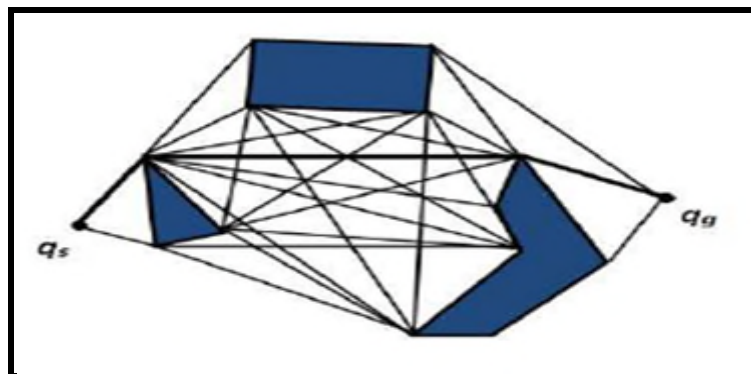


Figure II.7 : Graphe de visibilité

La décomposition cellulaire : à partir de l'espace libre, on procède à des décompositions successives en petites cellules numérotées. Pour la construction du graphe, on ne prend que quelques cellules qui ne coïncident pas avec les obstacles. Ainsi, le graphe résultant ne capture que l'espace libre. Après cela, on procède à une recherche dans le graphe afin de trouver le chemin optimal. Le principe de la décomposition cellulaire est décrit dans la figure suivante :

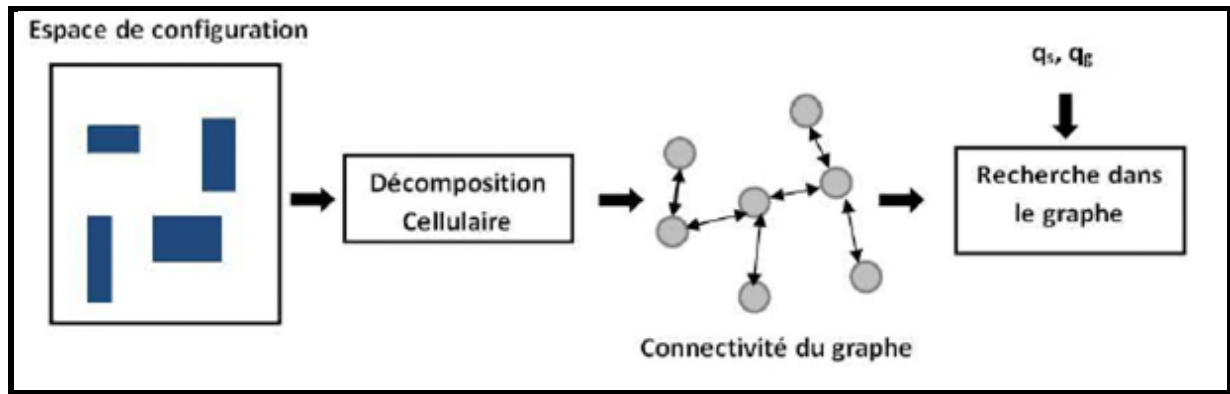


Figure II.8 : Décomposition cellulaire

Les réseaux probabilistes :

Dans le cas des réseaux probabilistes, pour chaque nœud du graphe, on fait un tirage aléatoire. Si la position tirée se trouve dans l'espace libre, on l'ajoute au graphe, sinon on fait un autre tirage. Après cela, on fait une recherche et un lissage de la trajectoire finale, c'est ce que montre le schéma ci-dessous

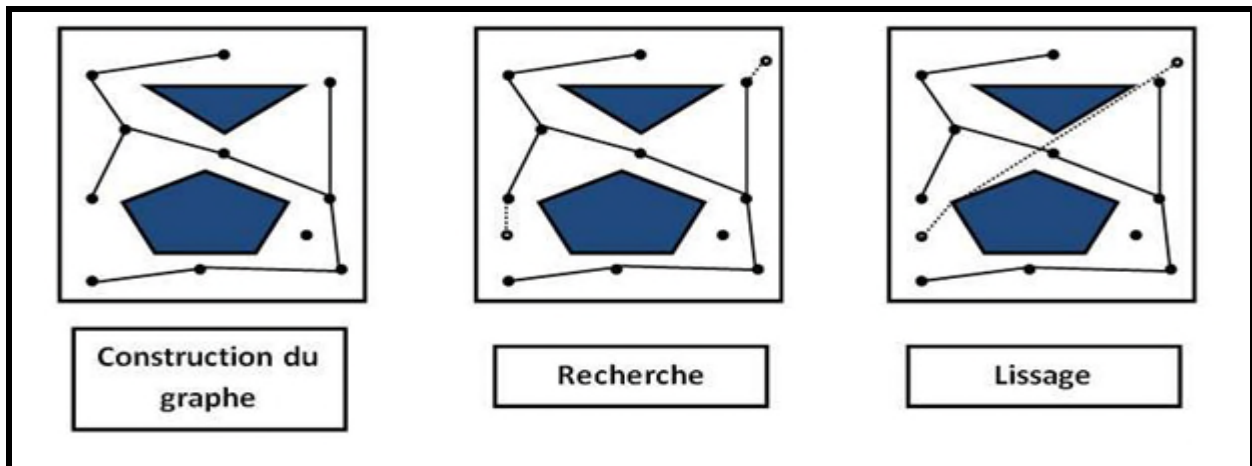


Figure II.9 : Réseaux probabilistes

Le champ potentiel (Potential Field)

Cette approche considère l'environnement du robot comme étant un champ artificiel qui applique des forces différentes sur le robot. Elle traite le robot comme étant un point sous influence d'un champ potentiel artificiel. Le point d'arrivée est une force d'attraction pour le robot, les différents obstacles sont des forces répulsives. Le chemin de navigation est à calculer en faisant la somme des différentes forces.

II.6. Planification de trajectoire sous contraintes

Dans cette partie nous allons voir comment peut-on planifier la trajectoire de robot sous contraintes.

II.6.1. Le lissage de courbes

La nature ou la forme des courbes résultantes est un point important qui doit être pris en considération lors de la résolution d'un problème de planification. Il joue un rôle déterminant dans la solution développée parce qu'il s'attache directement à la structure mécanique du robot. Dans le but de préserver cette structure, on doit assurer des courbes lisses au niveau des positions, des vitesses et des accélérations. Si on considère en plus la dynamique du robot, on doit assurer ainsi une courbe lisse pour le couple généré. Afin de prendre en compte le lissage des courbes dans la résolution du problème de planification, on peut procéder de deux manières différentes :

- Trouver l'ensemble des points de passage du robot et interpoler entre eux pour assurer le lissage.
- Supposer au préalable que l'effecteur et les différentes articulations suivent des trajectoires particulières puis déterminer les coefficients les caractérisant.

II.6.2. L'évitement d'obstacles

L'évitement des obstacles est lié au problème de planification de trajectoire. En effet, étant donné que la majorité des robots opèrent dans des environnements encombrés d'obstacles, il est nécessaire de développer des processus qui permettent de les éviter. En prenant en compte cet aspect lors de la planification, on a deux contraintes supplémentaires :

- ✓ La modélisation de ces obstacles.
- ✓ La manière de les éviter.

Pour la première contrainte, on recourt souvent à l'utilisation de surfaces lisses ou hyper-surfaces (sphères, ...). Ainsi, on entoure l'obstacle initialement avec une forme particulière de l'hyper-surface de rayon minimal. Cette procédure facilite la mesure de distance entre le robot et la nouvelle forme de l'obstacle. Pour la deuxième contrainte, la majorité des techniques développées se base sur une mesure de distance entre le robot et l'obstacle. Cependant, Certaines méthodes comme dans la méthode du champ de potentiel [21] utilisent un autre principe. En effet, le principe de cette méthode est d'effectuer une sorte de distribution de charges dans l'espace du travail, en distinguant un champ attractif attribué à la position finale

à atteindre et un champ répulsif attribué à chaque obstacle, ce qui permettra d'éviter les collisions.

II.6.3. Le test de collision

Le test de collision est une composante essentielle de la planification fondée sur l'échantillonnage. Même si elle est souvent traitée comme une boîte noire, il est important d'étudier son fonctionnement interne pour comprendre les informations qu'il fournit et son coût de calcul associé. Dans la plupart des applications de planification de mouvement, la majorité du temps de calcul est consacré au test de collision [22].

II.6.3.1. Problème du point dans le polygone

On peut classer les polygones en deux catégories : convexes et non convexes. On dit qu'un polygone est convexe si le segment qui relie entre deux points quelconques appartenant à ce polygone reste toujours à l'intérieur de ce dernier. Le teste d'appartenance d'un point à un polygone convexe est facile si on connaît les équations de ses frontières, en remplaçant les coordonnées de ce point dans les inégalités des demi-plans qui forment le polygone [23]. Mais si le polygone est non convexe le concept des demi-plans n'est plus utilisé. Deux méthodes communes se trouvent dans la littérature pour résoudre ce problème : la méthode de pair-impair et l'indice du point.

II.6.3.2. Test d'intersection de deux polygones

On dit que deux polygones P1 et P2 ne sont pas en collision s'il n'y a pas d'intersection entre chaque segment du polygone P1 et les segments du polygone P2, mais ce test prend $(n \times m)$ temps [20] (n, m : nombre des segments de deux polygone P1 et P2 respectivement). Pour cela plusieurs méthodes sont proposées afin d'optimiser le temps du test.

II.7. Les algorithmes d'évitement d'obstacle

Les algorithmes de planification traitent la tâche d'évitement des obstacles d'une manière générale, il existe aussi des algorithmes spécifiques à l'évitement des obstacles. Les algorithmes de planification sont plus généraux, ils ont plusieurs objectifs : 1) établir un plan de navigation, 2) éviter les obstacles et même optimiser. Par contre, les algorithmes d'évitement d'obstacles ont un seul objectif, c'est éviter les obstacles. Ainsi, un algorithme d'évitement d'obstacle peut être intégré dans un cas de planification ou bien un plan d'évitement d'obstacles peut être intégré dans un plan de navigation. Il existe plusieurs algorithmes pour l'évitement d'obstacles. Dont voici les plus importantes :

II.7.1. Algorithme du BUG :

L'objectif est de générer un chemin qui relie un point de départ à un point d'arrivé. Lorsqu'un obstacle i est trouvé, le robot définit un point de contact appelée H_i (Hit point). Le nombre de points de contact est égal aux nombre d'obstacles. Lorsque le robot quitte la zone de l'obstacle il définit un point de départ (leave point) appeler L_i , initialement $i = 0$ ($L_0 = \text{démarrage}$). La procédure utilise trois registres $R1$, $R2$, $R3$, pour enregistrer les informations intermédiaires. Tous les registres sont mis à zéro si un point de contact H est défini $R1$ est utilisé pour enregistrer les coordonnées du point courant Q_m , de la distance minimale entre les limites de l'obstacle et le point cible. C'est-à-dire une comparaison est effectuée à chaque point du chemin, $R2$ intègre la borne de l'obstacle en commençons par H_i , et $R3$ pour enregistrer la longueur de la borne de l'obstacle à calculer à partir de Q_m .

La procédure définit les étapes suivantes :

Étape 1: du point L_{i-1} aller vers la cible au long d'une droite jusqu'à ce que l'un de ces évènements suivant intervienne :

- Le robot a atteint son objectif.
- Un obstacle est trouvé et un point de contact défini.
- Aller à l'étape 2.

Étape 2: utilisant une direction locale (gauche ou droite) naviguer aux bornes de l'obstacle (faire un tour sur l'obstacle commençons par la gauche ou la droite)

- Si le robot a atteint son objectif alors arrêter.
- Sinon faire un tour complet sur l'obstacle, revenir au point H_i et définir un point $L_i = Q_m$ pour quitter la zone de l'obstacle.
- Aller à l'étape 3.

Étape 3 : appliquer un test sur la possibilité d'atteindre le but, c'est se n'est pas possible arrêter, sinon :

- Utiliser le contenu des registres $R2$, $R3$ pour déterminer le chemin le plus court entre tout point et les bornes de l'obstacle et avec l'objectif
- Définir L_i et mise à jour i ($i=i+1$)
- Aller a l'étape 1.

