

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A.MIRA-BEJAIA

Faculté de Technologie

Département de génie électrique



Mémoire présenté en vue de l'obtention du diplôme de

Master

Spécialité : Automatique et Système

Thème :

**Planification de chemins pour robots dans un monde polyédrique par
la méthode des champs de potentiel artificiel numérique avec une
recherche randomisée**

Présenté par :

OUTEMZABET Lydia

DJOUADI Hafidh

Soutenu le : 15/09/2020

Devant le jury composé de :

H. HADDAR

Encadreur

Université de Bejaïa

B. MENDIL

Examineur

Université de Bejaïa

F. YAHIAUOI

Examineur

Université de Bejaïa

Remerciements

*Avant tout, nous remercions **ALLAH** tout puissant de nous avoir donné le courage et la patience pour terminer ce travail.*

*Nous tenons également à exprimer nos vifs remerciements à tous ceux qui nous ont aidés durant cette année et **les responsables du département ATE** et surtout à tous nos enseignants qui ont contribué à notre formation, spécialement notre encadreur **M.HADDAR** pour l'intérêt qu'il a apporté à ce travail.*

*Nous remercions **les membres de jury** d'examen pour l'honneur qu'ils nous font en participant au jugement de ce travail.*

*Finalement, nous exprimons nos profonds gratitude à **nos parents** et notre famille pour leurs soutient moral et matériel. Qu'il soit de même pour tous **nos ami(e)s** dont le soutient indéfectible en amitié.*

Merci à tous !

Liste des figures :

Figure I.1 : Schéma Structurel générale d'un robot manipulateur.

Figure I.2 : Exemple de types d'architectures.

Figure I.3 : Symboles de l'articulation rotoïde.

Figure I.4 : Symboles de l'articulation prismatique.

Figure I.5 : Association des repères selon la convention DH.

Figure I.6 : Association des repères selon la convention MDH.

Figure II.1 : Illustration d'un espace de configuration 2D.

Figure II.2 : Illustration des potentiels primitifs.

Figure II.3 : Illustration de la combinaison des différents potentiels primitifs.

Figure II.4 : Illustration des sphères de sécurité pour un bras manipulateur à 3 *d.d.l.*

Figure II.5.a : Exemple d'un problème de minima local.

Figure II.5.b : Exemple de problème de rapprochement des obstacles.

Figure II.6.a : Fonction potentiel quadratique.

Figure II.6.b : Fonction potentiel conique.

Figure II.7 : Illustration d'une barrière du potentiel autour l'obstacle.

Figure II.8 : Algorithme de Gradient Descente à pas optimal

Figure II.9 : Illustration de la construction de $S(i, k)$.

Figure III.1 : Algorithme de calcul de WSAPF.

Figure III.2 : Algorithme de calcul de la distance L_1 des obstacles.

Figure III.3 : Algorithme de calcul du WSAPF amélioré.

Figure III.4 : Robot plan (4 *d.d.l*) et le chemin retourné par le planificateur à base d'APF déterministe.

Figure III.5 : Robot plan RRR et le chemin obtenu en utilisant APF amélioré randomisé.

Figure III.6 : Ensemble de poses intermédiaires le long du chemin obtenu en utilisant APF amélioré randomisé après élagage.

Figure III.7 : Robot plan RRR avec une région de configuration initiale encombrée et le Chemin après élagage.

Figure III.8 : Robot plan RRRR (4 *d.d.l*s) et le chemin obtenu en utilisant APF amélioré randomisé après élagage.

Figure III.9 : Robot plan (5 *d.d.l*s) et le chemin obtenu en utilisant APF amélioré randomisé après élagage.

Figure III.10 : Planification de chemin pour le cas d'une scène 3D et un robot RRR avec le planificateur randomisé utilisant le champ de potentiel artificiel numérique.

Liste des abréviations:

$d.d.l$	Degré de liberté
HRI	Interaction homme-robot
SMA	Système mécanisme articulé
q_i	Variables articulaires
T	Matrice de transformation homogène
P_A^B	Vecteur de position de l'origine du repère mobile (B) par rapport à (A)
R, Rot	Matrice de rotation
R_i^{i+1}	Matrice de rotation du repère r_{i+1} par rapport à r_i .
T_{trans}	Matrice de transformation
$T_{x,\alpha}$	Matrice de Rotation autour l'axe (ox)
$T_{y,\varphi}$	Matrice de Rotation autour l'axe (oy)
$T_{z,\theta}$	Matrice de Rotation autour l'axe (oz)
C_i	Corps de l'articulation (k).
K	Articulation
DH	Convention de Denavit-Hartenberg
MDH	Convention de Dénavint-Hartenberg modifié
MGD	Modèle géométrique direct
MGI	Modèle géométrique inverse
$J_n(q)$	Matrice Jacobéenne
MCD	Modèle cinématique direct
MDD, D	Modèle différentiel direct
W_n	Vitesse angulaire
W_i^{i+1}	Vitesse angulaire du corps C_{i+1} exprimée dans le repère r_i

W_i^i	Vitesse angulaire du corps C_i exprimée dans le repère r_i
V_n	Vitesse linéaire
V_i^{i+1}	Vitesse linéaire du corps C_i exprimée dans le repère r_i
V_{i+1}^{i+1}	Vitesse linéaire du corps C_i exprimée dans le repère r_{i+1}
MCI	Modèle cinématique inverse
Γ	Vecteur de (couple/force) des actionneurs
f_e	Vecteur représentant l'effort extérieur (forces et moments)
q	Vecteurs des positions articulaires
\dot{q}	Vecteur des vitesses articulaires
\ddot{q}	Vecteurs des accélérations articulaires
L	Lagrangien du système
E	Energie cinétique totale du système
P	Energie potentielle totale du système
$A(q)$	Matrice d'inertie du robot.
$C(q, \dot{q})$	Vecteur des forces de Coriolis et Centrifuges
$G(q)$	Vecteur des forces de gravité.
\mathcal{C}_{Space}	Espace de configuration
\mathcal{C}_{Obs}	Espace de configuration occupé par les obstacles
\mathcal{C}_{free}	Espace de configuration libre
\mathcal{W}_{Space}	Espace de travail
\mathcal{W}_{free}	Espace de travail libre
\mathcal{W}_{empty}	Espace de travail occupé par les obstacles
\mathcal{F}_e	référentiel cartésien \mathcal{F}_e associé à l'espace de configuration

$U(q), U_t(q)$	Fonction potentielle totale
$U_{Att}(q)$	Fonction potentiel attractive
$U_{Rep}(q)$	Fonction potentiel répulsive
$F(q), F_t(q)$	Force potentielle totale
$F_{Att}(q)$	Force potentielle attractive
$F_{Rep}(q)$	Force potentielle répulsive
$\rho(q, q_{goal})$	Distance euclidienne entre deux configurations
$WSAPF$	Champ de potentiel artificiel de l'espace de travail
$CSAPF$	Champ de potentiel artificiel de l'espace de configuration
G	Fonction d'arbitrage
GC	Grille d'occupation représentant l'espace de configuration
GC_{free}	Sous ensemble de la grille qui représente l'espace de configuration libre
GW	Grille d'occupation représentant l'espace de travail
GW_{empty}	Sous ensemble de la grille qui représente l'espace de travail occupé par les obstacles
$V_{p_i}(x)$	Champ de potentiel artificiel des points de contrôle dans l'espace de travail
$V_p(x)$	Champ de potentiel artificiel de l'espace de travail simple
L_i	Distance minimale entre la configuration initiale et finale à travers GW_{empty}
d_i	Distance minimale entre la configuration initiale et finale à travers GW_{empty} discrétisée
S	Ensemble des squelettes de l'espace de travail
itt_{max}	Nombre d'itérations maximal
A_{R_i}	Rayon d'attraction du minima local
N_{max}	Nombre de paire de mouvement gradient/Aléatoire

Table des matières

Remerciements

Introduction générale	1
-----------------------------	---

Chapitre I : Concepts de base en robotique

I.1 Introduction.....	4
I.2 Généralités sur la robotique	4
I.2.1 Définitions	4
I.2.2 Classification des robots	5
I.3 Structure générale d'un robot manipulateur	6
I.3.1 Système mécanique articulé	6
I.3.1.1 Types d'architecture de SMA	7
I.3.1.2 Types de liaisons mécanique	7
I.3.2 Actionneurs	8
I.3.3 Transmetteurs	9
I.3.4 Capteurs	9
I.3.5 Système de commande et gestion de taches	9
I.4 Modélisation d'un robot manipulateur	10
I.4.1 Matrice de transformations homogènes des repères	10
I.4.2 Modélisation géométrique	12
I.4.2.1 Description de la géométrie des robots manipulateurs	12
I.4.2.2 Modèle géométrique direct	15
I.4.2.3 Modèle géométrique inverse	15
I.4.3 Modélisation cinématique	17
I.4.3.1 Méthodes de calcul du modèle cinématique direct	17
I.4.3.2 Méthode de calcul du modèle cinématique inverse	20
I.4.4 Modélisation dynamique	22
I.4.4.1 Formalisme de Lagrange	22

I.4.4.2 Equation de mouvement d'un robot manipulateur	24
I.5 Conclusion	26
Chapitre II : Planification de trajectoire d'un robot guidée par des champs de potentiel numérique	
II.1 Introduction	28
II.2 Revue de la littérature	28
II.2.1 Formulation de problèmes de planification	28
II.2.2 Les approches proposées dans la littérature	29
II.2.3 Considérations générales	30
II.2.3.1 Eléments d'un planificateur	30
II.2.3.2 Espace de configuration	31
II.2.3.3 Contraintes	32
II.3 Choix de la méthode	33
II.4 Principe générale de la méthode	33
II.5 Champ de potentiel numérique	35
II.5.1 Champ de potentiel Attractif	36
II.5.2 Champ de potentiel Répulsif	38
II.5.3 Problème des minimas locaux	39
II.6 Génération de trajectoire guidée par champ de potentiel	39
II.6.1 Algorithme de gradient descente	39
II.6.2 Recherche du chemin de moindre cout	40
II.6.3 Vérification de collision	40
II.7 Conclusion	42
Chapitre III : Planification de chemins pour robots dans un monde polyédrique en utilisant le champ de potentiel numérique avec randomisation de la recherche	
III.1 Introduction	44
III.2 Construction du champ de potentiel artificiel	44
III.2.1 Calcul du champ de potentiel artificiel	45

III.2.1.1 Calcul du champ de potentiel de \mathcal{W}_{space} simple	47
III.2.1.2 Calcul du champ de potentiel de \mathcal{W}_{space} amélioré	48
III.2.1.3 Calcul du champ de potentiel de \mathcal{C}_{space}	51
III.2.2 Planification de chemins en utilisant l'APF (cas déterministe)	52
III.3 Planification de chemins en utilisant l'APF et une recherche randomisée	53
III.3.1 Mouvement selon l'inverse de gradient	54
III.3.2 Mouvement aléatoire	54
III.3.3 Méthode de recherche dans le graphe des minimas locaux	55
III.4 Résultats de simulation	56
III.5 Conclusion	64
Conclusion générale	66
Références Bibliographiques	

Introduction générale

Depuis quelques années, un intérêt croissant est porté au sein de la communauté robotique au développement de systèmes intelligents autonomes et de plateformes expérimentales dans le cadre de la robotique d'intervention. Un tel intérêt peut être perçu comme une conséquence logique à l'apparition d'applications potentielles et au désir de mettre les robots sur des nouvelles et diverses tâches. Dans de tels contextes, et tout comme dans le cas des applications traditionnelles de robotique mobile ou de manipulation, munir les systèmes robotiques d'une capacité de décision et plus particulièrement de planification de leurs mouvements reste l'un des éléments clé dans la mise en œuvre de leur autonomie.

Cependant, la planification de mouvement dynamique se définit comme le calcul de mouvements solutions pour un système qui part d'une situation initiale et doit rejoindre une situation objective tout en évitant des obstacles et en respectant certaines contraintes, de plus, c'est l'un des domaines qui ont attiré le plus de chercheurs en robotique et qu'un grand nombre de résultats ont été obtenus dans le cadre de sa résolution aussi bien en modélisation qu'en algorithmique. En outre, les comportements dynamiques non linéaires, les contraintes industrielles et surtout la difficulté même qu'il y a à résoudre ces problèmes de planification nous amèneront à développer plusieurs techniques différentes suivant les cas considérés.

Parmi ces techniques, la méthode de recherche utilisée dans l'approche de planification basée sur le champ de potentiel artificiel numérique sera l'objet d'étude du présent travail. Il se présentera en trois chapitres, selon l'ordre suivant :

- Dans le premier chapitre, nous débuterons par donner des notions générales sur la robotique qui nous permettent de montrer l'architecture mécanique des robots manipulateurs, ainsi, les types de tâches et les secteurs d'application. Pour commander un robot ou simuler son comportement, on doit disposer de modèles. Plusieurs niveaux de modélisation sont possibles selon les objectifs, les contraintes de la tâche et les performances recherchées: modèles géométriques, cinématiques et dynamiques qui doivent représenter le comportement du robot d'une façon réaliste.

- Dans le deuxième chapitre, nous allons formaliser les majeures problèmes de planification de mouvement pour les systèmes robotique, ainsi que, donner quelques notions de base en planification de trajectoire afin d'exposer la méthode basée sur les champs de potentiel artificiel numérique.

○ Dans le dernier chapitre, Nous commençons par donner un aperçu sur la procédure de construction du champ de potentiel et sur la méthode de recherche de chemin pour le cas déterministe. Et puis, nous introduisons une certaine randomisation dans la méthode de recherche utilisée précédemment pour robots dans une scène à 3D avec obstacles et segments du robot sous formes polyédriques.

Enfin, nous concluons ce mémoire en faisant le bilan de nos différentes contributions tout au long des chapitres, et l'analyse qualitative de l'ensemble des résultats obtenus.

Chapitre I

Concepts de base en robotique

I.1 Introduction :

Dans l'ensemble, l'étude du contrôle des manipulateurs n'est pas une science nouvelle, mais simplement un ensemble de sujets empruntés aux domaines classique.

Un robot manipulateur est considéré comme étant un système mécanique articulé, actionné et commandé. La conception et la commande d'un robot, exige à établir un modèle de mécanisme par des méthodes adéquates pour la description de leur morphologie, ainsi que, l'établissement des lois de commande, l'incorporation du robot dans les simulations informatiques, et la planification des tâches que nous présenterons dans ce chapitre.

I.2 Généralités sur la robotique :

Si les manipulations et les automates existant depuis nombreuse décennies, c'est seulement vers les années 1970 que sont véritablement apparues en exploitation industrielle les premières machines que l'on puisse qualifier comme robots. En effet, le robot est avant tout une machine industrielle poly-articulée, dotée de possibilité d'auto-adaptabilité et qui peut de ce fait agir d'une manière plus au moins autonome dans et sur son environnement [1].

I.2.1 Définitions :

- **La robotique :** C'est l'ensemble des activités de construction et de mise en œuvre des robots, on peut dire aussi que tout dispositif comporte une partie opérationnelle qui réalise la tâche et une partie décisionnelle ou commande qui contrôle la partie opérationnelle [2].
- **Le robot [2]:** Un robot est un dispositif mécanique articulé capable d'imiter certaines fonctions humaines telles que la manipulation d'objets ou la locomotion, dans le but de se substituer à l'homme pour la réalisation de certaines tâches matérielles, cette réalisation est plus ou moins autonome selon les facultés de perception de l'environnement dont est doté le robot.
- **Degré de liberté (*d.d.l*):** 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 [3].
- **Redondance :** 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 [1].
- **Porteur/Poignet :** on convient d'appeler les trois premières *d.d.l* d'un robot un porteur. Les *d.d.l* résiduels forment le poignet.

- **Commande** : ensemble des fonctions de commande logique et de puissance permettant de piloter et de commander la structure mécanique du robot et de communiquer avec l'environnement (matériels et utilisateurs).
- **Interaction homme-robot (HRI)** : Echange d'information et d'actions entre l'homme et le robot pour exécuter une tâche, au moyen d'une interface utilisateur.

I.2.2 Classification des robots :

Selon le type de tâche, on retiendra pour notre part 03 types principaux :

1) Les manipulateurs [4]:

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, aucune fonction de décision ne lui permet de réagir dans le cas d'un événement inattendu, c'est pour ça ils sont appelés (automate). Les domaines d'application des manipulateurs sont : la peinture, le soudage par position, la manipulation des objets . . . etc.

2) Les télémanipulateurs [4]:

Le télémanipulateur ou robot de téléopération est un système mécanique poly-articulé et multifonctionnel capable d'assister l'homme dans les opérations effectuées en milieu hostile ou Supplée un handicapé. Il est constitué par deux entités : Un bras maître et un bras esclave. Le bras esclave reproduit instantanément le mouvement imposé par un opérateur à un bras maître. Ce dernier est constitué d'une boîte à boutons, d'une poignée de commande ou d'une structure articulée identique à celle du bras esclave. Malgré la présence indispensable d'un opérateur, ce système est doté de potentialité comparable aux robots évolués :

- Prise de décision lors de vision réduite.
- Retour d'effort vers le bras maître.
- Amplification ou atténuation des mouvements.
- Autonomie de l'esclave pour alléger la charge de travail à l'homme.
- Création de mouvements élémentaires autonomes.

L'application des télémanipulateurs est répandue dans les domaines nucléaires (manipulation des produits radioactifs), spatial (exploration des planètes), sous-marin (forage), militaire (Démontage) et médical (prothèse).

Pour cette classe, on peut citer :

3) Robot mobile autonome :

Des nombreux robots mobiles sont contrôlés à distance. Ils utilisent leurs capteurs pour donner un accès à distance à des zone dangereuse (nucléaire, incendie, sécurité civile, déminage), inaccessible (océanographie, spatial). De tels robots font appel à des logiciels sophistiqués.

4) Robot didactique :

Ils sont des versions au format réduit des précédents robots. Ils ont un rôle de formation et d'enseignement, ils peuvent aussi être utilisés pour effectuer des tests de faisabilité d'un poste robotisé.

5) Robot de service :

Exécute des tâches utiles pour des humains ou des appareillages, excluant les applications d'automatisation industrielle (Robot de service personnel, Robots de service professionnel, Robot d'assistance à la personne ... etc.

Dans ce qui suit, on se restreint aux robots manipulateurs.

I.3 Structure générale d'un robot manipulateur:

On distingue classiquement cinq éléments principaux dans un robot manipulateur :

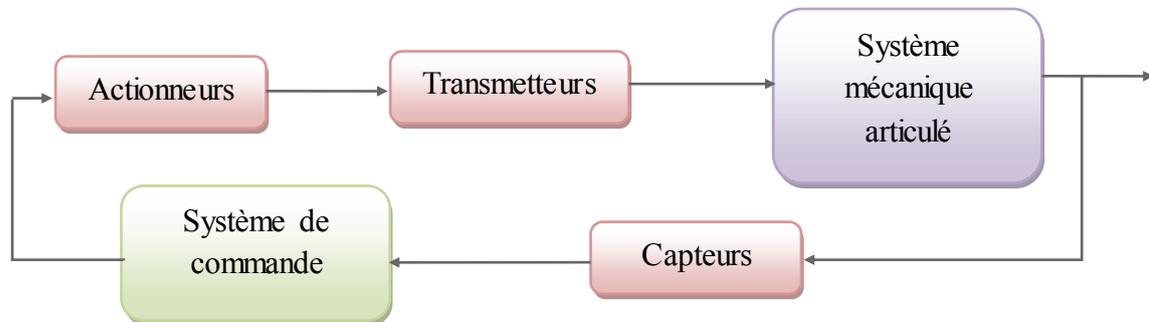


Figure I.1 : Structure générale d'un robot manipulateur [5].