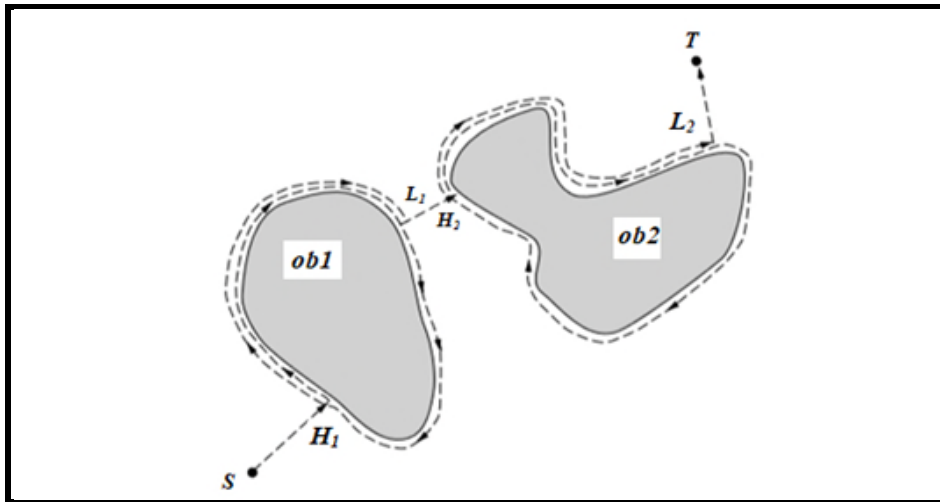


Figure II.10: Algorithme BUG 1

(H_1 H_2 point de contact L_1 L_2 point de départ S , T point de départ est d'arrivé)

Avantages de cette méthode :

Cette méthode présente quelques avantages :

- Elle ne fait pas explorer le même obstacle deux fois c'est-à-dire que le robot va éviter un nombre finis d'obstacles.
- L'algorithme est indépendant de la géométrie de l'environnement.
- Une méthode qui se caractérise par Simplicité.

Inconvénients de cette méthode :

D'une autre part on trouve des limitations importantes qui peuvent rendre cet algorithme inutilisable :

- Chemin très long: des tours sont effectués sur les différents obstacles trouvés
- la complexité de l'algorithme due aux multitudes des calculs de distance aux bornes d'obstacles et aux différentes comparaisons effectuées

II.7.2. Algorithme du BUG 2

On peut considérer cet algorithme comme une amélioration de l'algorithme précédent. L'algorithme peut explorer le même obstacle i plus qu'une seule fois parce que l'algorithme n'a pas de méthode pour distinguer les différents obstacles. Pour cette raison on va utiliser l'indice i pour se référer à un obstacle, de plus on va utiliser j pour indiquer la $j^{\text{ème}}$ occurrence

d'un point de contact ou de départ sur le même ou sur un obstacle différent. Initialement $j = 1$, $L^0 = \text{départ}$, l'algorithme exécute les étapes suivantes :

Étape 1 : du point L^{j-1} aller tout au long du segment de la droite (départ, arrivé) jusqu'à l'un des évènements suivants :

- Le robot a atteint l'objectif.
- Un obstacle est trouvé et un point de contact H^j est défini, aller à l'étape 2

Étape 2 : utilisant une direction locale acceptable (gauche, droite), suivre la borne de l'obstacle jusqu'à ce que l'un des évènements suivant se présente :

- Le robot a atteint son objectif, la procédure s'arrête.
- Le robot mobile et le segment droite (départ, arrivée) se mettent à un point Q , la distance $d(Q) < d(H^j)$ et la ligne (Q , arrivée) ne s'y mettent pas, tandis que l'obstacle courant aux point Q , définir un point de départ $L^j = Q$ mise à jour $j = j+1$, retour à l'étape 1
- Le robot revient aux point de contact H^j sans avoir défini le prochain point de contact H^{j+1} le robot ne peut pas atteindre son objectif, la procédure s'arrête.

C'est ce qu'illustre le schéma de figure ci-après :

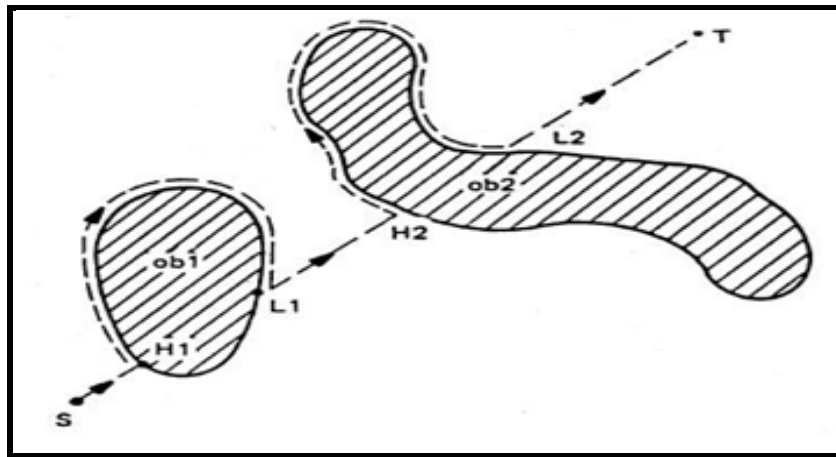


Figure II.11: Algorithme BUG 2

(H_1 H_2 point de contact L_1 L_2 point de départ S , T point de départ est d'arrivée)

Avantage de cette méthode : Cette méthode à quelques avantages par rapport au BUG 1 :

- Le chemin est court par rapport au BUG 1
- Les calculs de comparaison et de distance sont éliminés ce qui réduit la complexité de l'algorithme.

Limites à identifier :

Malgré les améliorations portées sur l'algorithme il reste infaisable, on cite les limites suivantes :

- Il peut explorer le même obstacle plusieurs fois (figure. II.12)
- La stratégie de choix de direction n'est pas claire et peut causer des scénarios où le robot explore tout l'obstacle (figure. II.13)

Les deux algorithmes présentés ci-dessous sont assez simples, leur inconvénient majeur est qu'ils ne trouvent jamais un chemin optimal. C'est ce qui est présenté ci-après :

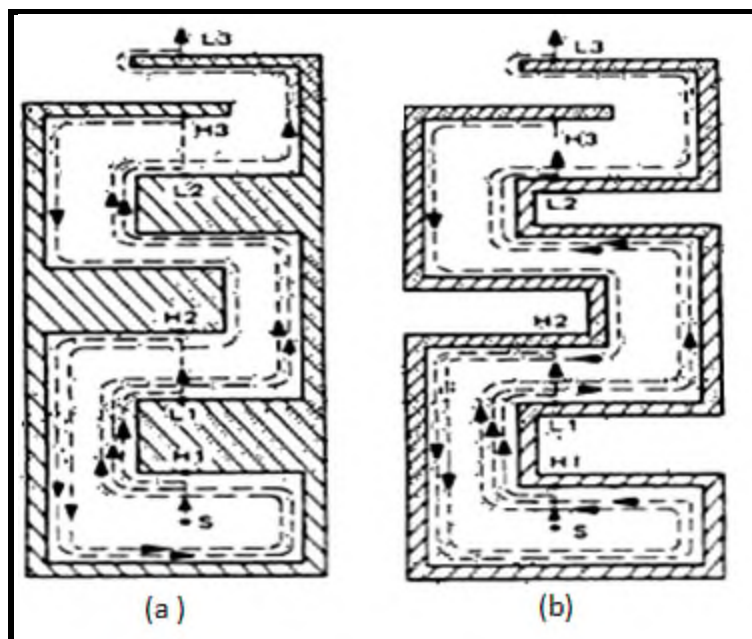


Figure II.12: Exploration de la même région plusieurs fois par un robot qui utilise BUG 2

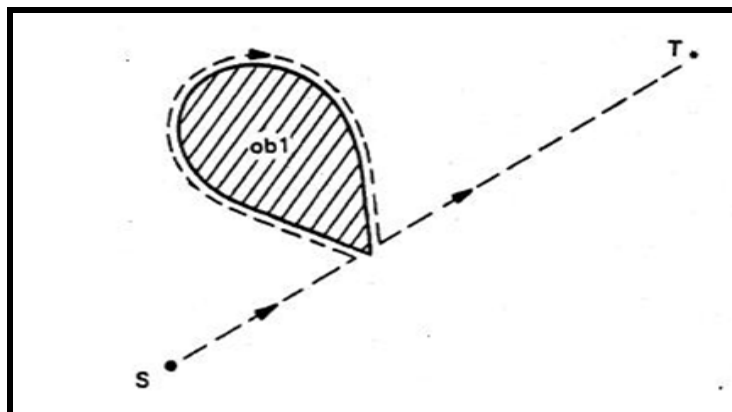


Figure II.13 : Un cas où un robot fait un tour complet sur l'obstacle sous BUG 2

II .8. Conclusion

Dans cette partie, on a abordé en général les problèmes qui concernent la planification de trajectoires pour les robots manipulateurs. On y a présenté les caractéristiques des robots et on y a illustré leurs principales propriétés, à savoir la redondance et l'étude des singularités, qui sont des éléments importants à prendre en considération lors du processus de planification. Par la suite, on a étudié le problème de planification de trajectoires.

Dans le cas des robots manipulateurs, la planification peut s'effectuer dans deux espaces différents, l'espace articulaire et l'espace cartésien. On a également, fait la classification des méthodes de planification : la première concerne la prise en compte des informations s'attachant à l'environnement de travail du robot et la deuxième porte sur la technique utilisée dans la résolution du problème.

Enfin, nous avons étudié les contraintes principales qui sont prises en compte dans les problèmes de planification de trajectoires : la première contrainte concerne le lissage de courbes résultantes ce qui a un impact direct sur la structure mécanique du robot et la deuxième concerne l'environnement de travail du robot. On a montré qu'il est impératif d'intégrer une technique d'évitement d'obstacles dans le processus de planification.

Chapitre III :

*Planification de chemins de robots en utilisant
le champ de potentiel artificiel numérique*

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

III.1. Introduction

Dans ce chapitre nous allons faire l'étude et la simulation de la planification de chemin en utilisant le champ potentiel artificiel. L'approche est basée sur la construction d'une grille de discrétisation à travers l'espace de travail (Work Space) du robot. Les cellules de cette grille sont marquées comme étant en collision ou hors collision selon leurs intersections avec les obstacles. Puis la grille est utilisée pour calculer un champ potentiel artificiel à travers les points de discrétisation.

La valeur du champ potentiel en chaque point caractérise sa distance aux obstacles et au p_{goal} (le point de destination). Le champ potentiel artificiel de l'espace de travail (WSAPF) construit sera utilisé pour construire un champ potentiel artificiel de l'espace de configuration (CSAPF). Ce dernier sera utilisé pour résoudre les requêtes de planification de chemins en poursuivant la direction de son gradient de descente. Nous commençons par la présentation de Bitmap de l'espace de travail et de l'espace de configurations.

III.2. Bitmap de l'espace de travail et Bitmap de l'espace de configuration

Dans cette section nous allons parler du Bitmap de l'espace de travail et du Bitmap de l'espace de configuration.

III.2.1. Bitmap de l'espace de travail

Nous supposons que l'espace de travail dénoté W est un sous-ensemble de \mathbb{R}^n , avec $n = 2$ ou 3 (le plan ou l'espace tridimensionnel). L'espace de travail W est modélisé par une pyramide de matrices de dimension n . A chaque niveau de résolution, la matrice est représentée par une fonction :

$$\begin{aligned} BM : W &\rightarrow \{0,1\} \\ x &\mapsto BM(x) \end{aligned} \tag{III.1}$$

Tel que le sous-ensemble de points pour lesquels $BM(x) = 1$ représente le sous-espace cartésien des obstacles (Work Space obstacles) et le sous-espace des points x tels que $BM(x) = 0$ représente la partie libre de l'espace de travail dénotée W_{free} .

La distance entre les centres de deux cellules consécutives de la même ligne ou de la même colonne d'une matrice est constante. Ainsi à chaque niveau de résolution, les points centraux des cellules forment une grille régulière dénotée par GW . Le sous-ensemble de la grille où la valeur de BM est nulle, est dénoté par GW_{empty} . Pour un entier $k \in [1, n]$, le voisinage k d'un point x dans une grille de dimension n est défini comme étant l'ensemble de points dans la

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

grille ayant au plus k coordonnées différentes de ceux de x . Dans GW nous utilisons toujours le voisinage 1 sauf si c'est spécifié autrement.

Pour le cas d'espace de travail de dimension 2, le niveau de résolution le plus grossier que nous considérons dans ce travail est (16×16) et le niveau le plus fin est (512×512) . La représentation de l'espace de travail est donnée au planificateur dans la grille la plus fine et les représentations dans les autres niveaux de résolution sont dérivées automatiquement de façon conservative. Le facteur entre deux résolutions successives est de 2.

III.2.2. Bitmap de l'espace de configuration

Du moment où nous avons discrétisé l'espace de travail de façon hiérarchique, il est convenable de discrétiser aussi l'espace de configurations en construisant une pyramide de grilles de multi-résolution. Cette pyramide a autant de niveaux de résolution que la pyramide de discrétisation de l'espace de travail. On peut démontrer que pour le même niveau d'erreur, les résolutions des deux pyramides sont étroitement liées. Dans chaque niveau de discrétisation, nous dénotons par GC la grille représentant l'espace de configurations et par GC_{free} le sous ensemble de la grille situé hors collision.

Si on dénote par δ la distance entre deux points adjacents dans une grille de l'espace de travail GW . Alors dans la pyramide de l'espace de travail, δ varie entre une valeur maximale δ_{max} et une valeur minimale δ_{min} . A titre d'exemple supposons que nous avons une pyramide de grilles dans l'espace de travail dont les dimensions varient entre (16×16) et (512×512) , alors si le diamètre de l'espace de travail est 1, on aura $\delta_{min} = \frac{1}{512}$ et $\delta_{max} = \frac{1}{16}$.

III.3. Calcul du champ de potentiel artificiel

Le calcul du champ potentiel artificiel dans l'espace de configuration peut se faire généralement en deux étapes principales :

Étape 1 : des champs potentiels artificiels sont calculés dans l'espace de travail pour un ensemble de points du robot appelés points de contrôle. Ces champs de potentiels sont appelés champs de potentiels artificiels de l'espace de travail WSAPF et ils attirent les points de contrôle vers leurs positions correspondant à la configuration de destination.

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

Étape 2 : la combinaison des champs potentiels de l'espace de travail dans une autre fonction appelée champ de potentiel de l'espace de configuration qui est défini dans l'espace de configuration du robot.

De façon plus formelle, soit p_i ($i = 1, \dots, s$) dénotant les points de contrôle du robot. Chaque champ de potentiel de l'espace de travail est une fonction :

$$V_{p_i}: x \in W_{empty} \mapsto V_{p_i}(x) \in \quad (III.2)$$

Le champ de potentiel de l'espace de configurations est défini par :

$$U(q) = G \left(V_{p_1}(X(p_1, q)), \dots, V_{p_s}(X(p_s, q)) \right) \quad (III.3)$$

Dans lequel G est appelée la fonction d'arbitrage.

La logique derrière le passage par ces deux étapes est d'utiliser l'espace de travail dont la dimension est petite comme source d'information nécessitant peu de calcul, pour construire un champ potentiel adéquat à l'espace de configuration dont la dimension est généralement grande. En effet, les champs potentiels artificiels de l'espace de travail WSAPF que nous allons calculer dans ce chapitre n'ont pas de minimums locaux. Ceci permet de construire un champ potentiel de l'espace de configuration qui empêche le robot d'être piégé dans des cavités simples formées par les obstacles. Cependant, le champ de potentiel de l'espace de configuration peut quand même avoir des minimums locaux. Une raison de ceci est que l'information contenue dans les WSAPFs ne caractérise pas complètement la connectivité de l'espace hors collision. En effet les WSAPFs ne sont calculés que pour un nombre fini de points du robot (points de contrôle). L'autre raison est que la combinaison de plusieurs fonctions qui n'ont pas des minimums locaux n'est pas garantie de résulter en une fonction qui à son tour n'a pas de minimums locaux. En réalité le champ de potentiels de l'espace de configuration CSAPF attire de façon concurrente les différents points de contrôle vers leurs positions de destination. Mais ces points de contrôle sont liés entre eux par les contraintes cinématiques du robot, ceci fait que les points de contrôle sont en réalité en compétition en eux pour atteindre leurs positions de destinations et cette compétition peut engendrer des minimums locaux. Le rôle de la fonction G est d'arbitrer cette compétition.

III.3.1. Calcul du champ de potentiel de l'espace de travail WSAPF

Pour calculer les champs potentiels artificiels de l'espace de travail WSAPF, nous utilisons deux approches :

- ✓ WSAPF simple.

- ✓ WSAPF amélioré.

III.3.1.1. WSAPF simple:

Soit p un point de contrôle du robot et $V_p(x)$ la fonction de champ potentiel de l'espace de travail correspondant à p . Nous voulons que $V_p(x)$ n'ait aucun minimum local c'est-à-dire, qu'elle doit avoir un seul minimum à la position de destination dans le sous-ensemble connecté de W_{free} contenant ce point de destination. En effet, ceci est une étape heuristique majeure vers la construction d'un champ potentiel de l'espace de configuration qui a un petit nombre de minimums locaux.

Soit x_{goal} la position de destination du point de contrôle p dans l'espace de travail W . La fonction V_p dans WSAPF est calculée comme suit :

- ✓ Au départ, la valeur de $V_p(x_{goal}) \rightarrow 0$.
- ✓ Après, la valeur de V_p est fixée à 1 à tous les voisins du point x_{goal} .
- ✓ Ensuite, la valeur de V_p est fixée à 2 aux voisins de ces voisins et ainsi de suite jusqu'à ce que tous les points de GW_{empty} soient explorés. L'algorithme suivant décrit le calcul de V_p .

```

1. Pour tout  $x \in GW_{empty}$ , fixer  $V_p(x)$  à  $\infty$ 
2. Fixer  $V_p(x_{goal})$  à 0 et  $L_0$  à  $(x_{goal})$ .
3. pour  $i=0,1,\dots$  jusqu'à  $L_i$  soit vide, faire :
   a. Initialiser  $L_{i+1}$  à l'ensemble vide.
   b.  $\forall x \in L_i$ , pour tout voisin  $y$  de  $x$  dans  $GW_{empty}$  :
       si  $V_p(y) = \infty$  puis  $V_p(y) = i + 1$ ; et insérer  $y$  à la fin  $L_{i+1}$ 
    
```

Figure III.1: Algorithme de calcul de WSAPF.

Il faut noter que l'algorithme précédent calcul V_p seulement dans le sous-ensemble connecté de GW_{empty} contenant x_{goal} . Ainsi, après que l'algorithme soit exécuté et que p est à la position initiale (x_{init}) avec $V_p(x_{init}) = \infty$, on peut directement retourner qu'il n'y a aucun chemin entre x_{init} et x_{goal} . La deuxième étape de l'algorithme peut être modifiée facilement pour accommoder le cas où la destination de p est un sous ensemble de GW_{empty} . La figure (III.2) montre les iso-contours du WSAPF que nous avons obtenus pour la scène donnée et la figure (III.3) donne sa table de valeurs pour une grille de (33x33). Dans cette figure, les obstacles 7 et 12 ont été modifiés pour garder les passages étroits.

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

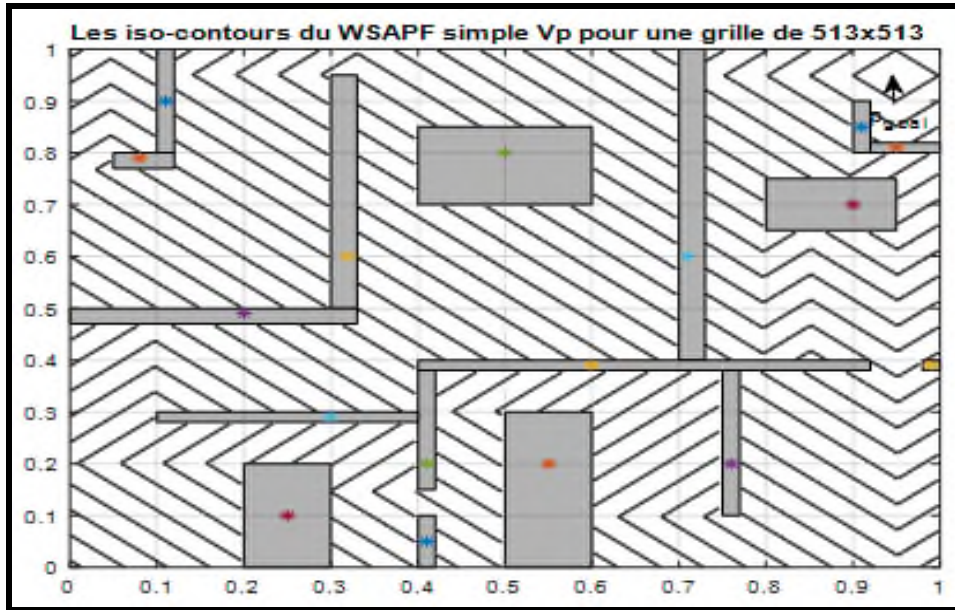


Figure III.2 : Les iso-contours du WSAPF pour une grille de (513x513)

88	87	86	85	84	83	82	81	82	83	84	85	86	87	88	NaN	NaN	137	136	135	134	133	132	131	132	133	134	135	136	137	138	139	NaN											
87	86	85	84	83	82	81	80	81	82	83	84	85	86	87	NaN	NaN	136	135	134	133	132	131	130	NaN	NaN	135	136	137	138	139	140	NaN											
86	85	84	83	82	81	80	79	80	81	82	83	84	85	86	NaN	NaN	135	134	133	132	131	130	129	NaN	NaN	136	137	138	139	140	141	NaN											
85	84	83	82	81	80	79	78	NaN	NaN	84	85	86	87	NaN	NaN	134	133	132	131	130	129	128	NaN	NaN	NaN	136	137	138	139	140	141	NaN											
84	83	82	81	80	79	78	77	NaN	NaN	84	85	86	87	88	NaN	NaN	133	132	131	130	129	128	127	126	125	124	123	122	121	120	119	NaN											
83	82	81	80	79	78	77	76	NaN	NaN	85	86	87	88	89	NaN	NaN	132	131	130	129	128	127	126	125	124	123	122	121	120	119	118	NaN											
NaN	NaN	NaN	NaN	NaN	NaN	NaN	75	NaN	NaN	86	87	88	89	90	NaN	NaN	131	130	129	128	127	126	125	124	123	122	121	120	119	118	117	NaN											
NaN	NaN	NaN	NaN	NaN	NaN	NaN	74	NaN	NaN	87	88	89	90	91	NaN	NaN	130	129	128	127	126	125	124	123	122	121	120	119	118	117	116	NaN											
NaN	NaN	NaN	NaN	NaN	NaN	NaN	73	NaN	NaN	88	89	90	91	92	NaN	NaN	129	128	127	126	125	124	123	122	121	120	119	118	117	116	115	NaN											
NaN	NaN	NaN	NaN	NaN	NaN	NaN	72	NaN	NaN	89	90	91	92	93	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN										
74	73	72	71	70	69	70	71	NaN	NaN	90	91	92	93	94	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN										
73	72	71	70	69	68	69	70	NaN	NaN	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	NaN											
NaN	NaN	NaN	NaN	NaN	NaN	NaN	67	NaN	NaN	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	NaN											
NaN	NaN	NaN	NaN	NaN	NaN	NaN	66	NaN	NaN	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	NaN											
70	69	68	67	66	65	64	63	62	61	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	NaN										
NaN	NaN	NaN	NaN	NaN	NaN	NaN	65	NaN	NaN	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	NaN										
NaN	NaN	NaN	NaN	NaN	NaN	NaN	64	NaN	NaN	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	NaN									
NaN	NaN	NaN	NaN	NaN	NaN	NaN	63	NaN	NaN	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	NaN								
NaN	NaN	NaN	NaN	NaN	NaN	NaN	62	NaN	NaN	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	NaN							
NaN	NaN	NaN	NaN	NaN	NaN	NaN	61	NaN	NaN	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	NaN						
NaN	NaN	NaN	NaN	NaN	NaN	NaN	60	NaN	NaN	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	NaN					
48	47	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	NaN				
47	46	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	NaN			
46	45	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	NaN		
45	44	43	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN			
44	43	42	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	NaN
42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	NaN	
41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	NaN		
40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	NaN			
39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	NaN				
38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	1	NaN			
39	38	37	36	35	34	33	32	31	30	29	28	NaN	24	23	22	21	20	19	18	17	16	15	14	13	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure III.3: Champs de potentiel dans l'espace de travail WSAPF simple dans la grille 33x33

III.3.1.2. WSAPF amélioré

Un inconvénient majeur du WSAPF simple est l'induction des chemins qui s'approchent des obstacles dans l'espace de travail. Dans le but de réduire le risque de création des minimums locaux dans le CSAPF et en même temps augmenter l'espace de manœuvres du robot pour pouvoir échapper aux minimums locaux rencontrés, on utilise le champ potentiel artificiel amélioré de l'espace de travail. Ce potentiel est comme le précédent (WSAPF simple) mais en

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

plus, ce champ est doté de l'inverse de son gradient qui sert à attirer le point de contrôle vers sa position de destination qui demeure éloigné le plus possible des obstacles. Ce champ potentiel amélioré est calculé en trois étapes :

Étape 1 : le calcul de la distance L_1 discrète dénotée $d_1(x)$ de chaque point $x \in GW_{empty}$ aux obstacles et un sous-ensemble $S \subset GW_{empty}$ est concurremment extrait. Ce sous-ensemble est appelé *Squelette de l'espace de travail*. Le calcul de la distance L_1 des obstacles est effectué selon l'algorithme d'expansion de front d'onde de la figure (III.4). Au début, les points situés aux frontières des obstacles sont identifiés et la valeur de d_1 en ces points est fixée à 0. Ensuite, la valeur de d_1 est fixée à 1 à tous les points voisins de ces points de frontière se trouvant dans GW_{empty} , et à 2 aux voisins de ces nouveaux points... etc. L'algorithme termine quand tous les points sont visités. Pour chaque point $x \in GW_{empty}$, soit $O(x)$ un tel point. Il faut noter que $d_1(x)$ est la distance L_1 entre x et $O(x)$.