I.3.1. Système mécanisme articulé (SMA) [6]:

C'est un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Il permet de remplacer, ou de prolonger, son action (le terme "manipulateur" exclut implicitement les robots mobiles autonomes). Son rôle est d'amener l'organe terminal dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données. Son architecture est une chaîne cinématique de corps, généralement rigides (ou supposés comme tels), assemblés par des liaisons appelées articulations.

1.3.1.1 Types d'architecture de SMA en robotique [2]:

Le nombre d'actionneurs (qui sera défini par la suite) représente ce qui d'usage d'appeler le *d.d.l* du robot. Les liaisons motorisées sont celles auxquelles sont associées les variables articulaires q_i de commande. C'est un moyen de distinguer les liaisons actives des liaisons passives.

- **Architecture série (ou chaîne cinématique ouverte) :**

Il n'y a qu'un chemin possible pour aller du bâti à la pince, les segments du robot ainsi que les liaisons sont bien mises en série.

- **Architecture parallèle (ou chaîne cinématique multi boucle) :**

Pour aller de la pince au bâti, il y a six chemins différents et sur chaque chemin, une seule liaison active prismatique.

- **Architecture mixte :**

On dit aussi série-parallèle ou parallèle-série selon la prédominance de liaisons en série ou en parallèle.

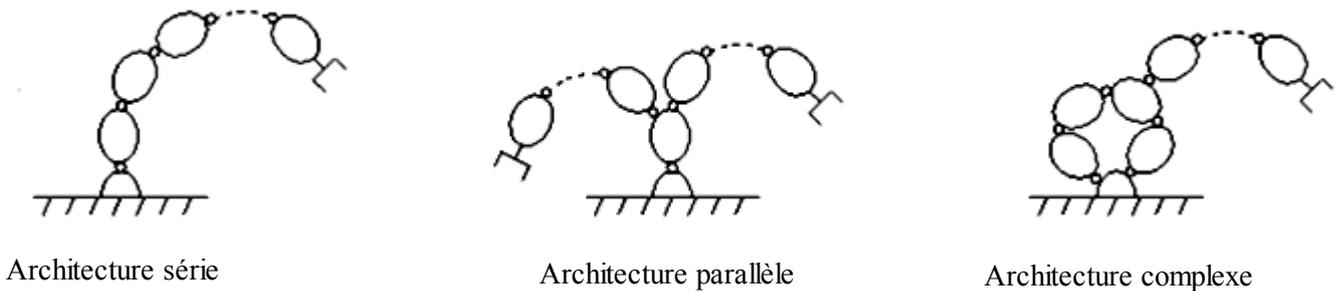


Figure I.2 : Exemple de types d'architectures.

1.3.1.2 Types des liaisons mécaniques (Articulations):

Dans l'assemblage de robots manipulateurs, les articulations les plus courantes sont :

- **Articulation rotoïde [6]:** Il s'agit d'une articulation de type pivote, notée « R », réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre les deux corps est donnée par l'angle (θ) autour de cet axe.



Figure I.3 : Symbole de l'articulation rotoïde.

- **Articulation prismatique [6]:** 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 (d) le long de cet axe.



Figure I.4 : Symbole de l'articulation prismatique.

- **Articulation rotule :** IL s'agit d'une articulation de type sphérique, notée S .
- **Articulation cardan :** IL s'agit d'une articulation de type joint universel, notée U .

I.3.2 Actionneurs [2]:

Pour être animé, le SMA comporte des moteurs le plus souvent avec des transmissions (l'ensemble constitue les actionneurs), des capteurs délivrant des informations sur l'état du SMA et à son extrémité un outil ou un préhenseur, ce que l'on qualifie plus généralement d'effecteur car c'est lui qui aura un effet direct sur l'environnement. On distingue:

- **Robots électriques :** Les actionneurs les plus fréquents utilisent des moteurs électriques. Les moteurs classiques ont une vitesse de régime élevée par rapport au niveau d'une articulation pivot, le moteur sera suivi d'un réducteur, ce qui permet d'avoir une amplification du couple moteur.
- **Robots hydrauliques:** Pour les robots devant manipuler de très lourdes charges, les actionneurs sont le plus souvent hydrauliques, agissant en translation (vérins hydrauliques) ou en rotation.

- **Robots pneumatiques (manipulateurs à cycles) :** Les actionneurs pneumatiques sont d'un usage général pour les manipulateurs à cycles. Un manipulateur à cycle (automate ou bras transfert), est un SMA, permettant une succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée

I.3.3 Transmetteurs :

Un transmetteur mécanique, est un élément de la chaîne d'énergie, qui fait le lien entre le capteur et le système de commande.

I.3.4 Capteurs :

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.3.5 Système de commande et gestion des tâches :

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.4 Modélisation d'un robot manipulateur :

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques, tels que [7]:

- les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal) et l'espace articulaire (dans lequel est définie la configuration du robot).
On distingue :
 - les modèles géométriques direct et inverse qui expriment la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement.
 - les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement.
- les modèles dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercés par les actionneurs et les positions, vitesses et accélérations des articulations.

I.4.1 Matrices de Transformations homogènes des repères [5]:

Un vecteur des coordonnées d'un point (p) dans un repère orthonormé: $op = (a, b, c)^T$ peut être écrit: $op = (x, y, z, w)^T$

Avec: $x = w.a$, $y = w.b$, $z = w.c$ w : facteur d'échelle (généralement égal à 1).

Ce qui permet de travailler avec des matrices (4×4) incluant, à la fois, la rotation et la position.

Une matrice de transformation homogène est de la forme:

$$T = T_A^B = {}_B^A T = \begin{pmatrix} R_{3 \times 3} & P_{3 \times 1} \\ N_{1 \times 3} & 1 \end{pmatrix} = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.1)$$

L'inverse de cette matrice est donné par :

$$T^{-1} = \begin{pmatrix} R_{3 \times 3}^T & -R_{3 \times 3}^T * P_{3 \times 1} \\ N_{1 \times 3} & 1 \end{pmatrix} \quad (1.2)$$

Avec :

$N_{1 \times 3}$: Représente la transformation de perspective (utilisée en vision).

$P_{3 \times 1} = P_A^B$: Est le vecteur de position de l'origine du repère mobile (B) par rapport à (A).

$R_{3 \times 3}$: Est la matrice de rotation.

Alors (n, s, a) représentent les axes de (B) dans (A) et $P_{3 \times 1}$ représente la position de l'origine de (B) par rapport à (A).

- **Matrice de Translation pure:**

Soit $T_{trans}(a, b, c)$ la transformation d'une translation pure ou (a, b, c) sont les translations le long des axes (x, y, z) respectivement. Quand l'orientation est conservée, la matrice de transformation de cette translation a la forme suivante :

$$T_{trans} = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.3)$$

- **Matrices de Rotation autour des axes principaux:**

1. La transformation correspondante à une rotation pure d'angle (α) autour de l'axe (ox) est donnée par :

$$T_{x,\alpha} = \begin{pmatrix} Rot(x, \alpha) & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & -s\alpha & 0 \\ 0 & s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.4)$$

2. La transformation correspondante à une rotation pure d'angle (φ) autour de l'axe (oy) est donnée par :

$$T_{y,\varphi} = \begin{pmatrix} Rot(y, \varphi) & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{pmatrix} = \begin{pmatrix} c\varphi & 0 & s\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -s\varphi & 0 & c\varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.5)$$

3. La transformation correspondante à une rotation pure d'angle (θ) autour de l'axe (oz) est donnée par :

$$T_{z,\theta} = \begin{pmatrix} Rot(z,\theta) & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{pmatrix} = \begin{pmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.6)$$

Une matrice $T_{(4 \times 4)}$ peut réaliser une transformation de coordonnées d'un point, exprimé dans (B) par le vecteur P^B , vers le repère (A) (ou bien l'inverse):

$$\begin{pmatrix} P^A \\ 1 \end{pmatrix} = T_A^B \cdot \begin{pmatrix} P^B \\ 1 \end{pmatrix} = \begin{pmatrix} R_A^B & P_A^B \\ 0_{1 \times 3} & 1 \end{pmatrix} \cdot \begin{pmatrix} P^B \\ 1 \end{pmatrix} \quad (1.7)$$

- **Matrice de transformation composée :**

- Si le repère (B) est tourné/translaté par rapport au repère fixe (A), alors on **pré-multiplie** par le résultat précédent (i.e., à gauche).
- Si le repère (B) est tourné/translaté par rapport à ses propres axes, alors on **post-multiplie** par le résultat précédent (i.e., à droite).

I.4.2 Modélisation géométrique :

I.4.2.1 Description de la géométrie des robots manipulateurs :

Une structure ouverte simple est composée de ($n+1$) corps notés C_0, \dots, C_n et de (k) articulations. Le corps C_0 désigne la base du robot et le corps C_n le corps qui porte l'organe terminal. L'articulation k_i connecte le corps C_i au corps C_{i-1} . Afin de définir la relation entre la position et l'orientation relative de deux corps consécutifs, les méthodes de Denavit-Hartenberg sont les plus répandus [7].

A) Convention de Denavit-Hartenberg (DH) [5]:

Pour décrire les relations de translation et de rotation entre corps adjacents, D-H proposent, en 1955, une méthode matricielle permettant d'établir systématiquement un repère pour chaque corps d'une chaîne articulée. En utilisant des matrices (4×4), on peut représenter le repère associé à chaque corps, au niveau de l'articulation, par rapport au repère du corps précédent. Ainsi, d'une manière séquentielle, on peut situer le poignet (orientation et position) par rapport au repère de base R_0 .

La méthode de description DH est fondée sur les règles suivantes :

- 1) Le repère R_{i-1} est placé à la fin du corps C_{i-1} .
- 2) L'axe z_{i-1} est selon l'axe de mouvement de l'articulation (k_i).
- 3) L'axe x_{i-1} est aligné entre les articulations (k_i) et (k_{i+1}). Si a_i est nul, alors x_{i-1} est perpendiculaire à z_{i-1} et z_i .
- 4) L'axe y_i complète le repère R_i pour former un repère « main droite »

Le passage du repère R_{i-1} au repère R_i s'exprime en fonction de quatre paramètres géométriques suivants la (figure I.5):

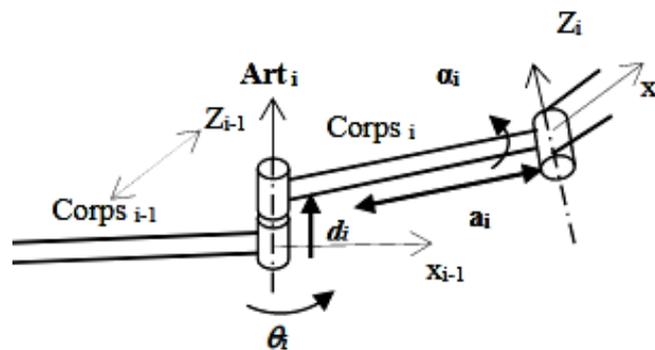


Figure I.5 : Association des repères selon la convention (DH) [5].

Avec :

- a_i : Distance entre z_{i-1} et z_i le long de x_i .
- α_i : Angle entre z_{i-1} et z_i correspondant à une rotation autour de x_i .
- θ_i : Angle entre x_{i-1} et x_i correspondant à une rotation autour de z_{i-1} .
- d_i : Distance de O_{i-1} à l'intersection de z_{i-1} avec x_i , le long de z_{i-1} .

La matrice de passage d'une articulation à la suivante est donnée par l'équation (1.8) :

$$T = T_{i-1}^i = T_{z,\theta_i} * T_{z,d_i} * T_{x,a_i} * T_{x,\alpha_i} = \begin{pmatrix} c\theta_i & -c\alpha_i.s\theta_i & s\alpha_i.c\theta_i & a_i.c\theta_i \\ s\theta_i & c\alpha_i.c\theta_i & -s\alpha_i.c\theta_i & a_i.s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.8)$$

Son inverse est donc :

$$T^{-1} = T_i^{i-1} = \begin{pmatrix} c\theta_i & -c\alpha_i.s\theta_i & s\alpha_i.c\theta_i & a_i.c\theta_i \\ s\theta_i & c\alpha_i.c\theta_i & -s\alpha_i.c\theta_i & a_i.s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.9)$$

B) Convention de Dénavit-Hartenberg modifié (MDH) [5]:

Dans la convention DH, le repère R_{i-1} est associé à l'articulation (k_i), par contre, dans la convention MDH, c'est le repère R_i qu'est associé à l'articulation (k_i) (c-à-d début du corps C_i).

Le passage du repère R_{i-1} au repère R_i s'exprime en fonction de quatre paramètres géométriques suivants la (figure I.6) :

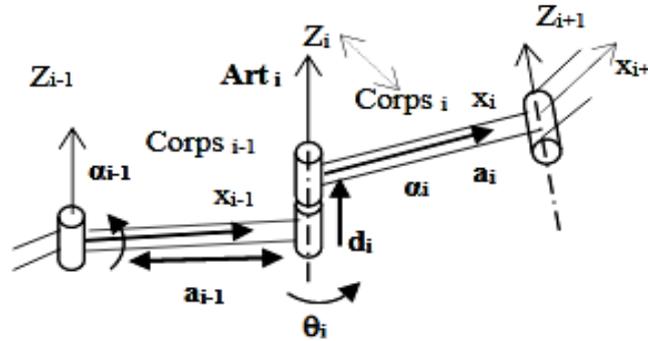


Figure I.6 : Association des repères selon la convention (MDH) [5].

Avec:

- a_i : Distance entre z_i et z_{i+1} , le long de x_i .
- α_i : Angle entre z_i et z_{i+1} , autour de x_i .
- θ_i : Angle entre x_{i-1} et x_i , autour de z_i .
- d_i : Distance entre x_{i-1} et x_i , le long de z_i .

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

$$T = T_{i-1}^i = T_{x,a_{i-1}} * T_{x,\alpha_{i-1}} * T_{z,\theta_i} * T_{z,d_i} = \begin{pmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ c\alpha_{i-1}.s\theta_i & c\alpha_{i-1}.c\theta_i & -s\alpha_{i-1} & -d_i.s\alpha_{i-1} \\ s\alpha_{i-1}.s\theta_i & s\alpha_{i-1}.c\theta_i & c\alpha_{i-1} & d_i.c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.10)$$

I.4.2.2 Modèle géométrique direct [7]:

Le modèle géométrique direct (MGD) est l'ensemble des relations qui permettent d'exprimer de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du robot, en fonction de ses coordonnées articulaires. Dans le cas d'une chaîne ouverte simple, il peut être représenté par la matrice de transformation :

$$T_{MGD} = T_0^n = T_0^1(q_1) T_1^2(q_2) \dots T_{n-1}^n(q_n) \quad (1.11)$$

Le modèle géométrique direct peut aussi être représenté par la relation suivante :

$$X = J(q) \quad (1.12)$$

Avec q étant le vecteur des variables articulaires tel que :

$$q = [q_1 \quad q_2 \quad \dots \quad q_n]^T \quad (1.13)$$

Les coordonnées opérationnelles sont définies par :

$$X = [x_1 \quad x_2 \quad \dots \quad x_n]^T \quad (1.14)$$

Soit avec les éléments de la matrice T_{MGD} :

$$X = [P_x \quad P_y \quad P_z \quad n_x \quad n_y \quad n_z \quad s_x \quad s_y \quad s_z \quad a_x \quad a_y \quad a_z]^T \quad (1.15)$$

I.4.2.3 Modèle géométrique inverse :

On a vu que le modèle géométrique direct d'un robot permettait de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires. Le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. On peut distinguer deux principales méthodes de calcul de modèle géométrique inverse :

1) Méthode de Paul :

Elle est basée sur l'identification successive des variables articulaires en connaissant la matrice MGD :

$$W_0 = T_{MGD} = T_0^n = T_0^1(q_1) T_1^2(q_1) \dots T_{n-1}^n(q_n) \quad (1.16)$$

On cherche à résoudre le système de l'équation (1.17) pour la situation désirée W_0 . Pour un robot de n d.d.l, on procède les opérations permettent d'isoler les variables d'articulations l'une après l'autres on pré-multipliant les deux membres de l'équation par les matrices T_{i-1}^i afin d'identifier la variable q_i de chaque articulation. Soit :

$$\left\{ \begin{array}{l} W_0 = T_0^1 T_1^2 T_2^3 \dots T_{n-1}^n \\ T_1^0 W_0 = T_1^2 T_2^3 \dots T_{n-1}^n = W_1 \rightarrow q_1 \\ T_2^1 W_1 = T_2^3 T_3^4 \dots T_{n-1}^n = W_2 \rightarrow q_2 \\ \vdots \\ T_{n-1}^{n-2} W_{n-2} = T_{n-1}^n = W_{n-1} \rightarrow q_{n-1} \end{array} \right. \quad (1.17)$$

Avec : $W_{j+1} = T_{j+1}^j W_j$ pour $j=0\dots n$.

2) Méthode de Pieper [5]:

Elle est basée sur la définition de l'équation de position et celle d'orientation. Pour un robot de n d.d.l :

- **Equation de position :**

La position de l'effecteur dépend de $(q_1 \ q_2 \ q_3)$. En comparant le vecteur de position désiré avec la dernière colonne de la matrice MGD, on peut calculer $(q_1 \ q_2 \ q_3)$.

- **Equation d'orientation :**

En utilisant les valeurs calculées de $(q_1 \ q_2 \ q_3)$, on peut calculer les variables restantes. Soit la matrice d'orientation $R_s = [n \ s \ a]$:

$$R_s = R_0^1 R_1^2 \dots R_{n-1}^n = R_0^3 R_3^n \quad (1.89)$$

D'où :

$$R_3^0 R_s = R_3^n \quad (1.19)$$

Le 1^{er} terme est connu. Il nous reste à calculer les q_i restants par mise en jeu dans 2^{ème} terme.

I.4.3 Modélisation cinématique :

I.4.3.1 Méthodes de calcul du modèle cinématique direct :

Le modèle cinématique direct (MCD) d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires [1]. Il s'écrit :

$$\dot{X} = J(q)\dot{q} \quad (1.20)$$

Ou, $J(q)$ désigne la matrice Jacobéenne cinématique de dimension $(m \times n)$ du mécanisme:

$$J_{i,j}(q) = \frac{\partial X_i(q)}{\partial q_j} \quad (1.21)$$

Avec : $i=1, \dots, m$ et $j=1, \dots, n$. et $\dot{q} = (\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n)^T$.

On peut obtenir la matrice Jacobéenne par deux méthodes :

- Par dérivation du MGD ou bien calcul du modèle différentiel direct.
- Par propagation de vitesses de translation et de rotation.

1) Modèle différentielle Direct (MDD) :

Le modèle différentiel direct (MDD) donne les variations élémentaires des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires. Cette méthode est facile à mettre en œuvre pour des robots à deux ou trois degrés de liberté.

Le calcul de la matrice jacobéenne peut se faire en calculant la variation différentielle de la matrice MGD [5]:

$$dT = [T_{trans}(dx, dy, dz) * Rot(k, dq) - I] * T = \Delta * T \quad (1.22)$$

Avec :

$T_{trans}(dx, dy, dz)$: Translation différentielle par rapport aux axes principaux.

$Rot(k, dq)$: Matrice de rotation différentielle autour de l'axe du repère de base.

dT : Variation différentielle de la matrice T.