Étape 2 : On fixe $O(x) = x \forall x \in L_0$.

Étape 3 : Soit x le point en cours de traitement dans L_i et y l'un de ses voisins. Si $d_1(x) = \infty$, on fixe $O(y) = O(x)$, sinon, si la distance L_1 entre $O(y)$ et $O(x)$ est supérieure à un certain seuil (typiquement 2) et $x \notin S$, on inclue y dans S . La figure (III.5) représente le squelette de l'espace de travail que nous avons obtenu.

```
1. Pour tout  $x \in GW_{empty}$ , fixer  $d_1(x) = \infty$ .
2. a. Scanner  $GW_{empty}$  et identifier tout point  $x$  tel que  $BM(x)=1$  et l'un de
   ses voisins est dans  $GW_{empty}$ .
   b. Mettre la liste de ces points dans  $L_0$ .
   c. Inclure les points de frontière de l'espace de travail dans  $L_0$ .
   d. Pour tout point de  $L_0$  fixer  $d_1(x) = 0$ .
3. For  $i=0,1,2,\dots$  until  $L_i$  soit vide
   For every neighbor  $y$  de  $x$  dans  $GW_{empty}$ 
if  $d_1(y) = \infty$  then  $d_1(y) = i + 1$ ; et insérer  $y$  dans  $L_{i+1}$ 
```

Figure III.4: Algorithme de calcul de la distance L_1 des obstacles.

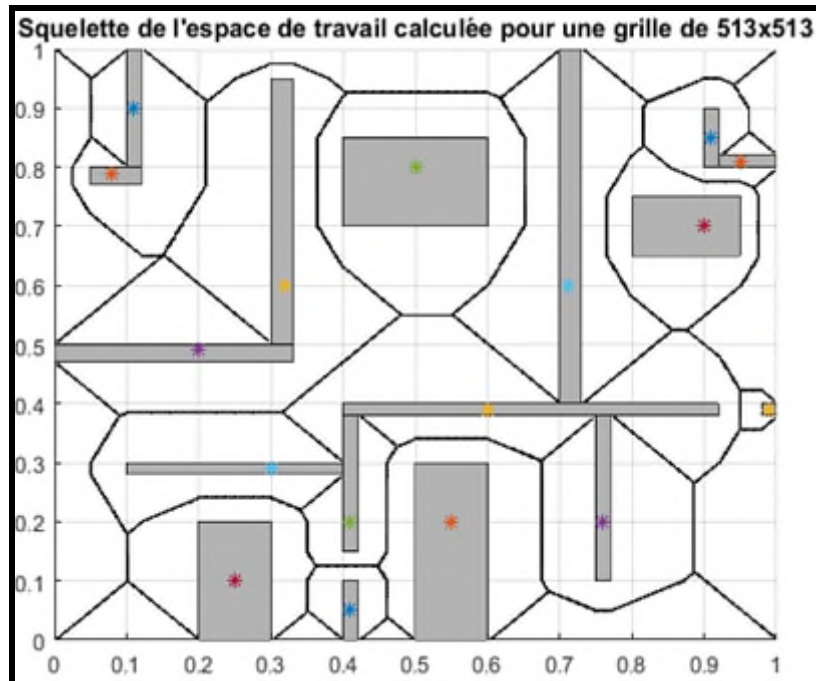


Figure III.5 : Squelette de l'espace de travail obtenu pour une grille de 513x513

Maintenant, le WSAPF amélioré est calculé tout d'abord sur S (squelette de l'espace de travail). Ceci est effectué comme suit :

Au premier lieu, la position de destination x_{goal} du point de contrôle p est connectée au squelette S par un chemin suivant le gradient de d_1 et les positions le long de ce chemin sont inclus dans S pour produire un nouveau squelette appelé squelette augmenté.

Ensuite le WSAPF amélioré est calculé dans le sous-ensemble du squelette augmenté accessible à partir de x_{goal} , en utilisant l'algorithme d'expansion de front d'onde partant de x_{goal} et guidé par la fonction d_1 comme suit : Le potentiel 0 est donné à x_{goal} et elle est insérée dans une queue Q dont les éléments sont les point de S triés par ordre décroissant de la valeur de d_1 . Le premier élément de Q (appelé x) est enlevé de Q et chaque n -voisins y de x dont le potentiel n'a pas encore été calculé reçoit une valeur de potentiel égale à $(V_p(y) = V_p(x) + 1)$ et il est inséré dans Q . L'algorithme termine quand Q devient vide, c'est à dire quand tous les points du squelette augmenté accessibles à partir de x_{goal} ont reçu une valeur du potentiel. La figure (III.6) représente l'algorithme de calcul de WSAPF amélioré.

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

```

1. Initialisation : Pour tout  $x \in GW_{empty}$ , fixer  $V_p(x) = \infty$ .
2. Connection de  $x_{goal}$  au squelette de l'espace de travail
a.  $x = x_{goal}$ ;
   While  $x \notin S$ 
Inclure  $x$  dans  $S$ ;
Choisir un voisin  $y$  de  $x$  ayant la plus grande valeur de  $d_1$ ;
    $x=y$ ;
endwhile;
3. Calcul du WSAPF amélioré dans le squelette augmenté de l'espace de travail
   a. fixer :  $V_p(x_{goal}) = 0$ ;  $Q \leftarrow x_{goal}$ ;  $L_0 = \text{ensemble vide}$ ;
   b. répéter
       Enlever le premier élément  $x$  de  $Q$  et l'insérer à la fin de  $L_0$ ;
       Pour tout voisins- $n$   $y$  de  $x$  dans  $GW_{empty}$ 
Si  $V_p(y) == \infty$  alors  $V_p(y) = V_p(x) + 1$ ; insérer  $y$  dans  $Q$ ;
Fin pour;
   Jusqu'à ce que  $Q$  est vide (A la fin de cette étape  $L_0$  contiendra
tous les points de  $S$  accessibles à partir de  $x_{goal}$ )
4. Calcul de WSAPF amélioré dans le reste de points de  $GW_{empty}$ 
    $i=0$ ;
   répéter
       initialiser  $L_{i+1}$  à l'ensemble vide;
       Pour tout point  $x \in L_i$ 
           Pour tout voisins  $y$  de  $x$  in  $GW_{empty}$ 
               If  $V_p(y) == \infty$  fixer  $V_p(y) = V_p(x) + 1$ ; insérer  $y$  à la fin de
 $L_{i+1}$ 
       Fin pour;
       Fin pour;
        $i=i+1$ ;
   Jusqu'à ce que  $L_i$  est vide

```

Figure III.6 : Algorithme de calcul du WSAPF amélioré.

La figure (III.7) représente les iso-contours du WSAPF amélioré calculé pour l'environnement représenté. La poursuite du gradient de descente de ce potentiel partant de n'importe quel point accessible mène vers x_{goal} en poursuivant un chemin qui s'éloigne le plus possible des obstacles.

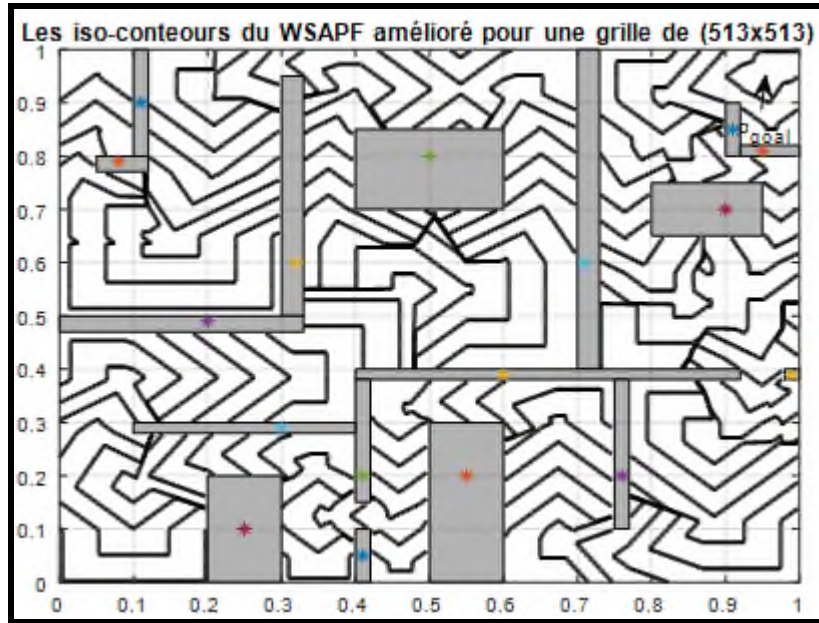


Figure III.7 : Les iso-contours du WSAPF amélioré pour une grille de (513x513)

III.3.2. Calcul du champ de potentiel de l'espace de configuration CSAPF

Le champ de potentiel de l'espace de configuration est calculé comme une combinaison :

$$U(q) = G \left(V_{p_1}(X(p_1, q)), V_{p_2}(X(p_2, q)), \dots, V_{p_s}(X(p_s, q)) \right) \quad (III.4)$$

des potentiels de l'espace de travail V_{p_i} ($i = 1, 2, \dots, s$) définis pour s ($s \geq 1$) points de contrôle distincts p_i , q est la configuration du robot. Cette combinaison attire de façon concurrente les différents points de contrôle vers leurs destinations. $U(q)$ atteint sa valeur minimale (c'est à dire 0) quand tous les point de contrôle sont à leurs positions de destination. A toute autre configuration $U(q)$ est *strictement* positive.

Les points de contrôle sont ceux utilisés pour décrire la configuration de destination. Par définition le robot est à sa configuration de destination si tous les points de contrôle sont à leurs points de destination respectifs.

La fonction d'arbitrage G est sous forme d'une somme simple des WSAPF :

$$G(V_{p_1}, V_{p_2}, \dots, V_{p_s}) = \sum_{i=1}^{i=s} V_{p_i} \quad (III.5)$$

Ce choix simple semble naturel car il ne favorise aucun point de contrôle par rapport aux autres. Cependant et précisément pour cette raison les conflits entre les points de contrôle tendent à être fréquents produisant de nombreux minimums locaux.

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

Le choix de G qui apparemment minimise le nombre de minimums locaux est :

$$G(V_{p_1}, V_{p_2}, \dots, V_{p_s}) = \min_{i=1, \dots, s} (V_{p_i}) \quad (\text{III.6})$$

En effet cette fonction d'arbitrage favorise l'attraction du point qui est déjà dans la bonne position pour atteindre sa destination. Cependant, quand l'un des points de contrôle atteint sa position de destination, le champ CSAPF devient identiquement nul, et les autres points ne sont pas attirés vers leurs destinations. Une façon d'éviter ce problème est de rajouter un autre terme à la fonction d'arbitrage pour obtenir :

$$G(V_{p_1}, V_{p_2}, \dots, V_{p_s}) = \min_{i=1, \dots, s} (V_{p_i}) + \varepsilon * \max_{i=1, \dots, s} (V_{p_i}) \quad (\text{III.7})$$

Où ε est un petit nombre (exemple 0.1) qui dépend du robot. Un autre choix de G est :

$$G(V_{p_1}, V_{p_2}, \dots, V_{p_s}) = \max_{i=1, \dots, s} (V_{p_i}) \quad (\text{III.8})$$

Ce choix favorise le point de contrôle qui est le plus éloigné de sa destination.

III.4. Planification en utilisant le champ de potentiel artificiel

La technique que nous avons considérée ici prend chaque niveau de résolution dans les pyramides de l'espace de travail et les pyramides de l'espace de configurations en commençant par le niveau le plus grossier vers le niveau le plus fin. Elle obtient un succès dès qu'un chemin a été généré. Elle constitue un échec si après avoir considéré les bitmaps les plus fins, elle n'a pas encore généré un chemin.

L'algorithme que nous avons implémenté performe essentiellement une recherche basée sur le principe "meilleur en premier" (best-first). Il construit de façon itérative un arbre T dont les nœuds sont des configurations dans GC_{free} . La racine de T est la configuration initiale q_{init} . A chaque itération l'algorithme examine les voisins- k ($k \in [0, n]$) de la feuille de T qui est la plus petite valeur du CSAPF, il retient les voisins qui ne sont pas déjà dans T et les installe comme successeurs de la feuille en cours de considération. L'algorithme termine quand la configuration de destination q_{goal} est atteinte (succès) ou bien quand le sous-ensemble de GC_{free} accessible de q_{init} est complètement exploré (échec). Dans nos applications nous avons fixé $k = n$.

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

Tant que l'algorithme n'a pas atteint un minimum local de CSAPF, la recherche se réduit à la poursuite d'une approximation de gradient de descente du CSAPF (la procédure de la plus grande descente). Quand un minimum local est atteint, l'algorithme remplit de façon naturelle le puit du minimum local jusqu'à ce qu'un point d'inflexion soit atteint. Après, il poursuit de nouveau le gradient de descente de CSAPF. Il s'arrête quand la valeur du potentiel artificiel de l'espace de configuration devient nulle $U = 0$ c'est à dire que q_{goal} est atteint.

Il y a un aspect important de cet algorithme qui nécessite plus de précision. Vu que le CSAPF dépend de la distance de quelques points de contrôle des obstacles, l'algorithme de recherche "Best-First" ne garantit pas que la collision entre le robot et les obstacles sera complètement évitée. Par conséquent, à chaque fois que le planificateur considère une nouvelle configuration dans GC , il doit vérifier qu'elle appartient à GC_{free} (hors obstacles). D'un autre côté et vu que le planificateur ne représente pas explicitement les obstacles dans l'espace des configurations, la vérification contre les collisions doit être effectuée dans l'espace de travail. Généralement, la vérification contre les collisions est très coûteuse en termes de temps de calcul. Parmi les techniques les plus utilisées sont celles basées sur le principe de *diviser et conquérir* qui consiste en l'ajustement itératif du nombre de points sur les frontières du robot qui doivent être vérifiés contre la collision. Quand le robot est loin des obstacles, seulement un petit nombre de points doit être vérifiés contre la collision et quand le robot s'approche des obstacles un nombre croissant de points (selon la résolution du bitmap de l'espace de travail) dans le segment du robot qui est proche des obstacles doit être vérifié contre les collisions.

III.5. Résultats de la simulation

Dans cette section, nous exposons les applications de simulation que nous avons effectuées et accompagnées de nos interprétations et des figures explicatives. Pour commencer, la scène que nous avons considérée est représentée par la figure (III.9). Cette scène est de dimension 2 et elle est comprise dans un environnement de surface (1mx1m). Elle contient 17 obstacles qui sont numérotés comme sur la figure. Ces obstacles sont représentés par des matrices contenant les coordonnées de leurs vertex et les coordonnées d'un point situé à l'intérieur de l'obstacle. Ce dernier point est très utile pour le test contre la collision. La figure (III.8) donne les spécifications nécessaires des obstacles.

```

ob1={ [0.1 0.12 0.12 0.1; 1 1 0.8 0.8] [0.11;0.9]};
ob2={ [0.05 0.12 0.12 0.05; 0.8 0.8 0.77 0.77] [0.08;0.79]};
ob3={ [0.3 0.33 0.33 0.3; 0.95 0.95 0.5 0.5] [0.32;0.6]};
ob4={ [0 0.33 0.33 0; 0.5 0.5 0.47 0.47] [0.2;0.49]};
ob5={ [0.4 0.6 0.6 0.4; 0.85 0.85 0.7 0.7] [0.5;0.8]};
ob6={ [0.7 0.73 0.73 0.7; 1 1 0.4 0.4] [0.71;0.6]};
ob7={ [0.8 0.95 0.95 0.8; 0.75 0.75 0.65 0.65] [0.9;0.7]};
ob8={ [0.9 0.92 0.92 0.9; 0.9 0.9 0.8 0.8] [0.91;0.85]};
ob9={ [0.92 1 1 0.92; 0.82 0.82 0.8 0.8] [0.95;0.81]};
ob10={ [0.4 0.92 0.92 0.4; 0.4 0.4 0.38 0.38] [0.6;0.39]};
ob11={ [0.75 0.77 0.77 0.75; 0.38 0.38 0.1 0.1] [0.76;0.2]};
ob12={ [0.4 0.42 0.42 0.4; 0.38 0.38 0.15 0.15] [0.41;0.2]};
ob13={ [0.1 0.4 0.4 0.1; 0.3 0.3 0.28 0.28] [0.3;0.29]};
ob14={ [0.2 0.3 0.3 0.2; 0.2 0.2 0 0] [0.25;0.1]};
ob15={ [0.4 0.42 0.42 0.4; 0.1 0.1 0 0] [0.41;0.05]};
ob16={ [0.5 0.6 0.6 0.5; 0.3 0.3 0 0] [0.55;0.2]};
ob17={ [0.98 1 1 0.98; 0.4 0.4 0.38 0.38] [0.99;0.39]};
    
```

Figure III.8 : Spécification des coordonnées des vertex des différents obstacles.

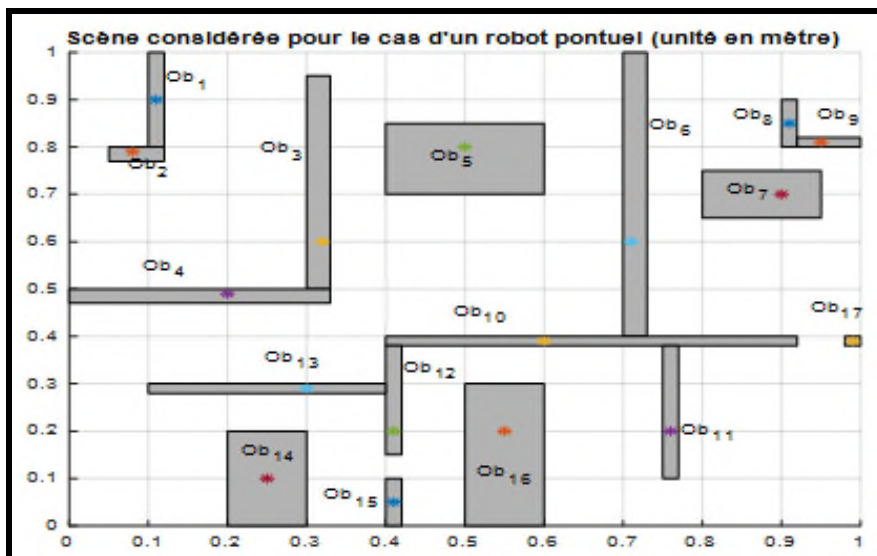


Figure : III.9 : Scène considérée pour le cas d'un robot ponctuel et les obstacles spécifiés : figure (III-6)

III.5.1. Cas d'un robot ponctuel

Le premier cas que nous avons considéré dans nos simulations est celui d'un robot ponctuel qui se déplace entre x_{init} et x_{goal} . Pour résoudre ce problème de planification, nous distinguons deux cas :

Le premier cas : nous utilisons le WSAPF simple.

Le deuxième cas : nous utilisons le WSAPF amélioré.

Le planificateur n'arrive pas à trouver un chemin entre x_{init} et x_{goal} avec la résolution correspondant à une grille de 33x33. Nous avons été obligés de passer à la résolution suivante avec une grille de (65x65). Pour éclaircir la cause du problème et voir l'effet de la résolution sur la planification nous donnons dans la figure (III.10) le bitmap de l'espace de travail correspondant à une résolution de (33x33).

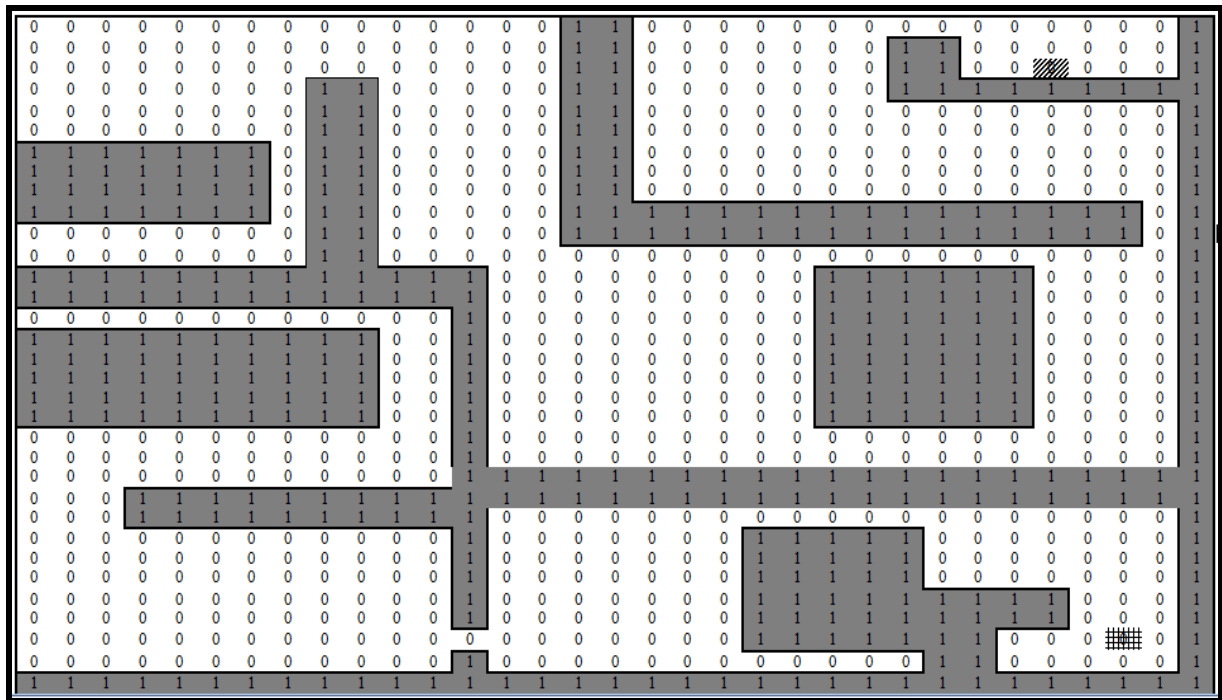


Figure III.10 : : Le bitmap de l'espace de travail (Workspace Bitmap) sous une résolution de 33x33.

On remarque la disparition complète du passage étroit entre les obstacles 12 et 15 sous la résolution 33x33. Aussi les obstacles 7, 8 et 9 se sont fusionnés dans un seul obstacle. Le passage étroit entre les obstacles 10 et 17 apparaît dans la grille comme ayant une largeur

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

égale à une cellule seulement alors qu'en réalité il a une taille de 6 cm. On voit que effectivement, il n'y a aucun passage entre x_{ini} et x_{goal} qui sont représentés par des cellules hachurées sur la figure. La première résolution sous laquelle le planificateur est arrivé à trouver un chemin est celle correspondant à une grille de (65x65). Le chemin obtenu en utilisant le champ de potentiel simple de l'espace de travail est donné sur la figure (III.11).

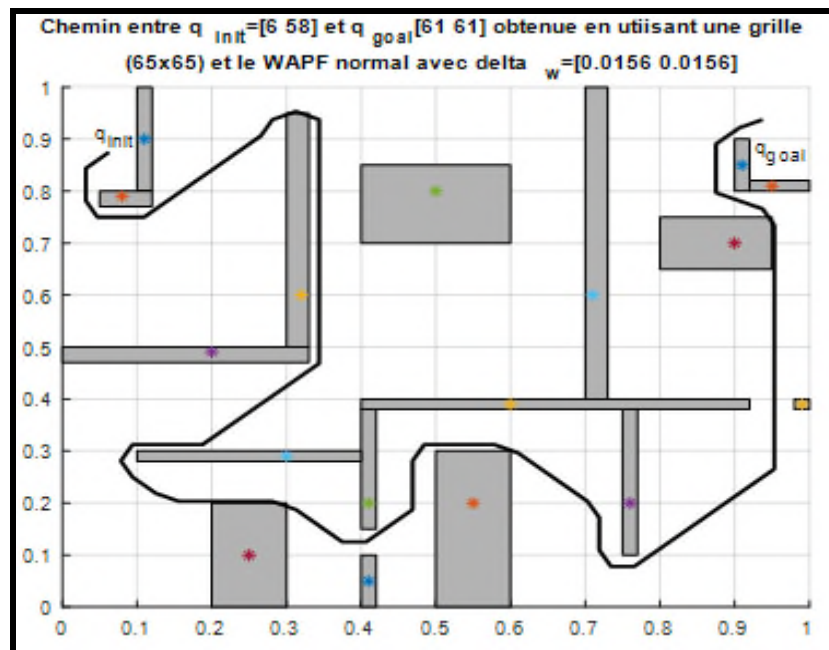


Figure III.11 : Chemin obtenue pour un robot ponctuel en utilisant le champ de potentiel artificiel simple dans l'espace de travail.