En utilisant l'approximation, quand : $\theta \rightarrow 0$, $\sin\theta \rightarrow d\theta$ et $\cos\theta \rightarrow 1$, on peut écrire 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 comme suit:

$$T_{Rot}(\delta x, \delta y, \delta z) = \begin{pmatrix} 1 & -\delta z & \delta y & 0 \\ \delta z & 1 & -\delta x & 0 \\ -\delta y & \delta x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.23)$$

Ce qui donne :

$$\Delta = [Trans(dx, dy, dz) . Rot(k, dq) - I] = \begin{pmatrix} 0 & -\delta z & \delta y & dx \\ \delta z & 0 & -\delta x & dy \\ -\delta y & \delta x & 0 & dz \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.24)$$

La transformation Δ est construite des deux vecteurs $dP_n = (dx, dy, dz)^T$ et

$dr_n = (\delta x, \delta y, \delta z)^T$ qui peuvent être arrangés dans un seul vecteur colonne D qui représente le mouvement différentiel total (MDD):

$$D = \begin{pmatrix} dP_n \\ dr_n \end{pmatrix} = \begin{pmatrix} dx \\ dy \\ dz \\ \delta x \\ \delta y \\ \delta z \end{pmatrix} = J_n(q) * \dot{q} \quad (1.25)$$

Les vitesses instantanées de l'outil dans l'espace de travail peuvent s'écrire (MCD) :

$$\begin{pmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{pmatrix} = J_n(q) * \dot{q} \quad (1.26)$$

2) Méthode de la Propagation des Vitesses :

La détermination de la matrice $J_n(q)$ consiste à calculer les vitesses linéaires et angulaires en fonction des vitesses articulaires et ce en commençant de la base vers l'effecteur. Les vitesses sont décrites d'abord dans les repères associés aux articulations, puis dans le repère de base.

a) Cas d'une articulation rotoïde :

• Vitesses angulaires :

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

- Elle est décrite dans le repère r_i par:

$$W_i^{i+1} = W_i^i + R_i^{i+1} * \dot{\theta}_{i+1} * Z_{i+1}^{i+1} \quad (1.27)$$

- Elle est décrite dans le repère r_{i+1} par:

$$W_{i+1}^{i+1} = R_{i+1}^i * W_i^i + \dot{\theta}_{i+1} * Z_{i+1}^{i+1} \quad (1.28)$$

Avec :

$$\dot{\theta}_{i+1} * Z_{i+1}^{i+1} = [0 \quad 0 \quad \dot{\theta}_{i+1}]^T$$

W_i^{i+1} : Vitesse angulaire du corps C_{i+1} exprimée dans le repère r_i

W_i^i : Vitesse angulaire du corps C_i exprimée dans le repère r_i .

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

$\dot{\theta}_{i+1}$ Est un scalaire et Z_{i+1}^{i+1} indique l'axe de rotation.

• Vitesse Linéaire :

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 du corps C_i .

- Elle est décrite dans le repère r_i par :

$$V_i^{i+1} = V_i^i + W_i^i \times P_i^{i+1} \quad (1.29)$$

- Elle est décrite dans le repère r_{i+1} par :

$$V_{i+1}^{i+1} = R_{i+1}^i (V_i^i + W_i^i \times P_i^{i+1}) \quad (1.30)$$

Avec :

P_i^{i+1} : Vecteur aligné le long du corps C_i (rayon de rotation).

$W_i^i \times P_i^{i+1}$: Produit vectoriel qui correspond à la vitesse linéaire de l'origine du repère r_{i+1} induit par la rotation du corps C_i à la vitesse W_i^i .

b) Cas d'une articulation prismatique :

Dans le cas des articulations prismatiques :

• Vitesse angulaire :

$$W_{i+1}^{i+1} = R_{i+1}^i W_i^i \quad (1.31)$$

• **Vitesse linéaire :**

$$V_{i+1}^{i+1} = R_{i+1}^i (V_i^i + W_i^i \times P_i^{i+1}) + \dot{d}_{i+1} * Z_{i+1}^{i+1} \quad (1.32)$$

En appliquant ces équations successivement de la base vers l'effecteur, on peut calculer W_n^n et V_n^n .

Pour représenter ces vitesses par rapport au repère de base r_0 , il suffit de les multiplier par la matrice de rotation R_0^n :

$$\begin{cases} V_n = V_0^n = R_0^n V_n^n = F_1(\dot{q}_1, \dot{q}_2 \dots \dot{q}_n) \\ W_n = W_0^n = R_0^n W_n^n = F_2(\dot{q}_1, \dot{q}_2 \dots \dot{q}_n) \end{cases} \quad (1.33)$$

L'écriture de l'équation (1.33) sous forme matricielle, nous donne alors le modèle (MCD) :

$$\begin{pmatrix} V_n \\ W_n \end{pmatrix} = J_n(q) * \dot{q} \quad (1.34)$$

I.4.3.2 Méthodes de calcul du modèle cinématique inverse :

L'objectif du modèle cinématique inverse (MCI) est de calculer, à partir d'une configuration q donnée, les vitesses articulaires \dot{q} qui assurent au repère terminal une vitesse opérationnelle \dot{X} imposée, et cela revient au calcul de la matrice Jacobéenne inverse.

A) Cas régulier :

Dans ce cas, la matrice Jacobéenne est carrée d'ordre n et son déterminant est non nul. La méthode la plus générale consiste à calculer $J^{-1}(q)$ qui permet de déterminer la vitesse articulaire grâce à la relation :

$$\dot{q} = J^{-1} * \dot{X} \quad (1.35)$$

B) Cas de singularités :

Au voisinage des positions singulières, l'utilisation du modèle cinématique inverse classique peut donner des vitesses articulaires importantes, incompatibles avec les caractéristiques des actionneurs. Cependant, il existe des solutions et méthodes de traitement pour éviter tels problèmes :

1) Méthode des variables principales :

Cette méthode consiste à considérer une des variables comme étant nulle pour se retrouver avec une matrice carrée. Néanmoins, on ne sait pas quelle est la meilleure variable à annuler.

2) Méthode d'utilisation d'un critère :

Le principe de cette méthode consiste à rajouter une contrainte (nombre d'équations=nombre d'inconnues). En termes physique, ceci revient à exploiter les *d.d.l* en plus pour optimiser un certain comportement du robot. Pour cela, on peut utiliser un critère quadratique de la forme :

$$C = \frac{1}{2} \dot{q}^T * M * \dot{q} \quad (1.36)$$

Avec : M une matrice symétrique définie positive (le choix le plus simple est $M=I$).

Le Lagrangien résultant s'écrit :

$$L = \frac{1}{2} \dot{q}^T * M * \dot{q} + \lambda^T (\dot{X} - J * \dot{q}) \quad (1.37)$$

La solution optimale du MCI est donc :

$$\dot{q} = M^{-1} * J^T (J * M^{-1} * J^T)^{-1} * \dot{X} \quad (1.38)$$

3) Méthodes des inverses générales :

L'inverse généralisé de J est toute matrice G satisfaisant la relation :

$$G * J * G = G \quad (1.39)$$

La matrice pseudo-inverse J^+ consiste à choisir parmi les inverses généralisés possibles, celle qui assure

La minimisation de la norme euclidienne $\|\dot{q}^2\|$ et la norme de l'erreur $\|\dot{X} - J * \dot{q}\|^2$.

La solution MCI est donnée :

$$\dot{q} = J^+ * \dot{X} \quad (1.40)$$

I.4.4 Modélisation dynamique :

Le modèle dynamique et la relation entre les couples (et/ou) appliqué aux actionneurs et les positions, vitesses accélérations articulaires [7]. On représente le modèle dynamique par une relation de la forme :

$$\Gamma = f(q, \dot{q}, \ddot{q}, f_e) \quad (1.41)$$

Avec :

Γ : Vecteur de (couple/force) des actionneurs (selon le type d'articulations).

q, \dot{q}, \ddot{q} : Sont les vecteurs des positions articulaires, vitesses articulaires et accélérations articulaires (respectivement).

f_e : Vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

Le modèle dynamique direct est celui qui exprime les accélérations articulaires en fonction des positions, vitesse et couples des articulations. Il est alors représenté par la relation :

$$\ddot{q} = g(q, \dot{q}, \Gamma, f_e) \quad (1.42)$$

Plusieurs approches sont proposées pour obtenir le modèle dynamique des robots. Les plus souvent utilisés dans la robotique sont le formalisme de Lagrange, et le formalisme de Newton-Euler.

La formulation de Newton-Euler peut être considérée comme une approche basée sur l'équilibre des forces, la formulation de Lagrange est une approche basée sur l'énergie [8]. Les deux formalismes donnent les mêmes équations de mouvements pour le même manipulateur (Dans ce qui suit, on considère que les robots est à chaîne ouverte simple).

I.4.4.1 Formalisme de Lagrange :

La formulation de Lagrange décrit le comportement d'un système dynamique en termes d'énergie [9], lorsque l'effort extérieur sur l'organe terminal est supposé nul.

Les équations de Lagrange sont généralement écrites sous la forme :

$$\Gamma_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (1.43)$$

Avec :

L : Lagrangien du système égal à $(E - P)$.

E : Energie cinétique totale du système.

P : Energie potentielle totale du système.

Afin d'obtenir l'équation générale de la dynamique, on détermine le Lagrangien. Dans notre usage, q représentera le vecteur des variables, se composant de l'angle d'articulation θ_i et le déplacement de l'articulation d_i . Alors que Γ est un vecteur qui a comme composants les couples n_i correspondants aux angles θ_i , et les forces f_i correspondantes aux déplacements d_i .

a) Energie cinétique :

L'énergie cinétique de la $k^{i\text{ème}}$ articulation d'un corps C_i est donnée par :

$$E_i = \frac{1}{2} m_i V_{C_i}^T V_{C_i} + \frac{1}{2} W_i^{iT} I_{C_i}^i W_i \quad (1.44)$$

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 :

$$E(q, \dot{q}) = \sum_{i=1}^n E_i(q, \dot{q}) \quad (1.45)$$

Les vitesses $V_{C_i}^T$ et W_i^{iT} 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 la vitesse. Cependant, l'énergie cinétique doit être toujours positive, on utilise la formule quadratique de la forme :

$$E(q, \dot{q}) = \frac{1}{2} \dot{q}^T A(q) \dot{q} \quad (1.46)$$

Avec $A(q)$ matrice d'inertie carrée qui doit être définie positive.

b) Energie potentielle :

L'énergie cinétique de la $k^{i\text{ème}}$ articulation d'un corps C_i est donnée par :

$$P_i = m_i g_0^T T_0^{C_i} + P_{ref} \quad (1.47)$$

Avec :

\mathbf{g}_0 : Vecteur de gravité de dimensions (3×1) .

$\mathbf{T}_0^{C_i}$: Transformation homogène localisant le centre de la $k^{i\text{ème}}$ articulation.

\mathbf{P}_{ref} : Constante choisie de sorte que la valeur minimum de \mathbf{P}_i soit nulle.

L'énergie potentielle totale du manipulateur est la somme de l'énergie potentielle des différentes articulations :

$$P(q) = \sum_{i=1}^n P_i(q) \quad (1.48)$$

Puisque $\mathbf{T}_0^{C_i}$ sont en fonctions des variables articulaires \mathbf{q}_i , aussi que l'énergie potentielle.

Donc le Lagrangien sera :

$$L(q, \dot{q}) = \frac{1}{2} \dot{q}^T A(q) \dot{q} - P(q) \quad (1.49)$$

En se reportant à l'équation (1.49), il est convenable d'écrire l'énergie cinétique sous forme d'une somme :

$$L(q, \dot{q}) = \frac{1}{2} \sum_{i,j=1}^n A_{i,j}(q) \dot{q}_i \dot{q}_j - P(q) \quad (1.50)$$

I.4.4.2 Equation de mouvement d'un robot manipulateur:

En utilisant l'équation (1.50) on a :

$$\frac{\partial L}{\partial q_i} = \frac{1}{2} \sum_{i,j=1}^n A_{i,j}(q) \dot{q}_j \quad (1.51)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) = \sum_{j=1}^n (A_{i,j}(q) \ddot{q}_j + \dot{A}_{i,j}(q) \dot{q}_j) \quad (1.52)$$

$$\frac{\partial L}{\partial q_i} = \frac{1}{2} \sum_{j=1}^n \frac{\partial A_{i,j}(q)}{\partial q_i} \dot{q}_i \dot{q}_j - \frac{\partial P(q)}{\partial q_i} \quad (1.53)$$

Le terme $\dot{A}_{i,j}(q)$ peut être augmenté en termes de dérivés partiels.

En substituant dans l'équation de Lagrange, les équations de mouvements sont données par

$$\Gamma_i = \sum_{j=1}^n A_{i,j}(q) \ddot{q}_j + \sum_{j=1}^n \dot{A}_{i,j}(q) \dot{q}_j - \frac{1}{2} \sum_{j=1}^n \frac{\partial A_{i,j}(q)}{\partial q_i} \dot{q}_i \dot{q}_j + \frac{\partial P(q)}{\partial q_i} \quad (1.54)$$

Afin de mettre les équations (1.54) sous forme d'un vecteur, on définit la matrice $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, tels que :

$$C_{i,j}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \sum_{k=1}^n \left(\frac{\partial A_{k,j}(\mathbf{q})}{\partial q_i} + \frac{\partial A_{i,k}(\mathbf{q})}{\partial q_j} + \frac{\partial A_{i,j}(\mathbf{q})}{\partial q_k} \right) \dot{q}_k \quad (1.55)$$

D'où :

$$\Gamma = A(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) \quad (1.56)$$

Avec :

Γ : Vecteur des couples appliquées aux articulations.

$A(\mathbf{q})$: Matrice d'inertie du robot.

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

$G(\mathbf{q})$: Vecteur des forces de gravité.

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

$$\Gamma = A(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) + F_e \quad (1.56)$$

Il s'agit d'une équation différentielle du second ordre pour le mouvement du manipulateur en fonction des couples articulaires appliqués [10].

- La matrice $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ vérifie les relations suivantes :

$$C(\mathbf{q}, x)y = C(\mathbf{q}, y)x \quad (1.57)$$

$$\|C(\mathbf{q}, \dot{\mathbf{q}})\| \leq v_0 \|\dot{\mathbf{q}}\| \quad (1.58)$$

Avec : $v_0 > 0$ est indépendant de \mathbf{q} .

- La norme du vecteur de gravité est bornée supérieurement :

$$\|G(\mathbf{q})\| \leq g_b(\mathbf{q}) \quad (1.59)$$

Avec : $g_b(\mathbf{q})$ une fonction scalaire.

I.5 Conclusion :

Dans ce chapitre, nous avons identifié quelques notions de base qui nous permettent de mieux comprendre le fonctionnement ainsi que les constitutions d'un robot manipulateur.

La nécessité d'étudier les matrices de transformations homogènes et leurs applications, pour exprimer les situations des différents corps du robot afin de prédire les paramètres de Denavit-Hartenberg, en plus, l'étude des différents modèles dont le modèle géométrique (direct et inverse), le modèle cinématique (direct et inverse) et finalement le modèle dynamique qui est présenté par le formalisme de Lagrange-Euler pour la détermination de l'équation de mouvement du robot.

Chapitre II

Planification trajectoire d'un robot guidée par des champs de potentiel numérique

II.1 Introduction :

Dans le domaine de simulation des systèmes dynamiques, une trajectoire quelconque peut être définie par une grande variété de courbes qui sont généralement représentées par des composées de primitives. La majorité des travaux de planification de mouvement utilisent la notion de l'espace des configurations introduite au début des années 80 par Lozano-Pérez. La planification de trajectoire dépend généralement de la nature de l'environnement, de la nature de tâche effectuée par le robot ainsi que sa structure générale.

Dans le chapitre précédent, nous avons décrit les différents modèles représentatifs pour les systèmes des robots manipulateurs. Etant donné un modèle de l'environnement, il est difficile d'explorer et de représenter explicitement l'espace de configuration, cependant, une alternative consiste à développer des algorithmes de recherches qu'explorent progressivement l'espace de travail tout en cherchant un chemin optimal.

Dans ce chapitre, nous allons présenter une première approche qui est largement utilisée dans la littérature, c'est une approche basée sur la diffusion du potentiel sur l'environnement à partir du potentiel de la position cible ainsi que le calcul optimal de la trajectoire par la descente du gradient.

II.2 Revue de la littérature :

II.2.1 Formulation de problèmes de planification :

Le problème de planification de trajectoire, dans sa formulation générale, revient à déterminer une séquence d'actions permettant au robot de se déplacer entre deux configurations données tout en respectant un certain nombre de critères tels que la distance des obstacles, l'optimisation des contraintes et le temps[11]. Les calculs à effectuer pour l'automatisation doivent donc être effectués en temps réel.

Pour effectuer ces opérations, il est nécessaire de définir un modèle de l'environnement. Pour ce faire, un système de vision 3D permet d'obtenir un modèle de l'espace de travail qui est utilisé lors de l'exécution des tâches [13]. Le système de planification de trajectoire doit donc être compatible avec le modèle de l'environnement fourni par le système de vision 3D. Celui-ci est disponible sous deux formes, soit l'octree [14] [15] et la grille d'occupation. L'environnement pouvant comporter des obstacles quelconques, la description de la scène a avantage à être générale. Dans un contexte plus général, le système doit effectuer la planification de trajectoire d'un robot redondant dans un environnement encombré d'une configuration de départ à une position (et orientation) terminal

demandée, en évitant toute collision. Dans ce contexte, il est intéressant de généraliser la stratégie à un robot manipulateur sériel quelconque [12]. La planification des trajectoires de tels robots nécessitera de pouvoir gérer de nombreux degrés de liberté (*d.d.l*), en particulier si une coopération entre eux est nécessaire. On peut affirmer que, la plupart du temps à chaque instant, le planificateur n'a à se soucier que d'un sous-ensemble de ces *d.d.l*. Mais déterminer quels *d.d.l* sont importants à chaque instant fait également partie de la planification et doit donc être géré par le planificateur [16]. Vu la grande variété de problèmes abordés, la planification de chemin a fait l'objet de nombreuses recherches durant les dernières décennies.

Le développement récent, et de plus en plus important, des capacités de planification rapide peuvent également être extrêmement utiles, dans des environnements virtuels a amené à adapter de nombreuses méthodes pour la programmation hors ligne des robots et pour la génération automatique de scènes animées sur un poste de travail graphique [16]. De nombreux résultats sont discutés dans les états de l'art proposés par Latombe, Choset et LaValle.

II.2.2 Les approches proposées dans la littérature [11]:

La planification de chemin est la formulation la plus simple du problème de planification de mouvement puisque l'environnement dans le quel évolue le robot est statique. Ainsi, le problème se réduit à l'aspect géométrique de la solution et par conséquent toute courbe qui assure les contraintes de continuité et de la non-collision est acceptable comme solution et le défi revient, en général, à calculer le plus court chemin entre la configuration initiale et la configuration finale. Plusieurs méthodes ont été développées et en fonction de la connaissance à priori que le robot dispose de son environnement, deux familles d'approches se présentent:

- **Les approches globales (délibératives ou déterministes):**

Le principe est de déterminer une solution complète du problème si elle existe, avant que le robot débute son déplacement, en se basant sur une connaissance aussi complète que possible de l'environnement de travail.

Parmi ces approches, nous citons : La méthode de décomposition cellulaire (Approchées et exacte), Les méthodes par Roadmaps (dont : les graphes de visibilité, la méthode des cônes généralisés, les diagrammes de Voronoï ou encore la méthode des silhouettes) et La méthode de résolution de type rétraction... etc.