Pour mieux voir la différence entre les trajectoires obtenues en utilisant le WSAPF normal et le WSAPF amélioré, nous passons à une meilleure résolution avec une grille de (513x513). Ceci donnera un squelette plus complet de l'espace de travail ce qui fait apparaître l'avantage du WSAPF amélioré. Les chemins obtenus sont donnés par la figure (III.12).

On voit que l'utilisation du WSAPF normal donne des chemins qui s'approchent des obstacles ceci engendre des risques de collision. Par ailleurs, les chemins obtenus en utilisant un WSAPF amélioré s'éloignent tout le temps des obstacles ce qui est plus sûr concernant l'évitement de collisions et offre une marge de manœuvre qui permet d'échapper aux minimums locaux.

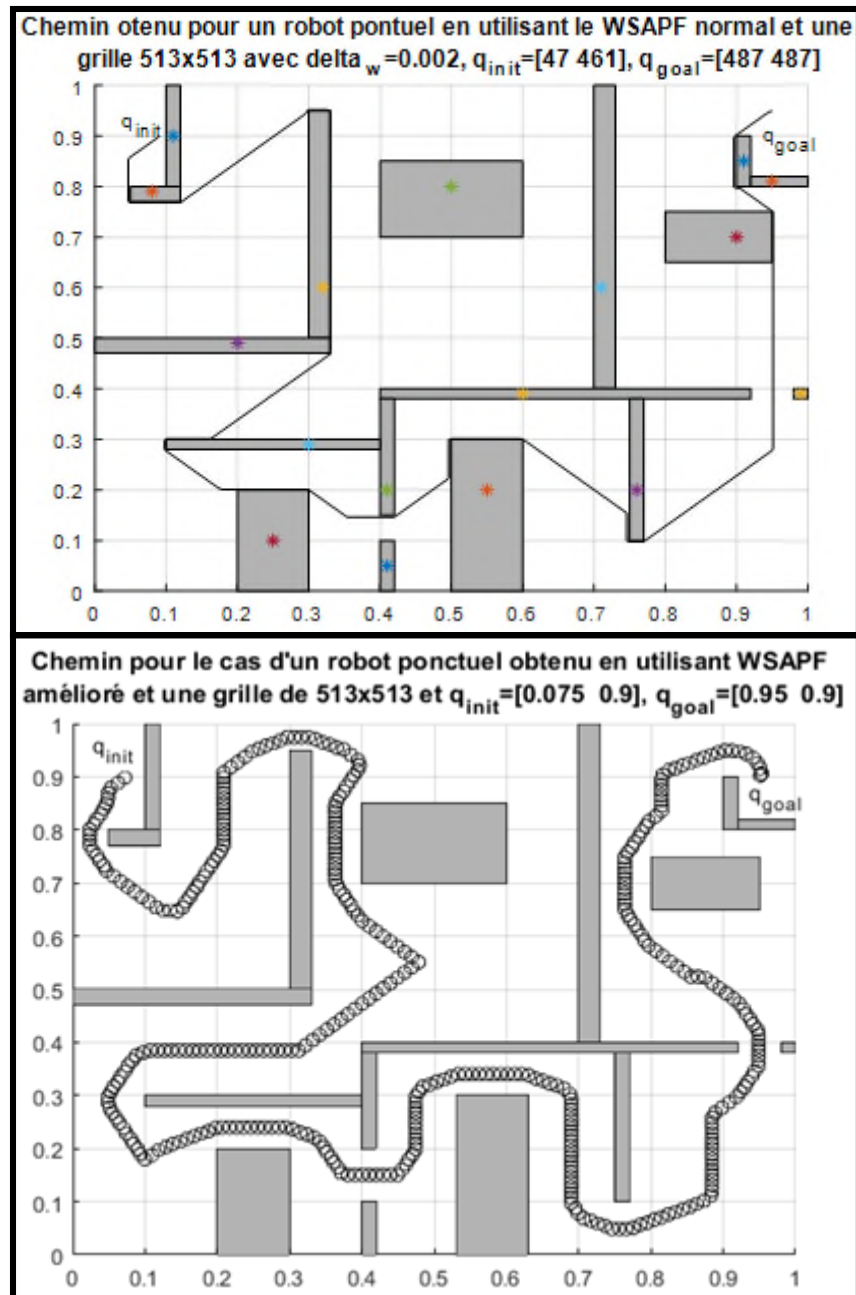


Figure III.12 :Chemins obtenus pour un robot ponctuel avec une résolution de 513x513 avec 1. WSAPF simple ; 2. WSAPF amélioré.

III.5.2. Cas d'un robot rectangulaire dans la scène 1

Pour la deuxième application, nous allons considérer le cas d'un robot rectangulaire se déplaçant dans une scène qui ressemble à la précédente sauf pour certaines modifications que nous avons introduites pour garantir l'existence de chemin entre une configuration initiale et une configuration finale. Les nouvelles spécifications sont :

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

```
ob12={ [0.4 0.42 0.42 0.4; 0.38 0.38 0.2 0.2] [0.41; 0.3] };  
qgoal=[0.9531 0.9063 1.5708]  
qinit=[0.0750 0.9000 1.5708]
```

Le robot effectue des manœuvres ne devient pas claire que si le mouvement est transformé en une séquence vidéo. Le robot doit retourner avant d'entrer dans la cavité contenant sa configuration de destination. Pour les cas de WSAPF simple, il y a une tentative de passer à droite de l'obstacle Ob7 qui a échoué car c'est impossible d'y passer. Cette tentative a été évitée pour le WSAPF amélioré car ce dernier s'éloigne des obstacles. Ce qui est commun entre les deux est que près de la configuration de destination, le robot tourne un nombre de fois dans l'une ou l'autre direction avant de s'orienter enfin correctement pour entrer.

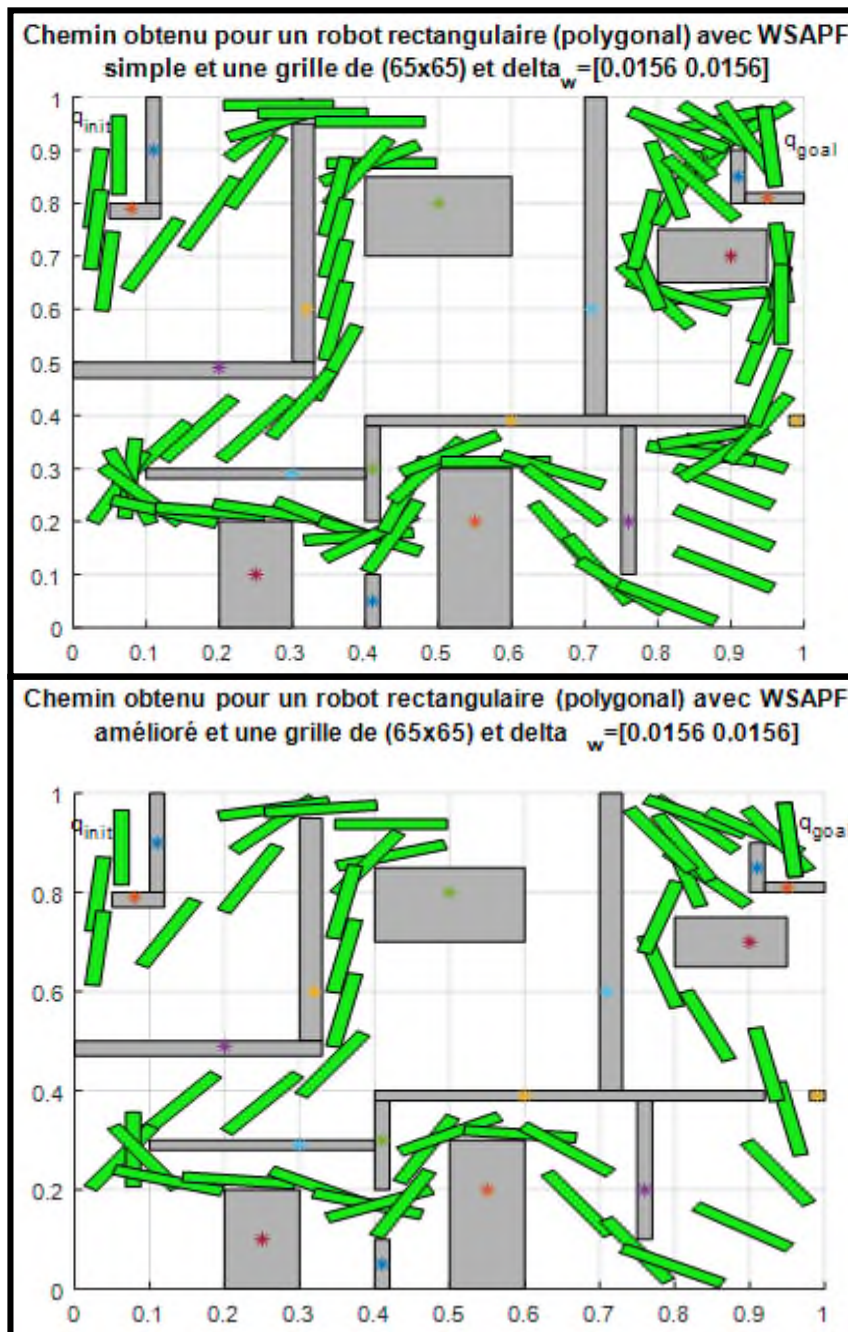
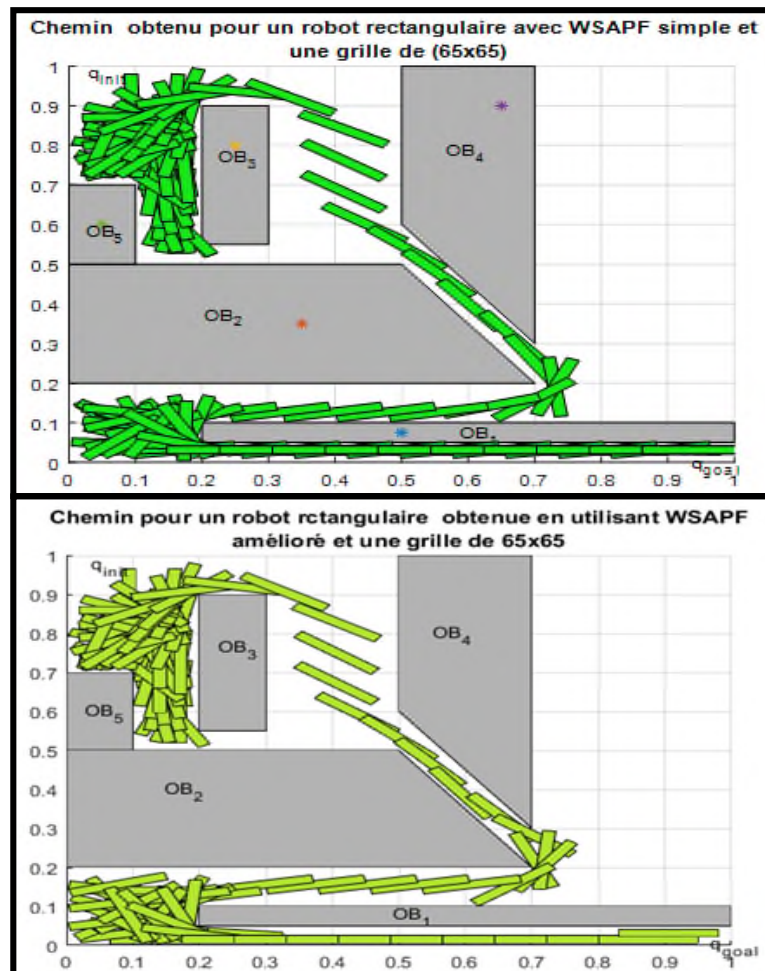


Figure III.13 : Chemin obtenu pour un robot polygonal avec une grille de (65x65) : 1. WSAPF simple 2.

WSAPF amélioré

III.5.3 Cas d'un robot rectangulaire dans la scène 2

Nous avons aussi fait une autre application avec un robot rectangulaire et une nouvelle scène contenant des obstacles polygonaux et nécessitant une manœuvre spéciale qui exige d'orienter le robot au préalable avant qu'il s'approche de sa configuration de destination. Le chemin que nous avons obtenu est donné sur la figure (III.14). Les parties les plus difficiles sont celles qui sont près des obstacles OB_3 et OB_1 où le robot doit être orienté avant de passer au dessous OB_1 . Pour le cas du WSAPF simple et au début de son chemin, le robot tente deux fois de passer sous OB_3 ce qui n'était pas possible. A l'entrée de la cavité sous l'obstacle OB_1 la compétition entre les points de contrôle est très claire. Le robot entre sous l'obstacle OB_1 jusqu'à ce qu'il atteigne son point de destination dans l'orientation contraire à ce qui est désiré, ensuite il ressort complètement pour pouvoir s'orienter correctement et il rentre de nouveau. Pour le cas du WSAPF amélioré, ceci n'est pas rencontré ce qui permet de conclure que le fait de s'éloigner des obstacles permet d'éviter certains minimum locaux.



Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

Figure III.14 : Chemin obtenu pour un robot polygonal avec une grille de (65x65) : 1. WSAPF simple ; 2. WSAPF amélioré. $q_{init} = [0.075 \ 0.9 \ \pi/2]$; $q_{goal} = [0.9 \ 0.03 \ 0]$;

III.5.4. Cas d'une chaîne articulée de 3 ddl

L'application que nous avons considérée est celle d'une chaîne articulée de 3 ddls (robot manipulateur de 3 ddls) se déplaçant entre une configuration initiale et une configuration finale. Les spécifications des obstacles de la nouvelle scène ainsi que des 3 segments du robot sont comme suit :

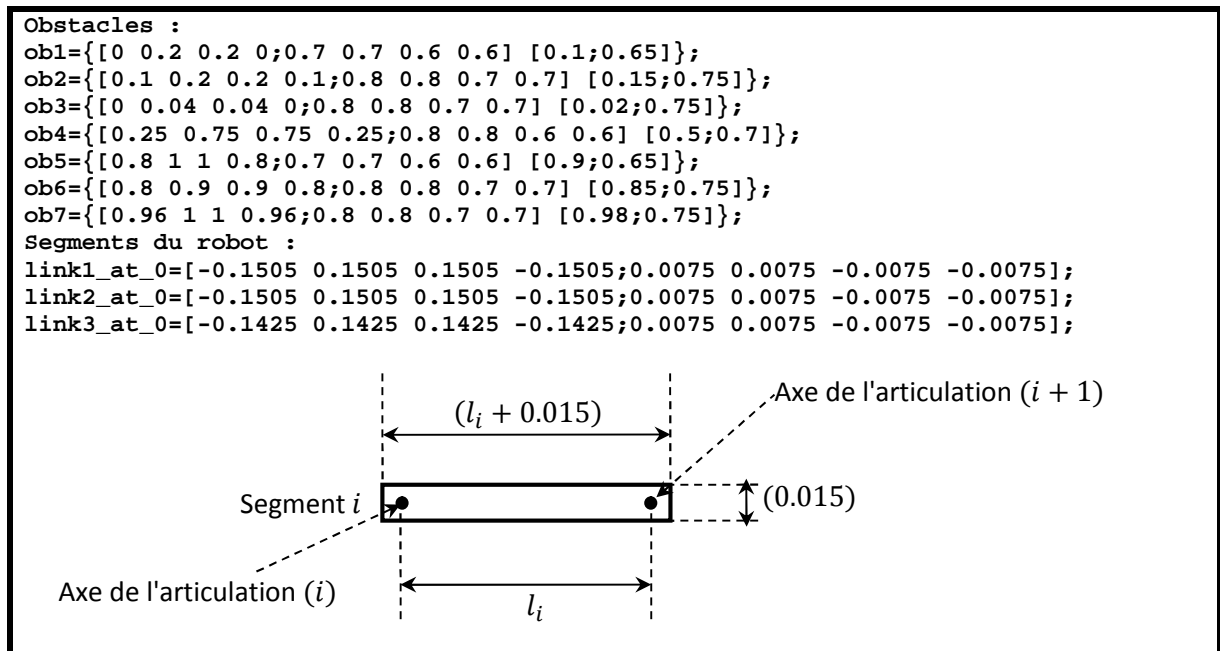


Figure III.15 : Spécification du robot manipulateur et des obstacles de la scène considérée.

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

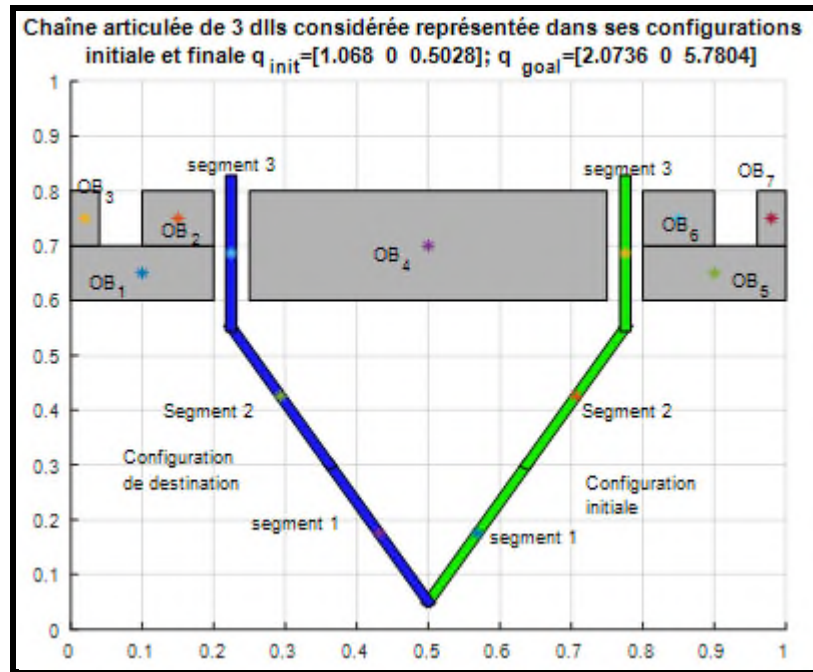


Figure III.16 : Scène et robot considérés pour la quatrième application.

Il faut noter que la complexité de cette application est proportionnelle au nombre de degré de liberté de robot manipulateur. Cependant, dans le cas de chaîne articulée nous sommes obligés de considérer l'arithmétique modulo 2π pour les angles articulaires (c'est à dire que $2\pi + \varepsilon \equiv \varepsilon$) pour pouvoir résoudre les requêtes de planifications désirées. Les valeurs négatives des angles sont traitées de la même façon. Aussi, pour le cas de la chaîne articulée, on doit considérer trois points de contrôle au lieu de 2 pour le cas d'une barre. Un point de contrôle est affecté à la deuxième extrémité de chaque segment à une distance l_i de l'axe de rotation précédent. Aucun chemin n'a été trouvé dans les grilles de dimension (33×33) et (65×65) . Il a fallu passer vers la grille (129×129) pour obtenir un chemin entre la configuration initiale et la configuration finale. La difficulté que représente ce problème réside dans la manœuvre nécessaire pour la pénétration du segment 3 entre les obstacles OB_1 et OB_4 .

Dans la figure (III.17), nous donnons le résultat de planification. Dans cette figure nous avons considéré seulement certaines configurations pour avoir une figure claire. Les détails sont plus clairs dans une séquence vidéo. On remarque que les deux premiers points de contrôle atteignent leurs positions de destinations avant le troisième, ce qui rend difficile la pénétration du troisième segment entre les obstacles OB_1 et OB_4 et le robot fait plusieurs tentatives sans succès et à la fin il se repositionne de nouveau pour qu'il puisse atteindre sa configuration finale. La majeure partie des configurations (près de 637 configurations d'un total de 837 configurations) du chemin obtenu est dans cette partie.

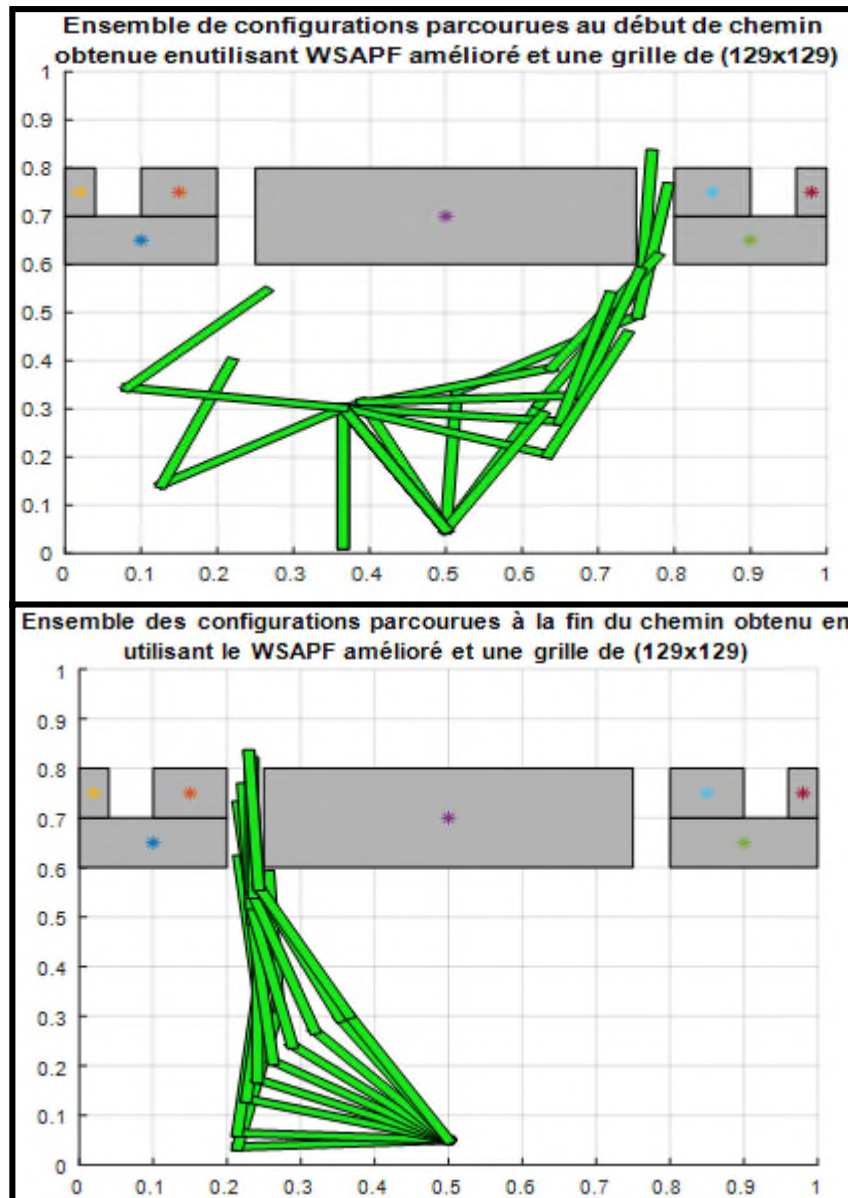


Figure III.17 : Planification du chemin pour un manipulateur plan de 3 ddls en utilisant WSAPF amélioré.

Avec le WSAPF simple, il a fallu que le robot passe par 2560 configurations avant qu'il atteigne sa configuration de destination. Le fait que le premier point de contrôle atteigne sa destination, il rend le positionnement de robot impossible pour pénétrer entre les obstacles OB_1 et OB_4 . Nous avons refait le même problème en considérant une autre fonction d'arbitrage $G(V_{p_i}) = \max_{i=1,2,3}(V_{p_i})$ qui augmente la compétition entre les points de contrôle. Cette fonction a donné un résultat meilleur avec un chemin nécessitant seulement 1032 configurations.

Chapitre III : Planification de chemins de robots en utilisant le champ de potentiel artificiel numérique

III.5.5. Cas d'une chaîne articulée de 4 dds

La dernière application que nous avons considérée est celle d'une chaîne articulée de 4 dds. La planification a duré quelques heures. Ceci est dû en une partie au fait qu'il fallait passer à une grille de (129×129) pour trouver un chemin et de l'autre partie au fait que la dimension de l'espace de recherche et maintenant égal à 4. Ceci implique l'exploration (dans le pire des cas) de (129×129) cellules dans l'espace de configurations et pour chaque cellule on a $k = (3^4 - 1) = 80$ voisins-4 pour lesquels il faut faire le test de collision pour trouver la prochaine cellule vers laquelle il faut aller. Ceci fait que cette façon de faire devient impraticable pour des dimensions supérieures à 4.