- **Les approches locales (réactives):**

Dans plusieurs applications, l'environnement n'est pas connu à priori. Dans ces conditions, il n'est pas possible de déterminer un mouvement complet jusqu'au but avant que le robot ne commence à se déplacer. La planification correspond alors à gérer dynamiquement les informations relatives à l'environnement. En effet, le principe est de calculer uniquement le mouvement à appliquer au prochain pas temporel et c'est à partir des données capteurs recueillies par le système robotique à chaque instant. Par conséquent, la représentation de l'environnement est construite au fur et à mesure du déplacement du robot.

Parmi ces approches, nous citons : Méthode par fonction de potentielle, Méthode de diagramme de proximité, La fenêtre dynamique, La logique flou, Les Rapidly Exploring Random Trees (RRT), Méthodes Elastic Bande, Les méthodes heuristiques,... etc.

- **Les approches hybrides :**

Ces dernières sont généralement composées d'un algorithme réactif dont les décisions se font en connaissance d'un chemin déjà planifié par une fonction globale. Parmi les méthodes hybrides les plus citées dans la littérature, nous citons [19][18]: Méthode de Vector Field Histogram*, Méthode de Global Dynamic Windows, Méthode de Global Nearness Diagram,... etc.

II.2.4 Considérations générales :

II.2.4.1 Eléments d'un planificateur [12]:

La revue de la littérature a permis d'identifier les principaux éléments qui composent un planificateur de trajectoire.

- L'acquisition des données initiales :
 - Le modèle des obstacles dans l'espace de travail.
 - Le modèle du manipulateur.
 - * Paramètres Hartenberg-Denavit.
 - * Modèle géométrique compatible avec le modèle des obstacles.
 - La description du manipulateur aux états initial et final.
- Calcul d'un espace de recherche à partir des données initiales (l'espace des configurations du robot, le champ de potentiel, etc.).
- La recherche d'un chemin sans collision. Le chemin crée un lien entre les points de départ et d'arrivée, mais il n'est pas utilisable.

- La recherche d'une trajectoire réalisable par un manipulateur réel. Celle-ci est continue et tient compte de critères tels que le temps, la longueur et la distance aux obstacles le long du parcours. La trajectoire peut être produite directement, provenir d'un post-traitement du chemin établi ou être une combinaison des deux.

II. 2.4.2 Espace de Configuration [18] [20]:

L'espace des configurations généralement noté \mathcal{C}_{Space} d'un système est constitué par l'ensemble des configurations qui lui sont accessibles, et définie sur un sous-ensemble continu de \mathbb{R}^n (avec $n = 2$ ou 3) . Dans cet espace, on trouve l'ensemble de configurations occupé par le terrain sur lequel est placé un ensemble d'obstacle \mathcal{C}_{Obs} à éviter et l'ensemble des configurations admissibles (appelé également espace libre) par le système robotique notée \mathcal{C}_{free} .

L'espace de travail dénoté \mathcal{W}_{Space} du système robotique peut être différent de son espace des configurations. L'espace de travail correspond à l'espace dans lequel se déplace ce système. Il peut être en 3D si ce dernier peut se mouvoir dans l'espace tridimensionnel, ou 2D s'il se déplace sur le sol. L'espace des configurations est généralement de plus grande dimension que l'espace de travail.

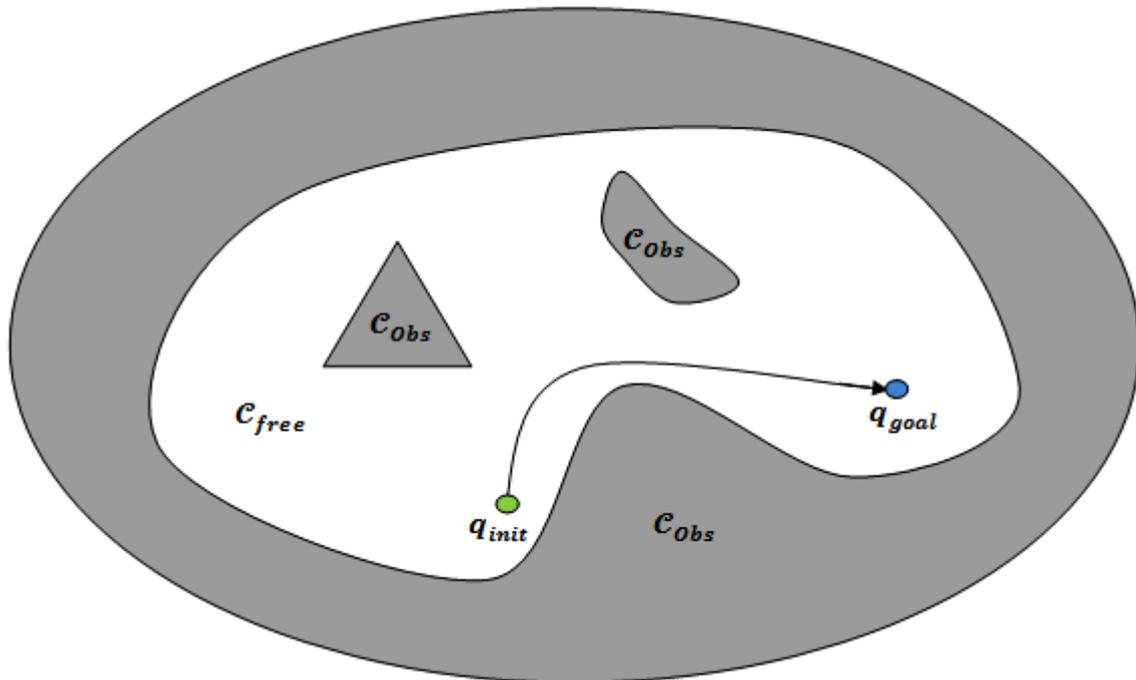


Figure II.1: Illustration d'un espace de configuration 2D [23].

Une des principales méthodes utilisée dans la littérature pour la représentation de terrain consiste à décrire le relief de celui-ci par un modèle numérique d'élévation. Ce modèle consiste en une grille régulière selon les axes (x, y, z) d'un référentiel cartésien \mathcal{F}_c associé à \mathcal{C}_{space} . Tout point de la grille est défini par un triplet (x_i, y_j, z_k) correspondant à sa position dans le plan horizontal et à son élévation par rapport à l'axe « z » de \mathcal{F}_c [23].

II.2.4.3 Contraintes [23][26]:

Les contraintes à considérer pour la planification de mouvements d'un système robotique peuvent être regroupées en deux classes selon qu'elles dépendent de la géométrie de l'environnement et du robot, ou de paramètres directement liés à la génération des déplacements de celui-ci.

- **Les contraintes géométriques :**

Les contraintes géométriques auxquelles est soumis le robot sont:

- La non collision et le non franchissement des zones \mathcal{C}_{Obs} désignées comme obstacles de l'espace de configuration \mathcal{C} .
- L'évitement du renversement du robot. Ceci se traduit par le fait que les angles du roulis et du tangage du robot soient constamment compris dans un intervalle donné.
- La non collision entre les corps du robot autres que les régions pleines de \mathcal{C}

- **Les contraintes de génération des déplacements :**

Le mouvement d'un robot est soumis à des contraintes cinématiques et d'autres contraintes engendrées par la dynamique de la tâche. A première vue, celles-ci proviennent de plusieurs facteurs différents influençant directement le comportement réel du robot: la dynamique du système mécanique de celui-ci, la mécanique du terrain y compris la déformation de certaines zones ou l'existence de composants mobiles placés sur leurs surfaces, et les interactions échangées au niveau des articulations du robot. Les contraintes dynamiques considérées sont liées aux fait que les vitesses, les accélérations linéaires et angulaires du robot et les couples de contrôle à appliquer sur ses articulations doivent être bornés tout au long du mouvement.

II.3 Choix de la méthode :

Avec l'augmentation exponentielle de l'explicité du système robotique, il devient impensable d'utiliser des algorithmes qui demandent un temps de calcul important du principalement aux calculs des images des obstacles dans l'espace des configurations.

En définitive, l'approche proposée est une des premières approche de planification de trajectoire qui tente de combiner les avantages des l'approches locales (éviter les précalculs couteux) et de l'approche globale (rechercher un graphique concis). Cette approche consiste à construire progressivement un graphique reliant les minima locaux d'une fonction potentielle définie sur l'espace de configuration du robot et à rechercher simultanément ce graphique jusqu'à ce qu'une configuration d'objectif soit atteinte. La principale différence, cependant, est qu'il est construit de manière incrémentielle pendant la recherche. Par conséquent, notre approche ne nécessite pas une étape de calcul préalable couteuse. Bien qu'il recherche définitivement un graphique beaucoup plus petit que la grille de discrétisation placée dans l'espace de configuration [16].

II.4 Principe général de la méthode [12][27]:

La planification de trajectoire à l'aide des champs de potentiel artificiel numérique est basée sur un simple et puissant principe, d'abord proposée par *Oussama Khatib*. L'espace de travail et l'espace des configurations sont initialement discrétisés et les obstacles sont modélisés par une grille d'occupation. Les cellules de cette grille sont, ensuite, marquées comme étant en collision ou hors collision selon leurs intersections avec les obstacles. La grille est, alors, utilisée pour calculer un champ de potentiel artificiel à travers les points de discrétisation.

Dans cette approche, Le robot est modélisé par des points de contrôle influencés par un champ de potentiel global généré par le but et les obstacles présents dans l'environnement. L'objectif génère un potentiel attractif tandis que chaque obstacle génère un potentiel répulsif. Un champ de potentiel peut être considéré comme un domaine de l'énergie et donc son gradient, à chaque point, est une force (figure (II.2)). Le robot immergé dans le potentiel déposé est soumis à l'action d'une force qui l'anime à l'objectif (il est l'action de la force d'attraction qui résulte de la pente du potentiel attractif généré par l'objectif) tout en maintenant l'écart des obstacles (c'est l'action d'une force répulsive qui est le gradient du potentiel de répulsion généré par les obstacles). L'idée est alors de demander au robot de se déplacer dans la direction du gradient de potentiel négatif le plus fort, sur le champ de potentiel global obtenu.

Pour éviter des temps de calcul trop longs, on ne calcule le potentiel combiné que pour un sous-ensemble des voisins de la configuration présente choisis de manière aléatoire.

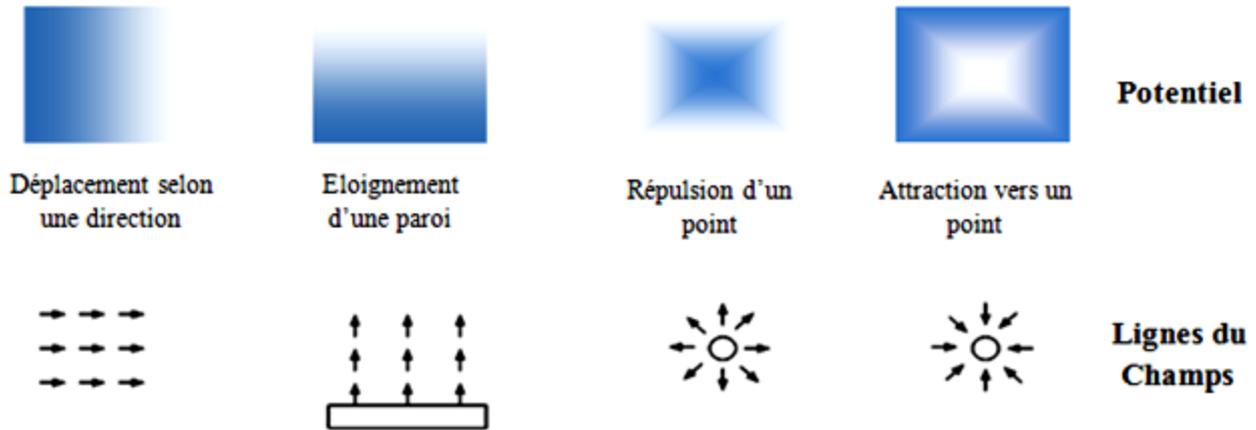


Figure II.2 : Illustration des potentiels primitifs dont la combinaison guide les déplacements du robot. Le robot se déplace selon les lignes de champ.

En titre d'exemple, pour un robot qui se déplace en ligne droite en espace de travail libre et sans collision, nous obtenant les lignes de champ illustrées par la figure (II.3) :

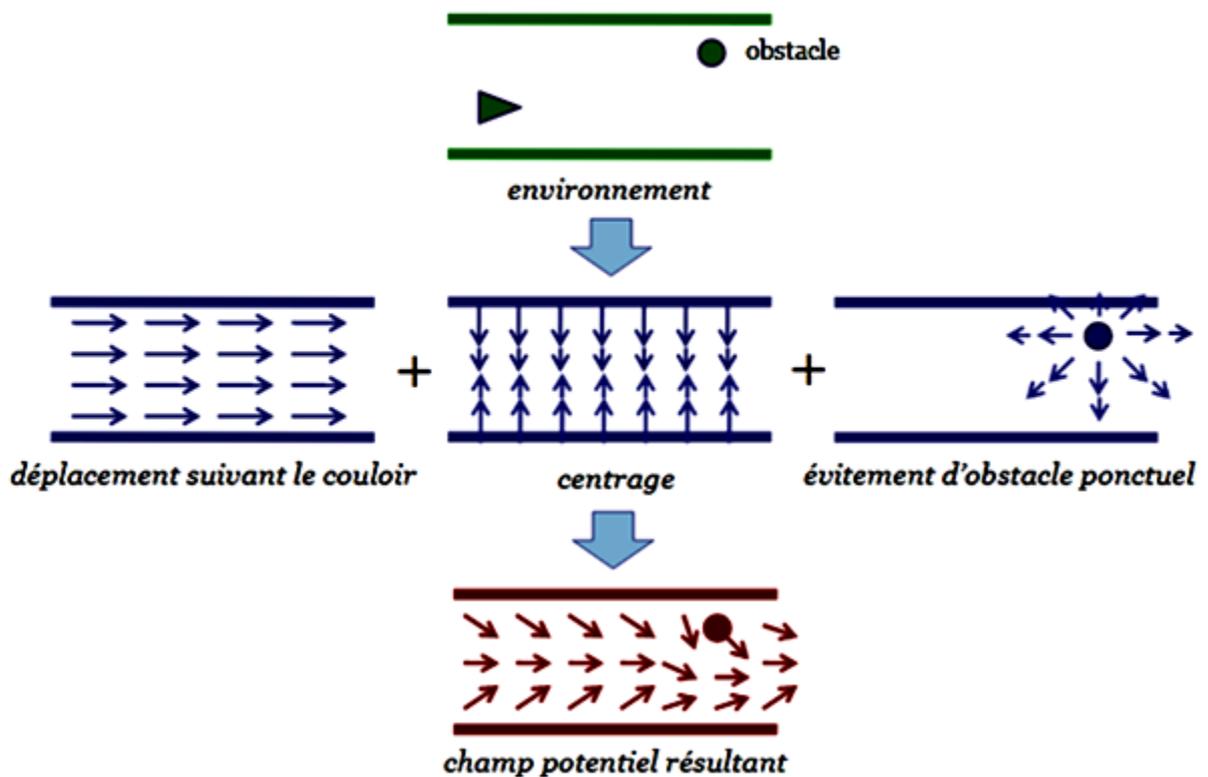


Figure II.3 : Illustration de la combinaison de différents potentiels primitifs.

De plus, Pour s’assurer de l’évitement des obstacles, on modélise le robot par des points de vérification spécifiques localisés par une table de distances aux obstacles. Ceci crée des sphères de sécurité autour de points de vérification. L’ensemble de ces sphères constitue une enveloppe de sécurité le long des segments du robot qui englobe complètement celui-ci, tel qu’illustré à la figure(II.4) :

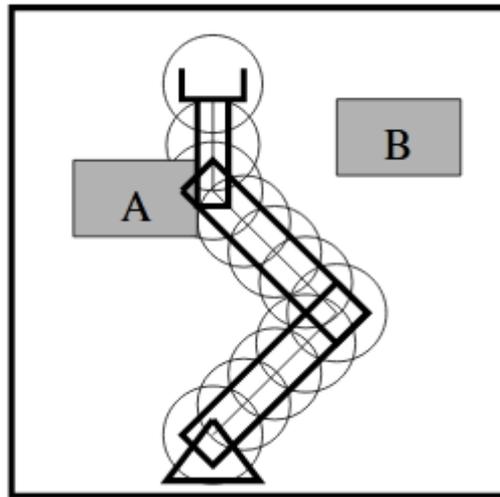


Figure II.4 : Illustration des sphères de sécurité pour la vérification de collision pour un bras manipulateur à 3 *d.d.l.*

II.5 Champ de potentiel numérique [24][25] [30]:

La fonction de potentiel est une fonction à valeur réelle différentiable : $U: \mathbb{R}^m \rightarrow \mathbb{R}$. Dans la planification de trajectoire, la fonction potentielle $U(\mathbf{q})$ est définie comme la somme des deux potentiels Attractif $U_{Att}(\mathbf{q})$, poussant le robot vers la configuration finale et le potentiel Répulsif $U_{Rep}(\mathbf{q})$, tenant le robot éloigné des obstacles.

$$U_t(\mathbf{q}) = U_{Att}(\mathbf{q}) + \sum U_{Rep}(\mathbf{q}) \quad (2.1)$$

La valeur d’une fonction potentielle peut être considérée comme de l’énergie et donc le gradient du potentiel est une force. A chaque itération, la force artificielle $\mathbf{F}(\mathbf{q})$ induite par le champ potentiel, indiquera alors la direction la plus prometteuse. Cependant, le robot suit un chemin « descendant » en suivant le gradient inversé de la fonction potentielle :

$$\mathbf{F}_t(\mathbf{q}) = \mathbf{F}_{Att}(\mathbf{q}) + \sum \mathbf{F}_{Rep}(\mathbf{q}) = -\nabla U_t(\mathbf{q}) = \left(\frac{\partial U}{\partial q_1}, \dots, \frac{\partial U}{\partial q_m} \right)^T \quad (2.2)$$

Cette force est un vecteur qui pointe dans la direction et à chaque q diminue localement au maximum $U(q)$. Elle peut être considérée comme le vecteur de la vitesse qui entraîne le robot. Mais, il est possible que cette résultante s'annule dans certains endroits lors de l'évolution. Cette situation est appelée minimum local et se traduit par l'arrêt du robot avant d'atteindre son objectif. La cause est évidemment l'opposition de la force répulsive et de la force attractive (formule (2.2)), avec la même intensité ou bien les obstacles sont légèrement espacés comme l'illustre les figures (II.5):

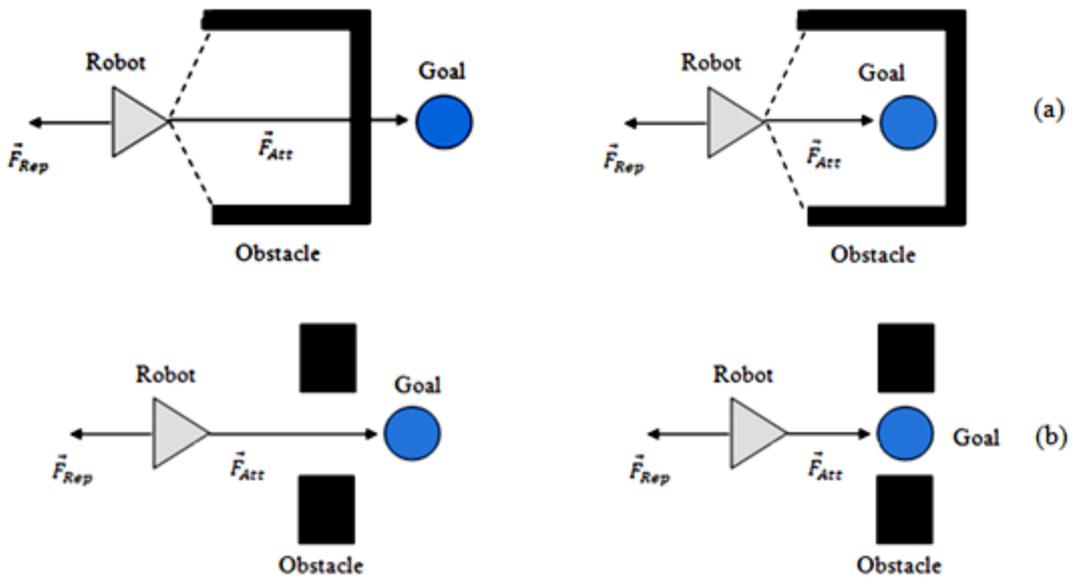


Figure II.5 : (a) : Exemple d'un minima local. (b) : Exemple de problème de rapprochement des obstacles.

En outre, cette situation provoque des oscillations sur la trajectoire du robot. Cela peut être géré en attribuant des critères de distance pour les obstacles qui peuvent indiquer la distance d'attraction.

II.5.1 Champ de potentiel attractif :

Le champ de potentiel U_{Att} devrait satisfaire plusieurs critères. Premièrement, $U_{Att}(q)$ doit être monotone croissant avec la distance de q_{goal} . On distingue deux types de potentiel attractif :

a) Potentiel attracteur de type conique :

Le choix le plus simple est le potentiel conique, mesurant une distance mise à l'échelle au but :

$$U_{Att}(q) = \xi \cdot \rho(q, q_{goal}) \tag{2.3}$$

Ou ξ est un paramètre utilisé pour mettre à l'échelle l'effet du potentiel attractif et $\rho(\mathbf{q}, \mathbf{q}_{goal})$ est la distance euclidienne $\|\mathbf{q} - \mathbf{q}_{goal}\|$.

On aura alors une force de type conique :

$$F_{Att}(q) = -\nabla U_{Att}(q) = -\xi \frac{(q - q_{goal})}{\|q - q_{goal}\|} \quad (2.4)$$

b) Potentiel attracteur de type parabolique :

La fonction potentielle croit de façon quadratique avec la distance à \mathbf{q}_{goal} . Soit :

$$U_{Att}(q) = \frac{1}{2} \xi \cdot \rho^2(q, q_{goal}) \quad (2.5)$$

La fonction $U_{Att}(q)$ est donc positive ou nulle, et atteint son minimum à \mathbf{q}_{goal} c'est-à-dire $U_{Att}(\mathbf{q}_{goal}) = 0$

La force est donc différentiable partout et on a :

$$F_{Att}(q) = -\nabla |U_{Att}(q)| = -\xi \rho(q, q_{goal}) \quad (2.6)$$

L'avantage de la forme conique sur la parabolique, c'est que la force est constante sur l'espace (elle ne tend pas vers l'infini lorsqu'on s'éloigne de \mathbf{q}_{goal} comme la parabolique. Cependant, elle n'est pas nulle à \mathbf{q}_{goal} . La figure ci-dessous illustre les différentes représentations d'un potentiel attractif :

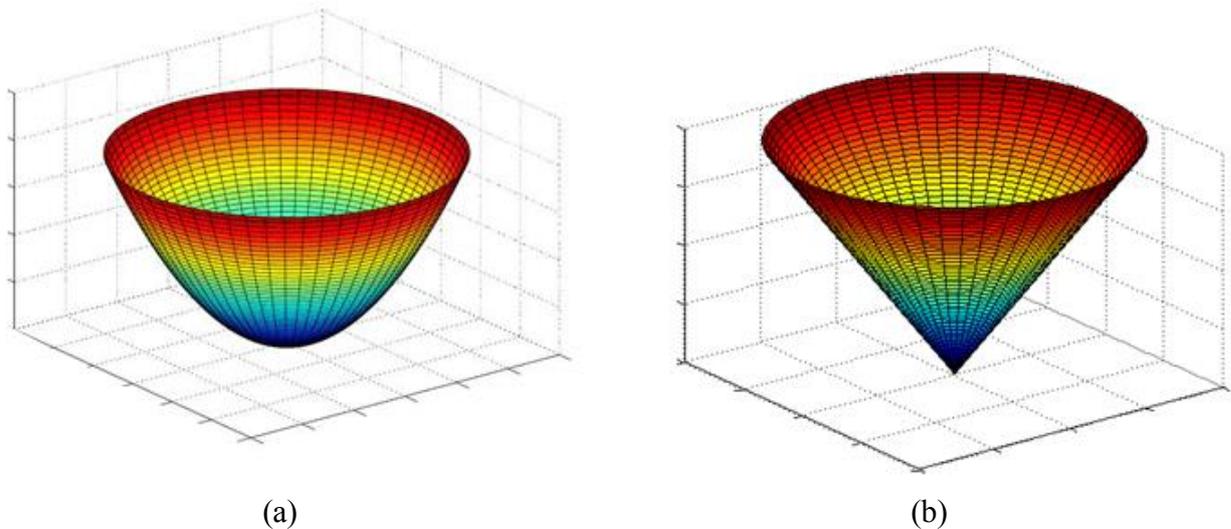


Figure II.6 : (a) Fonction potentiel quadratique. (b) : fonction potentiel conique.

II.5.2 Champ de potentiel répulsif :

Le potentiel répulsif doit servir à créer une barrière de potentiel autour les obstacles (autour de \mathcal{C}_{Obs}) qui ne peut être traversé par le robot (voir figure(II.7)). Par conséquent, le potentiel répulsif est généralement défini en termes de distance par rapport à l'obstacle le plus proche :

$$U_{Rep}(q) = \begin{cases} \frac{1}{2} \eta \left(\frac{1}{\rho(q,q_{o,i})} - \frac{1}{\rho_{0,i}} \right)^2 & \text{si } \rho(q,q_{o,i}) \leq \rho_{0,i} \\ 0 & \text{si } \rho(q,q_{o,i}) > \rho_{0,i} \end{cases} \quad (2.7)$$

Ou :

$\rho(q,q_{o,i})$: La distance minimale de Robot-Obstacles.

$\rho_{0,i}$: La distance d'influence imposée par les obstacles \mathcal{C}_{Obs} .

$\eta > 0$: Le gain de gradient répulsif.

La fonction $U_{Rep}(q)$ est positive ou nulle et tends vers l'infini lorsqu'on se rapproche de la frontière de \mathcal{C}_{Obs} . En outre, la force répulsive dépend généralement de la proximité du robot des obstacles, tel que plus le robot est proche de l'obstacle, plus la force de répulsion est forte. On a alors :

$$F_{Rep}(q) = \begin{cases} \eta \left(\frac{1}{\rho(q,q_o)} - \frac{1}{\rho_o} \right) \frac{1}{\rho^2(q,q_o)} \nabla \rho(q,q_o) & \text{si } \rho(q,q_o) \leq \rho_o \\ 0 & \text{si } \rho(q,q_o) > \rho_o \end{cases} \quad (2.8)$$

Enfin, on peut définir le potentiel répulsif global comme la somme de ces potentiels.

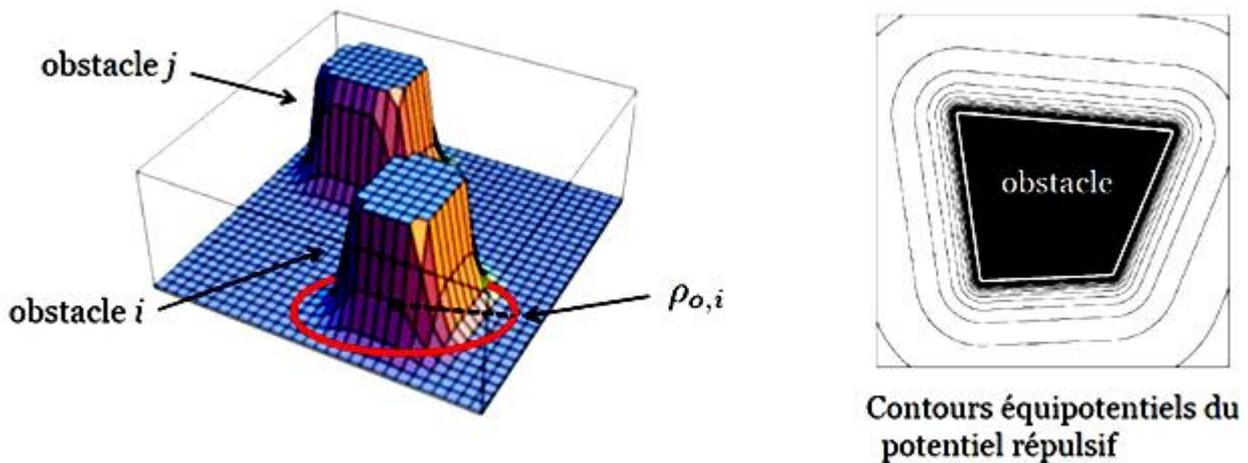


Figure II.7 : Illustration d'une barrière de potentiel autour l'obstacle.

II.5.3 Problème des Minima locaux :

Les problèmes des minima locaux dans les fonctions potentielles font cette méthode incomplète. La résolution de tels problèmes est essentielle pour rendre la méthode champs potentiels adaptés à la planification de trajectoire du robot.

La première idée proposée pour surmonter le problème des minima locaux consiste à créer un graphe reliant les minima locaux aux points de départ et le point de but. Une recherche de plus court chemin peut alors être effectuée sur ce graphe. La méthode nécessite encore la recherche de tous les minima locaux de l'espace ce qui peut augmenter le temps de calcul de cette approche.

Une autre idée est de coupler la méthode du champ potentiel avec une méthode heuristique afin de sortir le robot de ces minima locaux.

II.6 Génération de trajectoire guidée par champ potentiel:

Une fois définis les potentiels, plusieurs techniques sont possibles. Les deux plus courantes sont, pour des espaces de configuration à faibles dimensions ($n < 5$) :

- une descente de gradients
- une recherche du chemin de moindre coût.

II.6.1 Algorithme de Gradient Descente [24][30][31]:

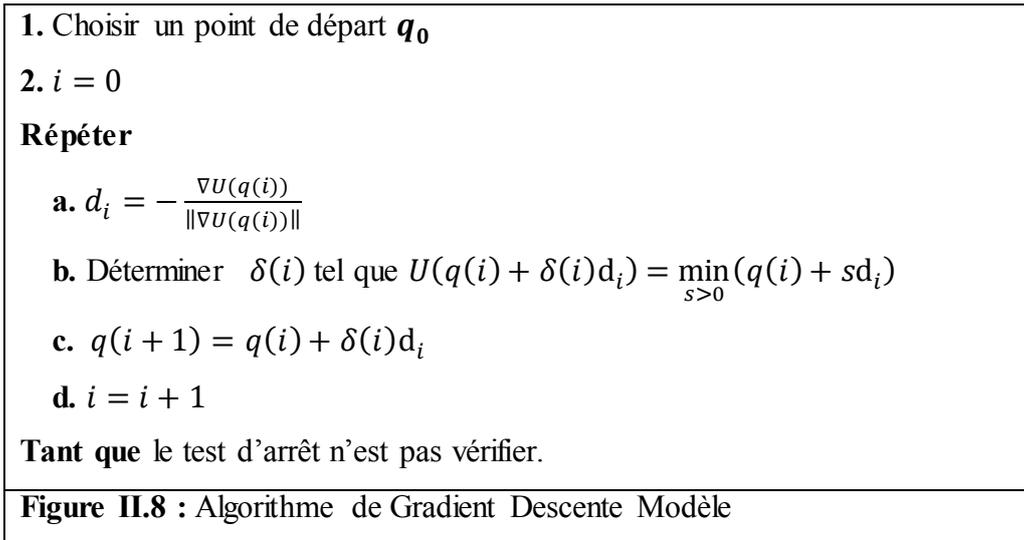
Cet algorithme est à la base des processus d'optimisation. Le terme optimisation est utilisé dans le sens de la minimisation ou la maximisation d'une fonction-objectif. Rappelons que les deux opérations sont équivalentes, ainsi :

$$Arg \max_q F(q) \Leftrightarrow Arg \min_q (-F(q)) \quad (2.9)$$

Le terme descente vient du fait que cette méthode recherche l'extremum suivant une direction \mathbf{d}_i opposée à celle du gradient de la fonction-objectif et à avancer dans cette direction d'un pas optimal de longueur δ_i . Afin d'atteindre le \mathbf{q}_{goal} , on utilise l'algorithme de gradient à partir de :

$$\dot{\mathbf{q}} = -\nabla U_t(\mathbf{q}) \quad (2.10)$$

L'algorithme s'arrêtera dans le cas où un chemin est déterminé. Plus formellement, nous pouvons définir un algorithme de descente de gradient :



On peut exiger que l'un des tests d'arrêt donnés ci-dessous soit vérifié sur plusieurs itérations ou que plusieurs tests soient satisfaits simultanément pour un ε donné:

1. le gradient est très petit : $\sum \left(\frac{\partial U}{\partial q}(q_i)\right)^2 \leq \varepsilon$.
2. La suite q_i est presque stationnaire: $|U(q(i + 1)) - U(q(i))| \leq \varepsilon$.

II.6.2 Recherche du chemin de moindre coût [16][27] :

La recherche du chemin de moindre coût consiste à échantillonner l'espace en une grille fine, puis à appliquer un algorithme de propagation, appelé Best-First Planning (BFP) qu'est utilisé pour désigner spécifiquement une recherche dont l'heuristique essaie de prédire la distance entre le but et la solution, de sorte à explorer en priorité le chemin le plus proche de la solution en un temps de calcul court.

Cet algorithme effectue essentiellement un meilleur BFP en utilisant la fonction potentielle de la configuration comme fonction de coût. Ce type d'algorithme réduit le nombre de nœuds de \mathcal{C}_{Free} en tenant compte des connaissances particulières du problème à résoudre. Le savoir heuristique diminue le temps de recherche et la capacité mémoire nécessaires, par contre ne garantit pas une solution optimale. Parmi les algorithmes les plus utilisés, nous citons l'algorithme de *Dijkstra*, l'algorithme de *A** et l'algorithme *D**.

II.6.3 Vérification de collision :

La vérification de collision est une opération fondamentale lors de planification d'un mouvement, d'animation graphique et de simulation physique [32]. Alors que le contrôle statique revient à tester

une configuration unique des objets pour les chevauchements et le contrôle dynamique consiste à déterminer si toutes les configurations sur une trajectoire continue sont sans collision.

Une technique est proposée pour effectuer la vérification de collision, consiste à calculer une sphère délimitant le volume balayé par chaque lien lorsque les paramètres de configuration varient sur leurs plages complètes [32]. La grosseur des sphères est ajustée selon le robot utilisé et la marge de sécurité voulue ainsi que le type d'articulation existante. Il y a des creux entre les sphères qui causent des variations de la distance de sécurité de l'enveloppe $S(i, k)$. Ces variations obligent à prendre des sphères de rayon $r(i, k)$ plus grand que la distance de sécurité minimale S_{min} , ce qui diminue inutilement l'espace libre du robot. On cherche à minimiser cet effet en gardant l'intervalle L_{max} entre le centre $c(i, k)$ des sphères suffisamment petit [12] :

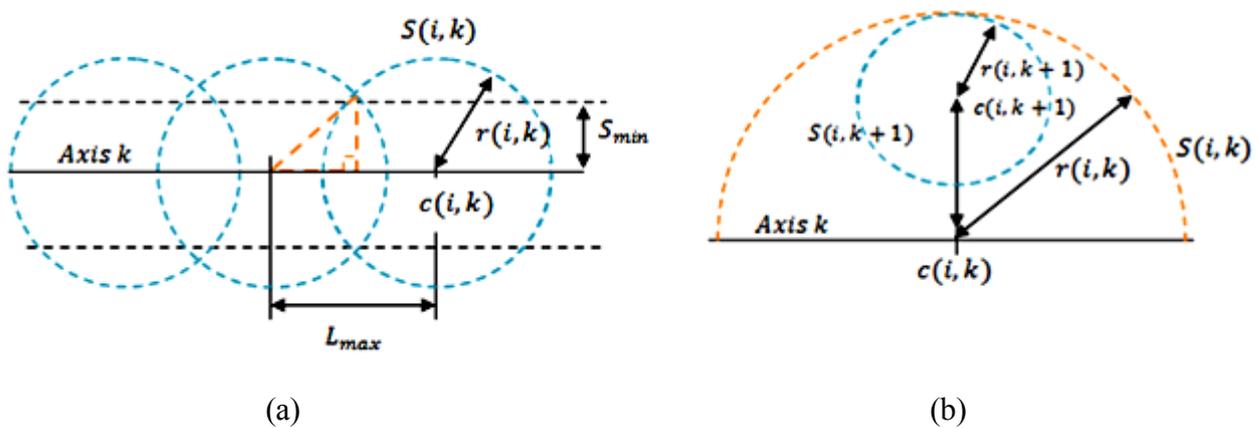


Figure II.9 : Illustrations de la construction de $S(i, k)$. (a) : Cas d'une articulation prismatique.
 (b) : Cas d'une articulation rotoïde.

On peut garantir que le robot n'est pas en collision avec un obstacle si tous les points de vérification sont à une distance aux obstacles plus grande que leur distance minimale respecter, puisque les obstacles se trouvent alors à l'extérieur des sphères de sécurité. Dès qu'un point de vérification se situe à une distance d'un obstacle inférieure à sa distance minimale à respecter, on considère qu'il y a collision, et donc, la fonction de répulsion entre les membres sera activée. On veut ainsi maintenir au minimum les calculs reliés à cette action de répulsion. Cependant, l'effet de cette répulsion mutuelle est inutile lorsque deux sphères sont très distancées. L'algorithme calcule alors les sphères $S(i, k)$ de manière récursive.

II.7 Conclusion :

Dans ce chapitre, nous avons présenté le problème général de la planification de trajectoire en robotique, évoluant dans un environnement connu/inconnu a priori, et nous avons passé en revue les approches les plus utilisées et les méthodes de résolution pertinentes proposées dans la littérature pour le problème de planification de mouvement.

Et puis, nous avons proposé une méthode de planification de mouvement d'un robot avec évitement d'obstacles qui est basée sur la technique des champs de potentiels. Cette approche est une voie simple de technique de planification qui a un principe de fonctionnement intuitif basé sur les champs de type d'énergie. Pour un environnement statique et complètement connu, le potentiel peut être évalué hors ligne, cela nous permet d'obtenir le profil de vitesse à appliquer à un robot quelconque pour se déplacer d'un point de départ à un point d'arrivée par l'utilisation de l'énergie sur le terrain. En outre, cette technique peut être appliquée en temps réel sur une version qui englobe un élément d'évitement d'obstacles. Le potentiel répulsif peut être le résultat de l'environnement des obstacles connus qui sont présentés dans un plan à priori ou des obstacles qui sont détectés par les capteurs du robot lors de sa motion.

Chapitre III

**Planification de chemins pour robots
dans un monde polyédrique en
utilisant le champ de potentiel
numérique avec randomisation de la
recherche**

III.1. Introduction :

Dans ce chapitre, nous traitons le problème de planification de chemins pour robots en utilisant l'approche de champ de potentiel numérique introduite dans [34][35] et étudiée en détail dans [33]. Dans cette dernière référence, ils ont traité les cas de robots ponctuels, polygonaux 2D et le cas de robots manipulateurs dans l'espace 2D. Les résultats qu'ils ont obtenus montrent certains avantages de la méthode. Le premier avantage concerne la rapidité de la méthode pour les cas qui ont un nombre de degré de liberté réduit mais l'inconvénient est que le temps de calcul augmente de façon exponentielle avec le nombre de *d.d.l* et la méthode devient impraticable quand le nombre de *d.d.l* dépasse 4. Le deuxième avantage concerne la complétude, où il a été montré que l'algorithme est complet dans le sens où il arrive toujours à trouver une solution si elle existe et à retourner échec s'il n'y a pas de solution.