La chaîne articulée que nous avons considérée est similaire à la précédente avec l'addition d'un quatrième segment dont les spécifications sont :

```
link5_at_0=[-0.0625 0.0625 0.0625 -0.0625;0.0075 0.0075 -0.0075 -0.0075];
```

Dans la figure (III.18) nous donnons les configurations initiale et finale désirées :

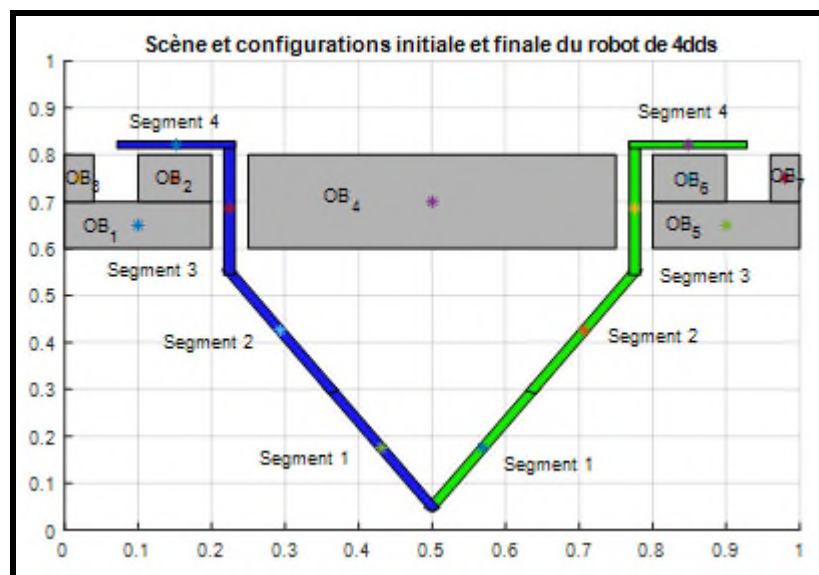


Figure III.18 : Configurations initiale et finale de la chaîne articulée de 4 dds.

Le chemin obtenu contient 105721 configurations et plusieurs manœuvres sont en réalité inutiles ceci signifie que la technique nécessite un traitement postérieur pour améliorer la qualité du chemin obtenu. Dans la figure qui suit nous portons le résultat obtenu dans lequel seuls les configurations de départ et de fin les plus utiles sont donnés.

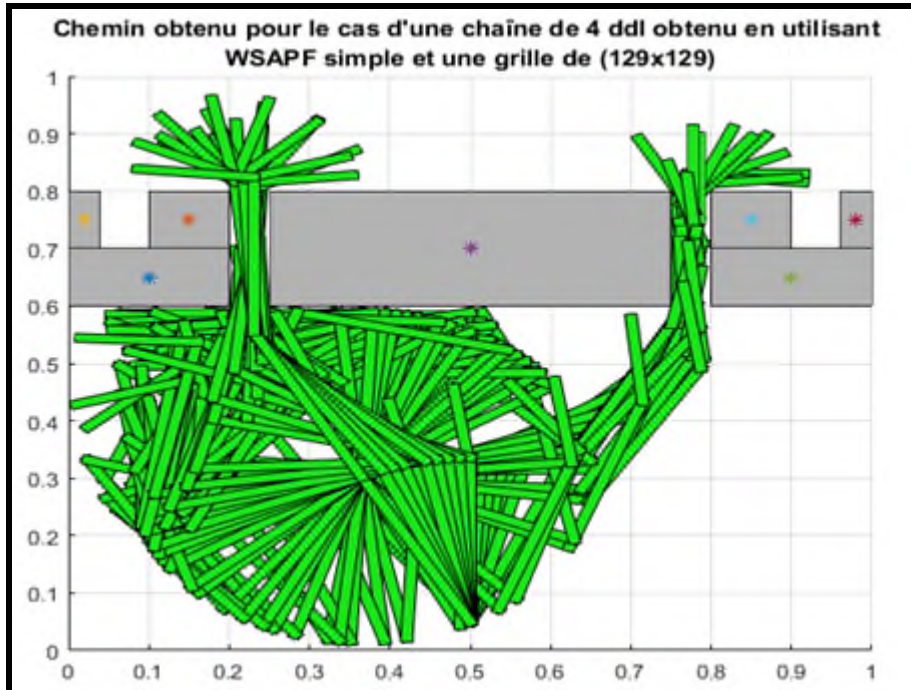


Figure III.19 : Représentation de certaines configurations obtenues pour la chaîne de 4ddl.

III.6. Conclusion

Dans ce chapitre nous avons décrit la méthode de planification de trajectoire en utilisant le champ de potentiel artificiel numérique et nous avons donné les algorithmes impliqués dans la technique. Des applications ont été effectuées pour montrer les propriétés de la méthode. Pour les problèmes à un nombre de degré de liberté réduit, la méthode est très efficace et des problèmes de planification ont été résolus en une fraction de seconde. L'utilisation de champs de potentiel amélioré de l'espace de travail permet d'obtenir un champ de l'espace de configuration avec moins de minimums locaux et améliore l'efficacité de la méthode et la qualité du chemin obtenu. Pour le cas de problème à plusieurs degrés de liberté, la méthode nécessite d'être modifiée surtout dans la façon de sortir des minimums locaux. Au lieu de remplir complètement le puits du minimum local en utilisant l'algorithme de best-first, il faut utiliser les méthodes probabilistes pour générer des mouvements aléatoires jusqu'à ce que le minimum soit quitté.

Conclusion générale

En rédigeant ce mémoire, nous avons pu approfondir et enrichir nos connaissances dans le domaine de la robotique, et en particulier dans la planification de trajectoires et la programmation des trajectoires pour robots manipulateurs.

En effet, la planification de trajectoires pour robots manipulateurs tout en évitant les collisions avec les obstacles, nécessite la création d'un Bitmap qui est une grille composée de cellule de résolution variant entre 16x16 et 512x512 et assurant une meilleure définition de l'espace de travail.

Il y a deux étapes qui caractérisent la méthode du champ potentiel, la première, est le champ potentiel de l'espace de travail dont la fonction essentielle est la création d'un gradient placé au niveau des points de discrétisation qui pointent vers la cible, en attirant les points de contrôle. La deuxième étape est le champ potentiel de l'espace de configuration combinant les champs potentiels de l'espace de travail lui-même combiné avec une fonction d'arbitrage assurant le choix adéquat entre les points de contrôle entrant en compétition.

La méthode du champ de potentiel permet la planification de trajectoire et assure l'arrivée à la destination finale du robot tout en évitant les obstacles et en faisant sortir le robot des minimums locaux.

La solution proposée en vue de répondre aux questions formulées dans notre problématique, a été vérifiée et validée. En effet, la méthode du champ potentiel artificiel est bien la méthode la plus simple et efficace qui permet la planification de trajectoire pour robot manipulateur.

On doit toutefois signaler que la technique de planification de trajectoires avec la méthode du champ potentiel artificiel est de nature complexe et nécessite beaucoup de temps pour déboucher sur des résultats de simulation. Cette complexité est elle-même proportionnelle au nombre de degrés de libertés du robot.

Enfin pour terminer cette conclusion, nous voulons exprimer notre souhait et notre vœux de pouvoir continuer de travailler sur le même thème, en l'approfondissant dans un travail de recherche dans le cadre d'une thèse de doctorat, et plus particulièrement en relation avec la détermination de trajectoires pour robot conçus pour assister les personnes handicapés dans leurs maisons.

Références bibliographiques

Références bibliographiques

- [1] www.larousse.fr/dictionnaires/français/technique (consulté le 18/06/2019).
- [2] www.linternaute.fr/dictionnaire/fr/definition/automatique (consulté le 18/06/2019).
- [3] www.le.dictionnaire.com/definition/robot (consulté le 18/06/2019).
- [4] Dombre Etienne. *Analyse et modélisation des robots manipulateurs*. LAVOISIER, 2001.
- [5] *Planète robots*, le magazine de la robotique et les nouvelles technologies de futur par Équipe rédactionnelle 24 avril 2016 » Consulté le 22/04/2019 à 15h30.
- [6] Sibylle Aoudjhane. *Metafly, l'objet volant comme un oiseau conçu en France, "l'usine nouvelle"*, 09/03/2019.
- [7] Bastien L. *Interface homme machine le magasin I.A Cloud et big data*, lien (<https://www.lebigdata.fr>) Consulté le 05/05/2019 à 10h45.
- [8] Boubekeur Mendil. *Modélisation d'un robot manipulateur*, support de cours, Université de Bejaia, 2019.
- [9] Wissama Khalil. *Modeling, Identification and control of robots* 2004.
- [10] Kerraci Abdelkader. *Synthèse des commandes robustes des robots manipulateurs rigides*. Thèse de magistère, Université d'Oran, 2004
- [11] J. Craig John. *Introduction to robotics, Mechanics and control*. Pearson Education International, 2005.
- [12] Sciavicco Lorenzo and Siciliano Bruno. *Modelling and Control of Robot of Robot Manipulators*. Springer-Verlag London, 1999.
- [13] M. Murray Richard, Zexiang Li, S. Shankar Sastry, and S. Shankara Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [14] C. Samson, M. Le Borgne, & B. Espiau. *Robot Control : the task function approach*. Oxford engineering science series. Clarendon Press, 1991. URL <https://books.google.fr/books?id=-DZSAAAAMAAJ>.
- [15] J.S. Yu & P.C. Muller. 'An on-line cartesian space obstacle avoidance scheme for robot arms. *Mathematics and Computers in Simulation*, 41(5_6): 627_637, 1996
- [16] B. Daachi, T. Madani, & A. Benallegue. Adaptive neural controller for redundant robot manipulators and collision avoidance with mobile obstacles. *Neurocomputing*, 79(0): 50_60, 2012.
- [17] M. D. G. Marcos, J.A. T. Machado, & T. P. Perdicoulis A. Trajectory planning of redundant manipulators using genetic algorithms *Communications in Nonlinear Science and Numerical Simulation*, 14(7): 2858_2869, 2009.

- [18]M. Pasquier. *Planification de trajectoires pour un robot manipulateur*. PhDthesis, Institut National Polytechnique de Grenoble, 1989.
- [19] D. P. Garg& M. Kumar. 'Optimization techniques applied to multiple manipulators for path planning and torque minimization. *Engineering Applications of Artificial Intelligence*, 15(3_4): 241_252, 2002.
- [20]T. Chakraborti, A. Sengupta, A. Konar, & R. Janarthanan. Application of swarm intelligence to a two-fold optimization scheme for trajectory planning of a robot arm. In *Swarm, Evolutionary, and Memetic Computing*, vol. 7077,pp. 89_96. Springer Berlin Heidelberg, 2011
- [21]O.khatib.Real-time obstacle avoidance for manipulators and mobile robots.In*Robotics and Automation. Proceedings.1985 IEEE International Conference on*,pp.500-505,1985.
- [22]LaValle, Steven M, *Planning Algorithms*, Cambridge University Press, 2006.
- [23]Christer E, *Real-Time Collision Detection*, CRC Press, 2004.
- [24] Jiménez, Juan José, Rafael J. Segura, and Francisco R. Feito, *Efficient collision detection between 2D polygons*, (2004).

Résumé

L'objectif de ce travail de Master concerne l'étude de la planification de trajectoire sans collision pour un robot manipulateur en utilisant l'approche de champs de potentiels artificiels. Le travail a été abordé par une exploration générale de la robotique. Puis, on a étudié la planification de trajectoire. La dernière partie concerne le fond du travail qui est la simulation sous Matlab d'un mouvement de robot dans une scène contenant des obstacles afin qu'il puisse atteindre sa destination finale sans se heurter aux obstacles en utilisant l'approche de champ potentiel artificiel.

Abstract

The aim of this master work is the study of trajectory planning without collision for a manipulator robot using the artificial potential field approach. The work was tackled by a general exploration of the field of robotics. Then ,the planning of trajectory have been studied the last part includes the essential of the work wich is the simulation with the software Matlab of a robot movement in a field contaning obstacles so that it can reach its final destination by avoiding obstacles using the artificial potential field approach.

ملخص

الهدف من هذا البحث هو دراسة تخطيط مسار سير روبوت مناور دون تصادمه مع العقبات باستخدام نهج مجال القوى الكامنة الاصطناعي. تمت معالجة العمل من خلال استكشاف عام للروبوتات, ثم درسنا تخطيط المسار. يتعلق الفصل الأخير بمحاكاة Matlab لحركة الروبوت في مشهد يحتوي على عقبات حتى يتمكن من الوصول إلى وجهته النهائية دون مواجهة العقبات باستخدام النهج المذكور.

Agzul :

Iswi seg unadi-yagi d leqraya n wahil n tikli n urubu swakken ur iṭemlili ara d wuguren, ayagi s usexdem n ṭṭawil n iger n tezmert yeffren. Aḥric aneggaru anaf deg-s amek i njaṛṛeb arubu-nney s usexdem n MATLAB akken a netḥeqqeq d akken abrid-is ur ikkat ara uguren.