Dans ce travail, nous allons introduire une certaine randomisation dans la méthode de recherche utilisée dans l'approche de planification par champ de potentiel artificiel numérique pour palier au problème de l'augmentation exponentiel de temps de calcul avec le nombre de degrés de liberté. Nous allons aussi changer les programmes mis au point pour qu'ils soient applicables au cas de scènes à 3D avec obstacles et segments du robot sous formes polyédriques.

Nous commençons par donner un aperçu sur la procédure de construction du champ de potentiel et sur la méthode de recherche de chemin pour le cas déterministe.

III.2 Construction du champ de potentiel artificiel :

Le champ de potentiel artificiel de l'espace de travail (*WSAPF*) ainsi construit sera utilisé pour construire un champ de 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 supposons que l'espace de travail dénoté \mathcal{W}_{space} est un sous ensemble de \mathbb{R}^n , avec $n = 2$ ou 3 (le plan ou l'espace tridimensionnel). L'espace de travail \mathcal{W}_{space} est modélisé par une pyramide de matrices \mathbf{GW} de dimension $n \in \{2,3\}$. A chaque niveau de résolution, une entrée (x) de la matrice est représentée par une fonction :

$$\begin{aligned} BM : W &\longrightarrow \{0,1\} \\ x &\longmapsto BM(x) \end{aligned} \tag{3-1}$$

tel que le sous ensemble de points pour lesquels $\mathbf{GW}(\mathbf{x}) = 1$ représente le sous espace cartésien des obstacles (\mathcal{W}_{space} obstacles) et le sous espace des points \mathbf{x} tels que $\mathbf{GW}(\mathbf{x}) = 0$ représente la partie libre de l'espace de travail dénotée \mathcal{W}_{free} .

Les points centraux des cellules forment une grille régulière et l'ensemble de point pour lesquels $\mathbf{GW}(\mathbf{x}) = 1$ est appelée \mathbf{GW}_{empty} . Pour $k \in \{1, 2, \dots, n\}$, le k -voisinage d'un point (\mathbf{x}) dans une grille de dimension n est défini comme étant l'ensemble des points dans la grille ayant au plus k coordonnées différentes de ceux de (\mathbf{x}). Dans \mathbf{GW} , nous utilisons toujours le voisinage 1 sauf si c'est spécifié autrement.

Pour le cas d'espace de travail de dimension 3, le niveau de résolution le plus grossier que nous considérons dans ce travail est $(16 \times 16 \times 16)$ et le niveau le plus fin est $(512 \times 512 \times 512)$. 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.

Nous discrétisons de la même façon l'espace de configuration 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 \mathbf{GC} la grille représentant l'espace de configurations et par \mathbf{GC}_{free} le sous ensemble de la grille situé hors collision.

III.2.1. Calcul du champ de potentiel artificiel :

Le calcul du champ de potentiel artificiel dans l'espace de configuration peut se faire généralement en deux étapes principales. Au début, des champs de potentiels artificiels sont calculés dans l'espace de travail pour un ensemble 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. Les Champs de potentiels de l'espace de travail sont, ensuite, combinés dans une autre fonction appelée champ de potentiel de l'espace de configuration qui est définie 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 :

$$V_{p_i}: x \in W_{empty} \mapsto V_{p_i}(x) \in \mathbb{R} \quad (3-2)$$

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

$$U(q) = G(V_{p_1}(X(p_1, q)), \dots, V_{p_s}(X(p_s, q))) \quad (3-3)$$

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 bon champ de potentiel de l'espace de configuration dont la dimension est généralement grande. En effet, les champs de 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 de 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 (*WSAPF*) ne sont calculés que pour un nombre fini (d'habitude petit) 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 entre 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. Différentes fonctions d'arbitrage sont possibles et conduisent à des minimums locaux plus ou moins nombreux ou plus ou moins profonds.

Les seuls calculs préliminaires dans cette approche de planification, sont les (*WSAPF*). Le (*CSAPF*) est généré par la fonction d'arbitrage et peut être calculé selon le besoin durant la recherche du chemin. Le calcul des (*WSAPF*) au départ est nécessaire car on veut qu'ils n'aient aucun minimum local ce qui fait qu'ils ne peuvent pas être calculés localement. Cependant, vu que

les (*WSAPF*) sont calculés dans le bitmap de l'espace de travail dont la dimension est 2 ou 3, ce calcul est faisable et si l'environnement reste inchangé alors il suffit de le sauvegarder une fois pour toute. Il faut noter que pour les scènes 3D, ce calcul peut prendre un temps important.

Deux approches sont possibles pour le calcul de (*WSAPF*). La première donne un champ appelé *WSAPF simple* et la deuxième donne un champ appelé (*WSAPF*) amélioré.

III.2.1.1. Calcul du champ de potentiel de \mathcal{W}_{Space} simple (*WSAPF simple*) :

Le calcul du (*WSAPF*) simple $V_p(x)$ se fait comme suit : Pour chaque point de contrôle p du robot, on commence par sa position de destination q_{goal} et initialise $V_p(q_{goal}) = 0$. Après, la valeur de V_p est fixée à 1 à tous les voisins du point q_{goal} qui sont hors collision. Ensuite, à 2 aux voisins de ces voisins et ainsi de suite jusqu'à ce que tous les points de \mathcal{GW}_{empty} soient explorés.

Ainsi, à chaque point de $x \in \mathcal{GW}_{empty}$, le (*WSAPF*) résultant n'est autre que la longueur L_1 du chemin de longueur minimale reliant q_{start} à q_{goal} à travers \mathcal{GW}_{empty} . Il n'a aucun autre minimum dans \mathcal{GW}_{empty} mis à part q_{goal} . L'algorithme suivant décrit le calcul de V_p .

1. **Pour** tout $x \in \mathcal{GW}_{empty}$, fixer $V_p(x)$ à ∞
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, fait :
 - a. Initialiser L_{i+1} à l'ensemble vide.
 - b. $\forall x \in L_i$, **Pour** tout voisin y de x dans \mathcal{GW}_{empty} :
 Si $V_p(y) = \infty$, **Alors** $V_p(y) = i + 1$; 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 \mathcal{GW}_{empty} contenant q_{goal} . Ainsi, après que l'algorithme soit exécuté et si la position initiale q_{start} de p est telle que $V_p(q_{start}) = \infty$, on peut directement retourner qu'il n'y a aucun chemin entre q_{start} et q_{goal} . La deuxième étape de l'algorithme peut être facilement modifiée pour accommoder le cas où la destination de p est un *sous ensemble* de \mathcal{GW}_{empty} .

III.2.1.2. Calcul du champ de potentiel de \mathcal{W}_{Space} amélioré (WSAPF amélioré) :

Un inconvénient majeur du (WSAPF) simple est qu'il induit des chemins qui s'approchent des obstacles dans l'espace de travail. Pour augmenter l'espace de manœuvres du robot, on utilise le champ de potentiel artificiel amélioré de l'espace de travail. Ce potentiel artificiel est comme le précédant dans le sens où il n'a aucun minimum local mise à part le minimum (global) à \mathbf{q}_{goal} . Mais, en plus, l'inverse de son gradient attire le point de contrôle vers sa position de destination le long d'un chemin non minimal qui demeure éloigné le plus possible des obstacles. Ce champ de potentiel amélioré est calculé en trois étapes :

Dans la première étape, la distance L_1 discrète dénotée $d_1(x)$ de chaque point $x \in \mathcal{GW}_{empty}$ aux obstacles est calculée et un sous ensemble $S \subset \mathcal{GW}_{empty}$ 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.2). Au début, les points situés aux frontières des obstacles sont identifiés et la valeur de d_1 en ces points est fixée à 0 (les points sur les frontières du bitmap de l'espace de travail sont aussi incluses comme points de frontière). Ensuite, la valeur de d_1 est fixée à 1 à tous les points voisins de ces points de frontière se trouvant dans \mathcal{GW}_{empty} ; et à 2 aux voisins de ces nouveaux points ; etc. L'algorithme termine quand tous les points sont visités.

1. **Pour** tout $x \in \mathcal{GW}_{empty}$, fixer $d_1(x) = \infty$.
2. a. Scanner \mathcal{GW}_{empty} et identifier tout point (x) tel que $BM(x) = 1$ et l'un de ses voisins est dans \mathcal{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 x de L_0 fixer $d_1(x) = 0$.
3. **Pour** $i = 0,1,2... \text{ jusqu'à } L_i$ soit vide

Pour chaque $x \in L_i$; **Pour** chaque voisin y de x dans \mathcal{GW}_{empty}

Si $d_1(y) = \infty$ **Alors** $d_1(y) = i + 1$; et insérer y dans L_{i+1}

Figure III.2 : Algorithme de calcul de la distance L_1 des obstacles.

Le squelette de l'espace de travail \mathcal{S} est extrait pendant le calcul de $d_1(\mathbf{x})$ comme étant l'ensemble de points où les ondes (les listes L_i dans l'algorithme) issues de différents points de frontière de \mathcal{GW}_{empty} se rencontre. Ceci est fait en propageant non seulement les valeurs de d_1 , mais aussi les points dans les frontières de \mathcal{GW}_{empty} qui sont à l'origine de cette propagation. Pour chaque point $\mathbf{x} \in \mathcal{GW}_{empty}$, soit $\mathcal{O}(\mathbf{x})$ un tel point. Il faut noter que $d_1(\mathbf{x})$ est la distance L_1 entre (\mathbf{x}) et $\mathcal{O}(\mathbf{x})$.

Dans la deuxième étape, on fixe $\mathcal{O}(\mathbf{x}) = \mathbf{x} \forall \mathbf{x} \in L_0$.

Dans la troisième étape, soit (\mathbf{x}) le point en cours de traitement dans L_i et (\mathbf{y}) l'un de ses voisins. Si $d_1(\mathbf{y}) = \infty$, on fixe $\mathcal{O}(\mathbf{y}) = \mathcal{O}(\mathbf{x})$; sinon, si la distance L_1 entre $\mathcal{O}(\mathbf{y})$ et $\mathcal{O}(\mathbf{x})$ est supérieure à un certain seuil (typiquement 2) et $\mathbf{x} \notin \mathcal{S}$, on inclue (\mathbf{y}) dans \mathcal{S} .

Maintenant, le (*WSAPF*) amélioré est calculé tout d'abord sur \mathcal{S} (*squelette de l'espace de travail*). Ceci est effectué comme suit : Au premier lieu, la position de destination \mathbf{q}_{goal} du point de contrôle \mathbf{p} est connectée au squelette \mathcal{S} par un chemin suivant le gradient de d_1 est les positions le long de ce chemin sont incluses dans \mathcal{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 \mathbf{q}_{goal} , en utilisant l'algorithme d'expansion de front d'onde partant de \mathbf{q}_{goal} et guidé par la fonction d_1 comme suit : Le potentiel $\mathbf{0}$ est donné à \mathbf{q}_{goal} et elle est insérée dans une queue \mathcal{Q} dont les éléments sont les point de \mathcal{S} triés par ordre décroissant de la valeur de d_1 (\mathcal{Q} est initialement vide). Ensuite et jusqu'à ce que \mathcal{Q} deviennent vide, le premier élément de \mathcal{Q} (appelé (\mathbf{x})) est enlevé de \mathcal{Q} et chaque n -voisins (\mathbf{y}) de (\mathbf{x}) dont le potentiel n'a pas encore été calculé reçoit une valeur de potentiel égale à $(V_p(\mathbf{y}) = V_p(\mathbf{x}) + 1)$ et il est inséré dans \mathcal{Q} . L'algorithme termine quand \mathcal{Q} devient vide, c'est à dire quand tous les points du squelette augmenté accessibles à partir de \mathbf{q}_{goal} ont reçu une valeur du potentiel.

Le (*WSAPF*) amélioré à tout autre point de \mathcal{GW}_{empty} est calculé en utilisant l'algorithme d'expansion de front d'onde en partant du squelette augmenté de l'espace de travail. Le potentiel aux voisins-1 de chaque élément (\mathbf{x}) de \mathcal{S} est fixé à $V_p(\mathbf{x}) + 1$. Le (*WSAPF*) amélioré aux voisins-1 de ces nouveaux points est ensuite itérativement incrémenté jusqu'à ce que tous les points accessibles à partir de \mathbf{q}_{goal} on été visités. La figure (III.3) représente l'algorithme de calcul de (*WSAPF*) amélioré :

```

1. Initialisation : Pour tout  $x \in \mathbf{GW}_{empty}$ , fixer  $V_p(x) = \infty$ .
2. Connexion de  $x_{goal}$  au squelette de l'espace de travail
  a.  $x = x_{goal}$ ;
     Tant que  $x \notin S$ 
       Inclure  $x$  dans  $S$ ;
       Choisir un voisin  $y$  de  $x$  ayant la plus grande valeur de  $d_1$ ;
        $x = y$ ;
     Fin tant que;
3. Calcul du WSAPF amélioré dans le squelette augmenté de l'espace de travail
  a. fixer :  $V_p(x_{goal}) = 0$ ;  $Q \leftarrow x_{goal}$ ;  $L_0 =$  ensemble vide;
  b. Répété
     Enlever le premier élément ( $x$ ) de  $Q$  et l'insérer à la fin de  $L_0$ ;
     Pour tout voisins- $n$   $y$  de  $x$  dans  $\mathbf{GW}_{empty}$ 
       Si  $V_p(y) == \infty$  Alors  $V_p(y) = V_p(x) + 1$ ; insérer  $y$  dans  $Q$ ;
     Fin pour ;
     Jusqu'à  $Q$  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  $\mathbf{GW}_{empty}$ 
    $i=0$ ;
Répété
  initialiser  $L_{i+1}$  à l'ensemble vide;
  Pour tout point  $x \in L_i$ 
    Pour tout voisins  $y$  de  $x$  in  $\mathbf{GW}_{empty}$ 
      Si  $V_p(y) == \infty$  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  $L_i$  est vide

```

Figure III.3 : Algorithme de calcul du *WSAPF* amélioré.

III.2.1.3. Calcul du champ de potentiel de \mathcal{C}_{Space} (CSAPF) :

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

$$U(q) = G(V_{p_1}(X(p_1, q)), V_{p_2}(X(p_2, q)), \dots, V_{p_s}(X(p_s, q))) \quad (3-4)$$

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 points 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 son configuration de destination si tous les points de contrôle sont à leur point de destination respective. La configuration de destination peut ne pas être unique. Par exemple, la spécification de la position de destination de quelques points de l'organe terminal détermine une région dans l'espace de configuration. Dans tous les cas nous dénotons par \mathcal{C}_{goal} le sous ensemble des configurations de destination. Prendre seulement une configuration unique parmi ce sous ensemble comme l'ensemble de destination rend, généralement, la tâche de planification plus difficile.

Parmi les premiers formes de la fonction d'arbitrage G , la somme simple des (*WSAPF*) :

$$G(V_{p_1}, V_{p_2}, \dots, V_{p_s}) = \sum_{i=1}^{i=s} V_{p_i} \quad (3-5)$$

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.

Le choix de la fonction G est spécialement important car il influence beaucoup le nombre et la profondeur des minimums locaux de U . Le choix de G qui apparemment minimise le nombre de minimums locaux est :

$$G(V_{p_1}, V_{p_2}, \dots, V_{p_s}) = \min_{i=1, \dots, s} (V_{p_i}) \quad (3-6)$$

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*) $\mathbf{U}(\mathbf{q})$ 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 :

$$G(V_{p_1}, V_{p_2}, \dots, V_{p_s}) = \min_{i=1, \dots, s} (V_{p_i}) + \varepsilon * \max_{i=1, \dots, s} (V_{p_i}) \quad (3-7)$$

où ε est un petit nombre (exemple 0.1) qui dépend du robot. Un autre choix de \mathbf{G} est :

$$G(V_{p_1}, V_{p_2}, \dots, V_{p_s}) = \max_{i=1, \dots, s} (V_{p_i}) \quad (3-8)$$

Ce choix favorise le point de contrôle qui est le plus éloigné de sa destination. Il tend à accroître le nombre de compétitions entre les points de contrôle et, par conséquent, le nombre de minimums locaux. Cependant, il peut constituer un bon choix pour les robots de plusieurs degrés de liberté. En effet, le nombre de minimums locaux n'est pas la seule mesure de qualité du (*CSAPF*). Un autre facteur critique est la profondeur de ces minimums locaux. Parfois, il est préférable d'avoir plusieurs minimums locaux de petites profondeurs que d'avoir un seul minimum très profond. Ceci est spécialement vrai pour les robots de plusieurs degrés de liberté, vu que le nombre de configurations contenues dans un puits de minimum local augmente exponentiellement avec le nombre de *d.d.l.* Divers expériences montrent que la dernière fonction d'arbitrage tend à augmenter le nombre de minimums locaux mais réduit leurs volumes.

III.2.2. Planification en utilisant le champ de potentiel artificiel (cas déterministe) :

Les techniques que nous avons considérées ici prennent 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. Ils sortent par un succès dès qu'un chemin a été généré. Ils retournent un échec si après avoir considéré les bitmaps les plus fins, ils n'ont 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 \mathbf{GC}_{free} . La racine de T est la configuration initiale \mathbf{q}_{Start} .

A chaque itération, l'algorithme examine les voisins- k ($k \in [0, n]$) de la feuille de T qui ont 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 \mathcal{GC}_{free} accessible de q_{init} est complètement exploré (échec). Dans nos applications nous avons fixé $k = n$.

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 puits du minimum local jusqu'à ce qu'un point d'inflexion soit atteint. Après quoi, 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.

Cette algorithme est complet selon la résolution, c'est dire qu'il garantie d'atteindre la configuration de destination en un temps fini quand un chemin de solution existe (à la résolution de \mathcal{GC}), ou il retourne échec quand il n'ya pas de solution. Dans la plupart des cas (virtuellement tous les cas raisonnables), un sous ensemble très petit de \mathcal{GC} doit être exploré avant que l'algorithme termine. Mais, il est possible d'avoir des cas qui n'admettent aucune solution pour lesquels l'algorithme doit explorer presque la totalité de \mathcal{GC}_{free} avant d'abandonner.

III.3. Planification de chemins en utilisant le champ de potentiel artificiel et une recherche randomisée :

L'inconvénient majeur de la méthode déterministe exposée dans la section précédente réside dans le fait que son application est limitée aux cas avec nombres de $d.d.l.s$ inférieurs à 4. Ceci est dû essentiellement au fait que le mouvement selon la plus grande descente (gradient) nécessite la considération de tous les k -voisins d'une configuration donnée et de prendre celui qui est hors collision et a la plus petite valeur de ($CSAPF$). Quand le nombre de degrés de liberté est par exemple 3, on aura ($3^3 - 1 = 27$) voisins à examiner ce qui peut se faire assez rapidement. Par contre si le nombre de ddls est 5, on aura ($3^5 - 1 = 242$; près de 60000 pour 7ddls) voisins pour lesquels on doit calculer leurs $CSAPF$ et on doit les tester contre la collision et ceci pour chaque cellule dans un espace qui est très grand (513^5 pour une résolution de $(\frac{1}{512})$). Ce qui fait que la méthode n'est pas faisable pour ce genre de cas.

Dans cette section, nous présentons la méthode qui permet de surmonter ce problème en changeant la technique utilisée pour échapper des minimums locaux de (*CSAPF*). Partant de la configuration initiale, l'algorithme exécute un mouvement selon l'inverse du gradient jusqu'à ce qu'il atteigne un minimum local de (*CSAPF*). Si ce minimum local n'est pas la configuration de destination, il exécute un mouvement aléatoire pour sortir de ce minimum local pendant un temps t . Cette séquence de mouvement (gradient/aléatoire) continue jusqu'à ce que la configuration de destination q_{goal} soit atteinte. Nous donnons dans ce qui suit les détails de l'implémentation de ces deux mouvements.

III.3.1. Mouvement selon l'inverse du gradient :

Le mouvement selon l'inverse du gradient, considère le voisinage maximal (les k -voisins) en connexion avec une technique aléatoire. Pendant ce mouvement, l'algorithme choisi de façon aléatoire selon une distribution uniforme l'un des k -voisins q' de la configuration actuelle q , s'il est hors collision et qu'il a une valeur de (*CSAPF*) inférieure à celle de la configuration actuelle q , alors, le mouvement progresse vers q' qui sera considéré comme le successeur de q le long de la portion du chemin du mouvement du gradient. Par contre si q' est en collision ou qu'il a une valeur $CSAPF(q') \geq CSAPF(q)$, l'algorithme choisi aléatoirement selon une distribution uniforme une autre configuration parmi les k -voisins restant. Ceci continu jusqu'à ce qu'un voisin valide pour le mouvement selon l'inverse du gradient soit trouvé ou que le nombre d'essais dépasse un nombre maximal d'itération itt_{max} (qui peut être choisi selon le nombre de *d.d.l.s* de quelques dizaines à quelques centaines). Dans ce cas la configuration q est considérée comme un nouveau minimum local q_{loc} . L'idée derrière ceci est que si la courbure de la surface $U = CSAPF(q)$ est petite et que q n'est pas un minimum local, alors la probabilité de deviner une configuration q' telle que $CSAPF(q') < CSAPF(q)$ est de 0.5. La probabilité de trouver une telle configuration q' dans 10 essais successifs est de 0.999. Ceci permet de déduire que même si le nombre d'essais augmente avec le nombre des k -voisins il n'a pas à augmenter de façon linéaire.

III.3.2. Mouvement aléatoire :

A partir d'un minimum local qui n'est pas q_{goal} , l'algorithme effectue un mouvement aléatoire. Ce mouvement aléatoire est implémenté sous forme d'une séquence de pas aléatoires discrets. Ceci consiste en la discrétisation du temps en des unités est l'exécution d'un pas dont la projection selon les axes des q_i est soit $+\Delta_i$ ou $-\Delta_i$ pendant chaque unité de temps (Δ_i dénote la distance entre deux points consécutifs le long de l'axe i de la grille des configurations).

Le mouvement dure un certain nombre d'unités de temps t . A chaque unité de temps, le pas de mouvement est indépendant des pas exécutés précédemment. Une façon d'estimer la valeur de t est de considérer le rayon d'attraction du minimum local $A_{R_i}(q_{loc})$. Dans [35] il a été montré que la durée du mouvement Brownien permettant d'échapper un minimum local est donnée :

$$t(q_{loc}) = \max_{i \in [1, n_{dl}]} \left(\frac{A_{R_i}(q_{loc})}{\Delta_i} \right)^2 \quad (3-10)$$

Mais il est difficile d'inférer des propriétés statistiques de A_{R_i} pour tout minimum local. Cependant, en général, on peut considérer que le rayon d'attraction d'un minimum local ne peut dépasser la distance qui provoquera un mouvement du robot plus long que le rayon de l'espace de configuration lui même. Ainsi, si on considère que l'espace de configuration est cubique :

$$t \approx \frac{1}{\delta^2} \quad (3-11)$$

Avec δ la distance entre deux points successifs le long du même axe du bitmap de l'espace de travail (Nous supposons que l'espace de travail est carré ou cubique avec la même résolution le long de chaque axe). Cependant, ce choix suppose que tous les rayons d'attraction sont les mêmes ce qui n'est pas le cas. Au lieu de ceci, on choisit t comme étant une variable aléatoire dont la racine carré obéit à une distribution de Laplace troquée dont l'espérance est $\left(\frac{1}{\delta^2}\right)$. Ainsi la densité de t est :

$$p(t) = \frac{\delta}{2\sqrt{t}} e^{-\delta\sqrt{t}} \quad (3-12)$$

Au fait, la valeur de t n'est qu'une valeur maximale et à chaque instant, le planificateur vérifie si la nouvelle configuration à $CSAPF(q') < CSAPF(q_{loc})$ et ceci est vérifié, il termine le mouvement aléatoire.

III.3.3. Méthode de recherche dans le graphe des minima locaux :

La méthode de recherche adoptée est une variante de l'approche profondeur d'abord. Le planificateur mémorise le chemin construit connectant q_{start} à la configuration actuelle. A chaque minimum local q_{loc} , le planificateur génère de façon itérative un maximum de N_{max} (typiquement $N_{max} = 20$) paires de mouvements gradient/aléatoire comme décrit précédemment. Si l'un de ces mouvements atteint un meilleur minimum q'_{loc} , le planificateur insère les deux nouvelles portions (gradient/aléatoire) entre q_{loc} et q'_{loc} à la fin du chemin planifié jusqu'à maintenant et génère de nouveau N_{max} mouvements en partant de q'_{loc} .

Si aucune des paires de mouvements partant de q_{loc} un meilleur minimum, le planificateur choisi aléatoirement selon une distribution uniforme une configuration dans les portions aléatoires du chemin enregistré entre q_{start} et q_{loc} et recule vers cette configuration à partir de laquelle il continue la recherche par un mouvement de gradient. Vu que la configuration choisie appartient à une portion de mouvement aléatoire, le mouvement de gradient peut atteindre un minimum local qui n'a pas été visité jusqu'à maintenant. Cette procédure permet d'éviter d'enregistrer un graphe de minima locaux.

Si le premier minimum local atteint est une impasse (*dead end*), le planificateur choisit la configuration ayant la plus petite valeur de (*CSAPF*) (sans qu'elle soit inférieure à $CSAPF(q_{loc})$) et reprend la recherche de cette configuration.

Le problème majeur avec cet algorithme est qu'on ne peut pas connaître quand est ce qu'il faut arrêter la recherche si on n'arrive pas à trouver un chemin. Une façon de faire qui n'est pas exacte est de limiter le temps de calcul par une estimation maximale. Une autre façon, consiste à limiter le nombre de recules que l'algorithme fait sans qu'il y ait amélioration du minimum local. Ceci peut constituer une indication qu'il faut changer les paramètres de l'algorithme et reprendre la recherche.

III.4. Résultats de simulation :

Dans cette section, nous présentons les résultats de simulation que nous avons obtenus. Nous commençons par appliquer la méthode déterministe de planification utilisant le champ de potentiel artificiel numérique à un robot plan de 4 *d.d.ls*. La figure (III.4) représente les poses initiale et finale ainsi qu'un ensemble de poses par lesquelles passe le robot le long du chemin obtenu qui contient 252138 cellules. Notons que cette requête de planification a pris plusieurs heures pour être résolue, ce qui montre les difficultés rencontrées en utilisant le planificateur déterministe pour le cas de robots à nombre de *d.d.ls* important.

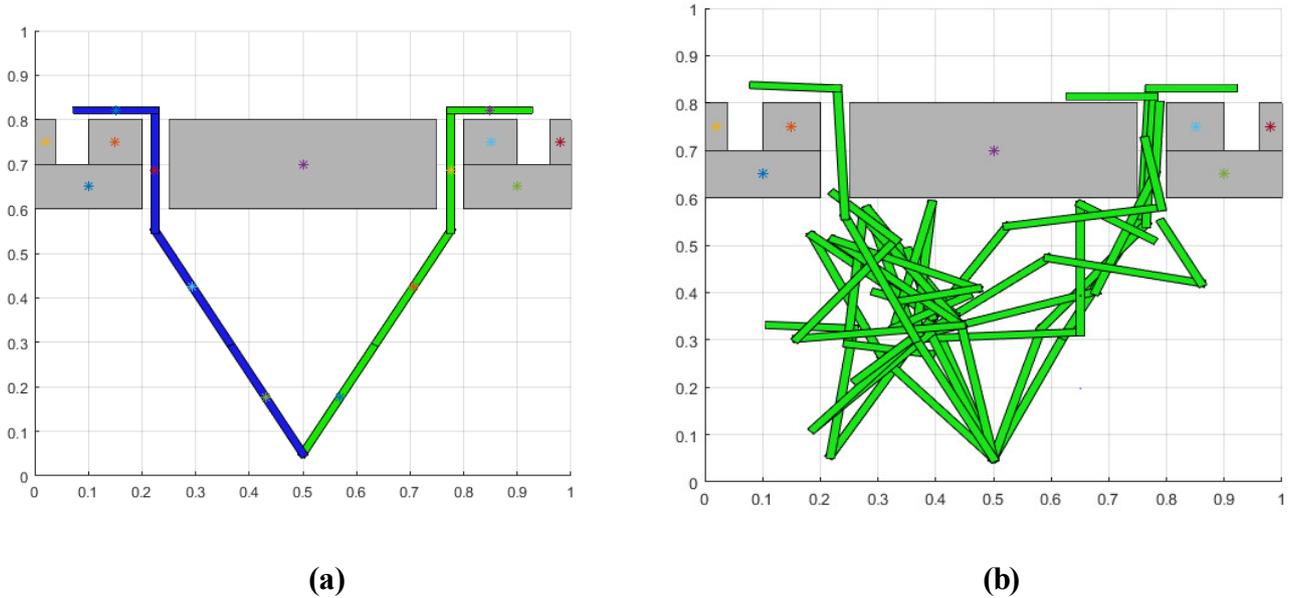


Figure III.4 : Robot plan (4 *d.d.l*) et le chemin retourné par le planificateur à base d'APF déterministe : **(a)** Poses initiale (en vert) et finale (en bleu). **(b)** Quelques Poses intermédiaires.

Dans l'exemple qui suit, nous appliquons le planificateur randomisé à un robot plan de 3 *d.d.l.s*. Nous avons fixé $itt_{max} = 7$ et $N_{max} = 20$. La durée du mouvement aléatoire est choisie selon une distribution de Laplace tronquée. Ce problème est très facile pour le planificateur randomisé surtout quand la configuration initiale n'est pas proche d'un minimum local profond. La seule difficulté représenté par cette requête se trouve près de la configuration de destination.

Nous constatons que la méthode est très sensible à itt_{max} , plus ce paramètre est petit plus la solution devient difficile à trouver et parfois on est obligé de passer à une résolution élevée pour obtenir la solution comme pour le cas de la figure précédente où la solution existe pour une résolution de $\left(\frac{1}{128}\right)$ mais le planificateur a été obligé de passer à la résolution $\left(\frac{1}{256}\right)$ pour la trouver. Il faut aussi se rappeler que itt_{max} n'est qu'une valeur maximale et que le planificateur prend la première configuration avec une valeur de (*UCSAPF*) plus petite que celle de la configuration courante sans faire toutes les itérations permises. Pour le paramètre t (durée du mouvement aléatoire), nous avons utilisé une distribution de Laplace tronquée pour l'obtenir, mais nous avons limité sa valeur par $round\left(\left(\frac{1}{\delta}\right)^2\right)$ pour éviter des valeurs inutilement grandes.

Si t est grand, le planificateur a plus de chance pour échapper les minimums locaux ce qui donne plus de garanti pour obtenir une solution si elle existe mais aussi augmente le temps de calcul. Dans notre application, nous avons constaté qu'une valeur assez grande est nécessaire.

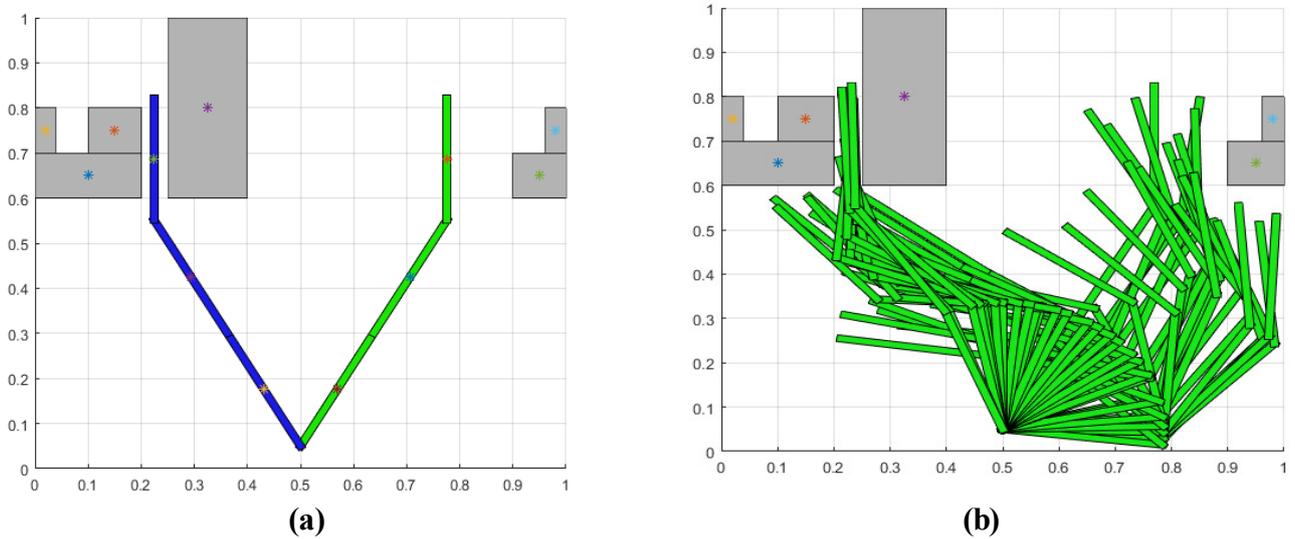


Figure III.5 : Robot plan RRR et le chemin obtenu en utilisant APF amélioré randomisé :

(a) Poses initiale (en vert) et finale (en bleu) **(b)** : Ensemble de poses intermédiaires.

Il faut aussi noter que puisque le planificateur est randomisé, deux exécutions pour un même problème de planification peuvent être différentes. Pour le cas de la figure précédente une solution a put être trouvé plus rapidement sous une résolution de $\left(\frac{1}{128}\right)$ où le robot passe par seulement 714 poses (204 après élagage). Ainsi, plusieurs exécutions doivent être faites pour le même problème pour obtenir une solution meilleure.

Dans l'exemple qui suit, nous avons modifié la scène pour le même robot de l'exemple précédent avec les mêmes paramètres pour la simulation.

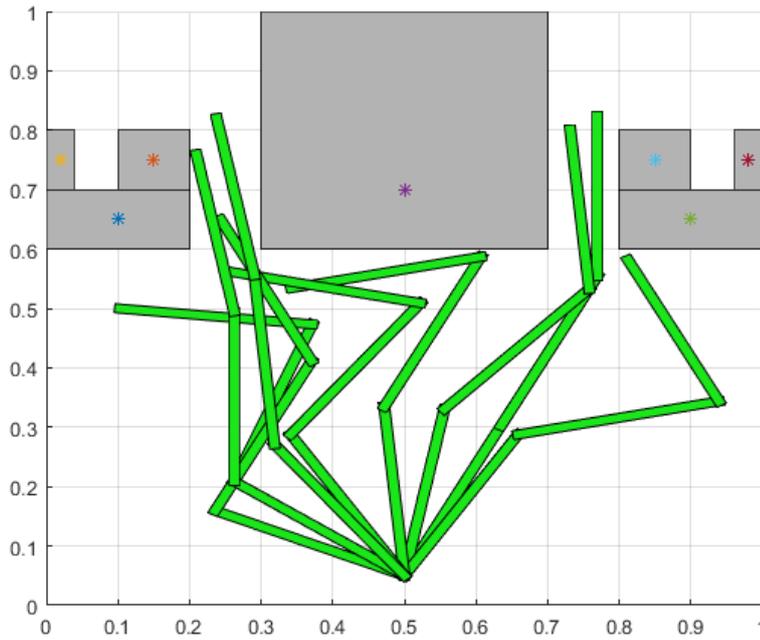


Figure III.6 : Ensemble de poses intermédiaires le long du chemin obtenu en utilisant APF amélioré randomisé après élagage.

Le cas de la figure (III.6) est particulièrement simple pour le planificateur, il arrive à trouver la solution rapidement avec une résolution très réduite de $\left(\frac{1}{64}\right)$. Mais si on rétréci le passage entre les obstacles, le problème devient particulièrement difficile à cause du fait que la région de la configuration initiale est encombrée, ce qui engendre un ou des minimums locaux à partir desquelles il est difficile pour le planificateur de s'en échapper ceci est le genre de problème typiquement difficile à surmonter par les méthodes basées sur les APF. Nous avons aussi traité ce problème (voir figure (III.7)) et le planificateur a due passer à une résolution de $\left(\frac{1}{256}\right)$ pour trouver la solution quoique la méthode déterministe arrive à trouver une solution pour une résolution de $\left(\frac{1}{128}\right)$.

Le mouvement selon l'inverse du gradient se plante trop souvent ce qui fait que le planificateur passe a une résolution supérieure. Nous avons configuré le planificateur pour ignorer ce problème et après un certains nombre d'exécutions, le planificateur arrive à trouver un chemin sous une résolution de $\left(\frac{1}{128}\right)$. Mais pour certaines exécutions, il passe même a une résolution de $\left(\frac{1}{512}\right)$.

Nous ne disposons pas de critère exact pour juger avec certitude quand est ce qu'il faut passer à une résolution supérieure. L'idée que nous avons faite est de travailler beaucoup avec une résolution réduite car même si le temps qui lui est consacré est un temps perdu il sera petit devant celui nécessaire pour les résolutions élevées. Ceci, bien sûr, n'est pas ci-pointu pour le planificateur randomisé. La figure suivante montre le résultat pour ce problème :

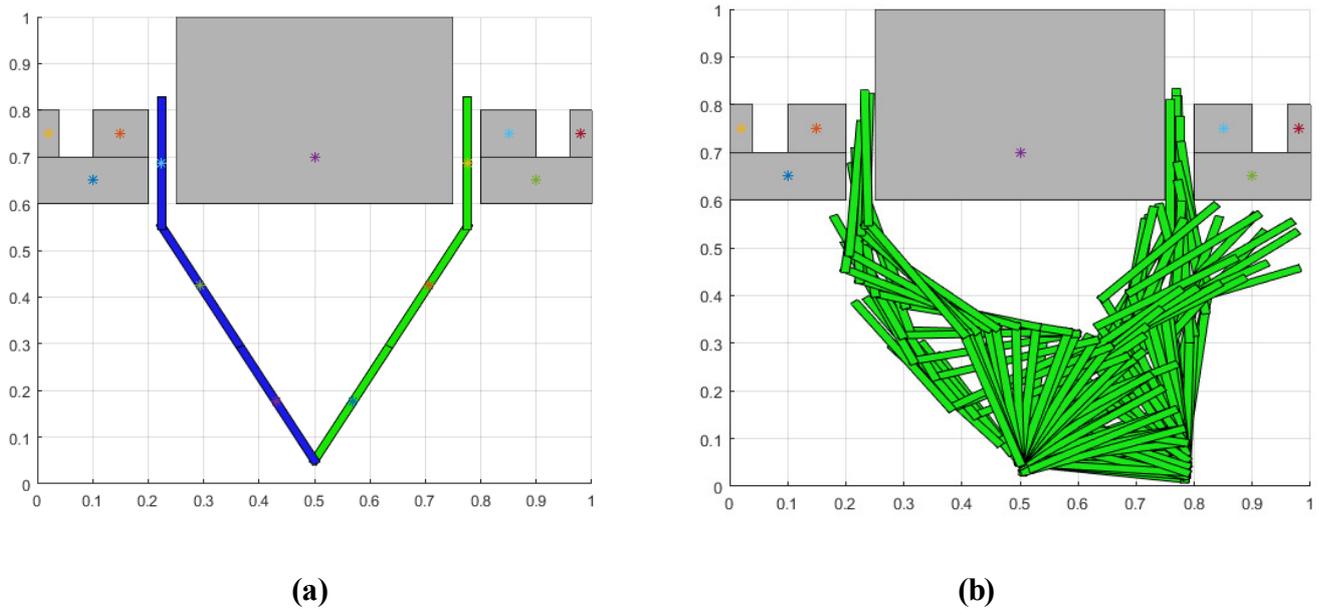


Figure III.7 : Robot plan RRR avec une région de configuration initiale encombrée et le Chemin après élagage : **(a)** Poses initiale (en vert) et finale (en bleu). **(b)** Poses intermédiaires.

Nous avons aussi traité le cas d'un robot à 4 *d.d.ls* et puisque nous savons d'avance à partir du cas déterministe qu'une solution existe sous la résolution $\left(\frac{1}{128}\right)$, nous avons alors paramétré le programme pour n'utiliser que cette résolution et nous avons, effectivement, obtenu la solution que nous donnons dans la figure suivante. Notons que nous avons augmenté $itt_{max} = 20$ car pour ce cas (4 *d.d.ls*) le *k*-voisin contient 81 configurations et prendre seulement 7, parmi ces configurations, fait que le programme traîne beaucoup autour d'un minimum local sans tendre vers la solution.

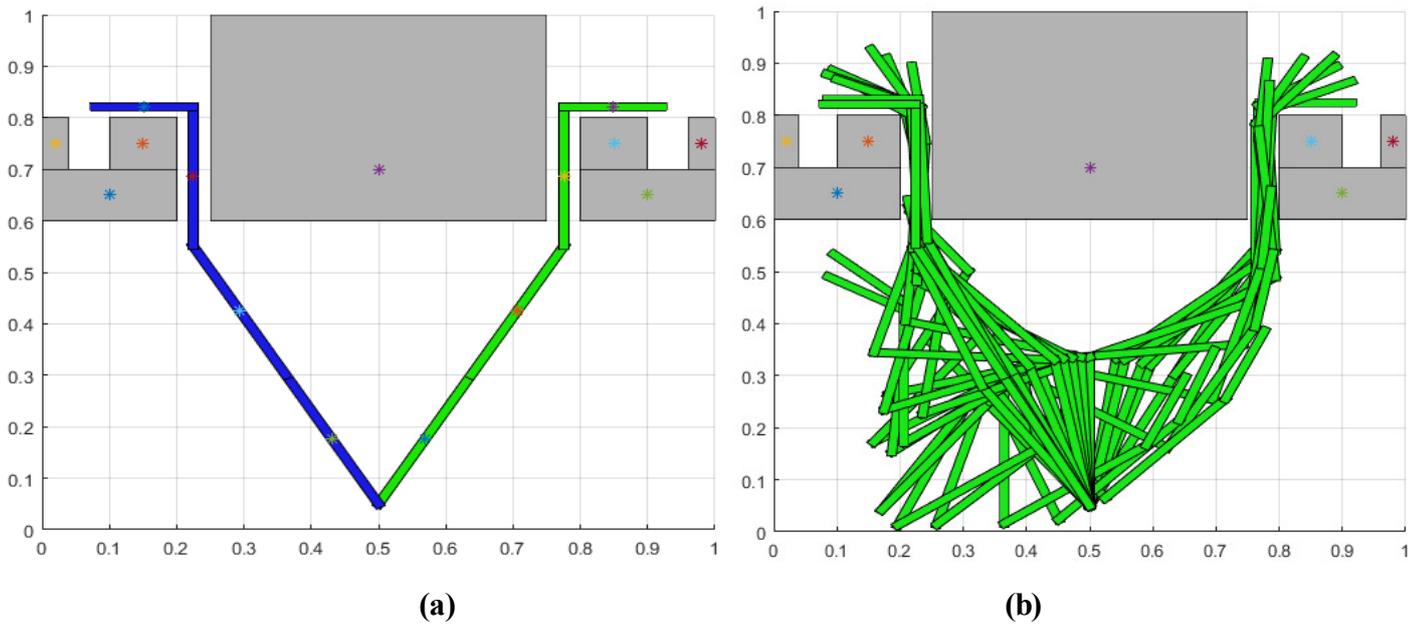


Figure III.8 : Robot plan RRRR (4ddl) et le chemin obtenu en utilisant APF amélioré randomisé après élagage : **(a)** Poses initiale (en vert) et finale (en bleu) **(b)** Poses intermédiaires.

Pour voir l'effet de l'augmentation du degré de liberté sur la rapidité de l'approche de planification par l'utilisation du champ de potentiel artificiel numérique randomisé, nous l'avons appliqué aussi au robot de 5ddl de la figure (III.9). Ce qu'on constate c'est que l'augmentation du temps de calcul avec le nombre des *d.d.l.s* est loin d'être exponentiel et le problème a été résolu rapidement comme les problèmes précédents et avec la plus petite résolution de $\left(\frac{1}{64}\right)$, ce qui prouve la puissance de la méthode. Ce qu'il faut noter aussi est que pour certains problèmes particulièrement difficiles, la méthode échoue dans la recherche d'une solution et elle est incapable de retourner une réponse négative quand la solution d'un problème n'existe pas (impossible). Un autre point reste aussi à renforcer est que le nombre $itt_{max} = 27$ pour le cas du robot à 5ddl était largement suffisant quoique le nombre de k-voisins pour ce cas est 242. Ce qui signifie que le nombre itt_{max} n'a pas à augmenter exponentiellement avec le nombre de degré de liberté du robot et qu'une valeur (même constante) peut être suffisante. Ceci montre bien le fait que le planificateur randomisé est adapté pour les cas à degrés de libertés élevés.

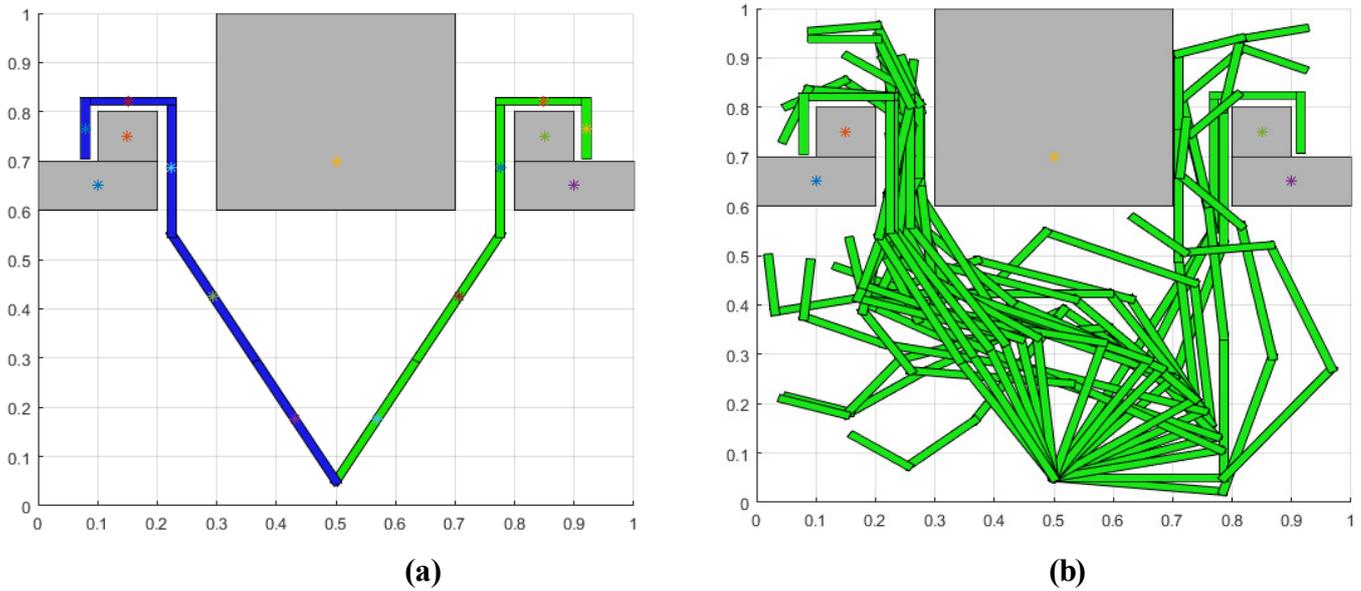


Figure III.9 : Robot plan (5 *d.d.l.s*) et le chemin obtenu en utilisant APF amélioré randomisé après élagage : **(a)** Poses initiale (en vert) et finale (en bleu) **(b)** Poses intermédiaires.

Dans l'exemple suivant, nous considérons le cas d'une scène à 3D avec un robot RRR (3 *d.d.l*). La figure (III.10) donne les poses initiale et finale du robot ainsi qu'un nombre de poses intermédiaires le long du chemin retourné par le planificateur randomisé. Nous avons choisi $itt_{max} = 9$ et la même distribution est utilisée pour le choix de la durée du mouvement aléatoire et le même nombre $N_{max} = 20$. Notons que la difficulté pour le cas de 3D réside dans le calcul du Bitmap pour l'espace de travail car ce calcul prend quelques heures pour être calculé. Mais ceci ne constitue pas un grand problème, car pour le cas où la scène ne change pas entre les requêtes de planification, on peut calculer le Bitmap une seule fois et l'enregistré pour l'utiliser après et son enregistrement ne constitue pas de problème vu que l'espace de travail est de dimension 3 seulement (comparé avec l'espace de configuration qui peut être beaucoup plus grand). L'autre difficulté est dans le calcul du champ de potentiel amélioré qui prend un temps relativement long.

Cependant, la planification elle-même ne montre aucune difficulté particulière malgré que le test de collision soit plus couteux en termes de temps de calcul que pour le cas à 2D.

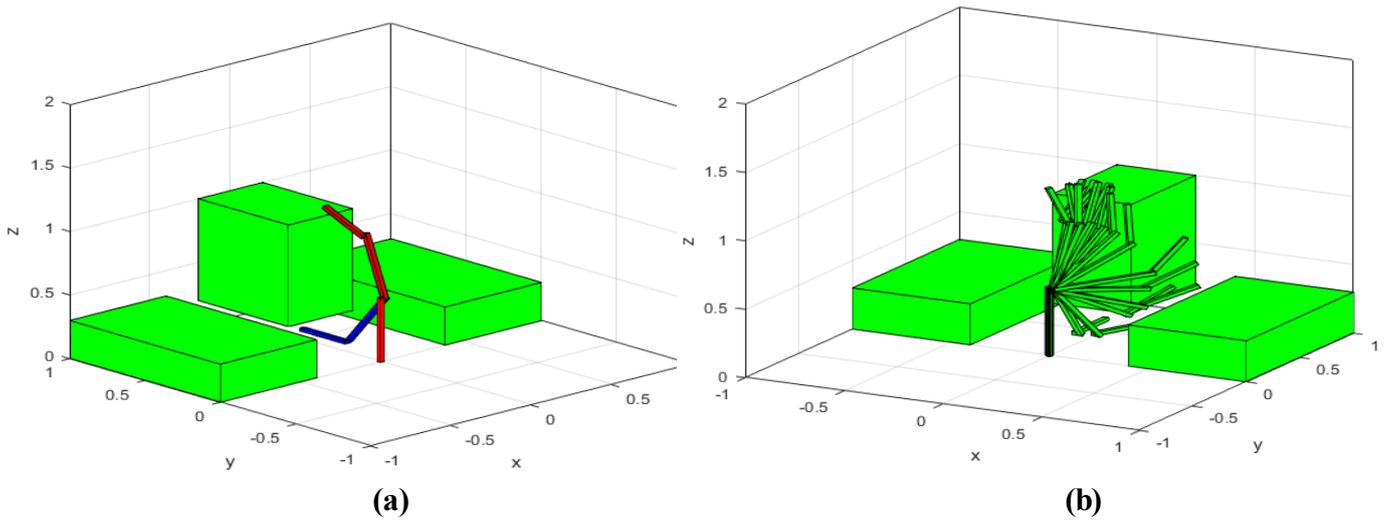


Figure III.10 : Planification de chemin pour le cas d'une scène 3D et un robot RRR avec le planificateur randomisé utilisant le champ de potentiel artificiel numérique.

(a) Poses initiale (en rouge) et finale (en bleu) (b) Poses intermédiaires.

III.5. Conclusion :

Dans ce chapitre nous avons présenté le planificateur déterministe à base de champs de potentiel artificiel numérique ainsi que son implémentation. Nous avons discuté ses avantages et ses inconvénients sur un exemple de planification. Nous avons vu que le temps de calcul nécessaire pour ce planificateur augmente de façon exponentiel avec le nombre de *d.d.l* du robot ce qui limite l'utilisation du planificateur aux cas à nombres de *d.d.l* réduit.

Ensuite, nous avons introduit la version randomisée du planificateur. La stratégie utilisée par la version randomisée permet de limiter le nombre de configurations voisines (d'une configuration donnée) à explorer pendant la phase de recherche par un nombre maximal qui est indépendant du nombre de *d.d.l* du robot. La recherche se fait en utilisant des séquences de mouvement aléatoire/gradient dans laquelle le mouvement selon le gradient n'est qu'une approximation du mouvement selon la plus grande descente. Ces techniques permettent de réduire énormément le temps de calcul et en particulier fait que la dépendance de ce dernier avec le nombre de *d.d.l* n'est pas exponentielle. Le plus grand problème de la technique réside dans la complétude car l'algorithme randomisé ne peut pas reconnaître si une requête de planification n'a pas de solution et on doit le laisser tourner pendant un temps assez important avant d'abandonner sous l'hypothèse qu'il n'y a pas de solution. Pour les cas où une solution existe, l'algorithme réussit généralement à la trouver assez rapidement sauf pour les cas particulièrement difficiles, car on peut toujours construire des requêtes de planification pour lesquelles le planificateur n'arrivera pas à trouver la solution.

Un autre problème consiste au fait que les temps de deux exécutions du planificateur pour la même requête de planification peuvent être très différents ainsi que les deux chemins retournés. Ceci suggère qu'il faut faire plusieurs exécutions pour la même requête de planification et prendre le meilleur résultat.

Conclusion générale

Le travail présenté dans notre mémoire concerne la planification de trajectoire pour robot manipulateur, dans un environnement qui peut être inconnu a priori, est basée sur la méthode des champs de potentiel numérique artificiel. Notre motivation était de rendre ce système capable de percevoir et d'utiliser ses données perceptuelles pour augmenter son autonomie de déplacement, pouvoir agir et réagir face à son environnement et accomplir ses tâches de façon autonome tout en évitant les obstacles imprévus. En particulier, il est nécessaire de modéliser l'environnement ainsi que le robot à partir de ses données perceptuelles, puisqu'elle est à la base des décisions futures sur laquelle reposera le traitement pour établir les déplacements.

Les résultats de simulation pour la méthode basée sur des champs de potentiel numérique artificiel, nous montrent que, le temps de calcul nécessaire pour ce planificateur augmente de façon exponentiel avec le nombre de *d.d.l* du robot ce qui limite l'utilisation du planificateur. Pour cette raison, nous avons introduit la version randomisée du planificateur. La stratégie utilisée par cette dernière permet de limiter le nombre de configurations voisines à explorer pendant la phase de recherche par un nombre maximal qui est indépendant du nombre de *d.d.l* du robot et alors réduire le temps de calcul.

Les plus grands problèmes de la technique résident dans la complétude car l'algorithme randomisé ne peut pas reconnaître si une requête de planification n'a pas de solution. De plus, il est recommandé de faire plusieurs exécutions pour la même requête de planification afin de prendre le meilleur résultat.

Bibliographie :

- [1] : Dombre Etienne. "*Analyse et modélisation des robots manipulateurs*". LAVOISIER, 2001.
- [2] : J.P.Lallemand & S.Zegloul. «*Robotique, Aspects fondamentaux, Modélisation mécanique, CAO robotique, Commande*». 1994.
- [3] : Kerraci Abdelkader. "*Synthèse des commandes robustes des robots manipulateurs rigides*". Thèse de magistère, Université d'Oran, 2004.
- [4] : Alain Pruski. *Robotique générale*. 1998.
- [5] :B.Mendil. "*Modélisation d'un robot manipulateur*". Support de cours, Université de Bejaia, 2019.
- [6] : J.L. Boimond. "*Robotique*". Support de cours, ISTIA, Université d'Angers. 2008.
- [7] : W.Khalil & E.Dombre. "*Base de la modélisation et de la commande des robots manipulateurs de type série*". 2012.
- [8] : J.J. Craig . "*Introduction to robotics, Mechanics and control*". Pearson Education International. 2005.
- [9] : Sciavicco Lorenzo & Siciliano Bruno. "*Modelling and control of robot manipulators*". Springer-Verlag London, 1999.
- [10] : M.Murray Richard, Zexiang Li & S.Shankar Sastry. "*A mathematical introduction to robotic manipulation*". CRC Press, 1994.
- [11] : J. Sawssen. "*Optimisation de la navigation robotique*". Thèse Doctorat, Université de Toulouse, 2016.
- [12]: Thierry Laliberté. "*Planification de trajectoire d'un manipulateur sériel redondant dans un environnement encombré*". Mémoire de maîtrise , Université de Laval, 1994.
- [13]: F. Auclair. "*Modélisation de l'encombrement de l'espace de travail d'un robot à partir de données 3D obtenues d'un capteur*". Mémoire de maîtrise, Université de Laval, 1993.

- [14]: M.Herman . "*Fast Three-Dimensional, Collision-Free Motion planning*". Proceedings of the IEEE International conference on robotics and automation, San Francisco, 1986.
- [15]: E.L. Dupont & S. Derby. "*A simple heuristic path planner for redundant robots*". Proceedings of the ASME Mechanism's conferece, 1988.
- [16]: C.Puech et al. " Motion synthesis, Planning and control, Tutorial ". SIGGRAPH'91, Las Vegas, 1991.
- [17]: A.O. Djekoune. "*Localisation et guidage du robot mobil Atrv2 dans un environnement naturel*".Thèse Doctorat USTHB, 2010.
- [18]: Stéphane Laforet. "*Evaluation de la sureté de techniques de navigation autonomes*". Examen probatoire en informatique, concervatoire national des arts et métiers, 2006.
- [19]: T. Lozano-Perez, J.L. Jones, E. Mazer, & P.A. O'Donnell . "*HANDEY: A Robot Task Planner*". M.I.T. Press, Cambridge, MA, 1992.
- [20]: Simon Kent, "*Evolutionary Approaches to Robot Path Planning* ", Thèse Doctorat Université de Brunel, UB8 3PH, United Kingdom, March 1999.
- [21]: Bernard Bayle, "*Robotique Mobile*", Ecole Nationale Supérieure de Physique de Strasbourg, année 2005-2006.
- [22] : M.Cherif . "*Planification de mouvement pour un robot mobile autonome tout-terrain: Une approche par utilisation des modèles physique*". Thèse Doctorat, Institut national polytechnique de Grenoble, 1995.
- [23]: S.M. Lavalla. " Planning algorithme ". Université Illinois, 2006.
- [24]: Support de cours UE Informatique pour la Robotique. " Planification de trejectoires de robot guidé par champs de potentiel ". UPMC/SSIR 2004-2005.
- [25]: M.GUEZOURI, N. BENATTOU & A. B.DJOUB. "*Planification de trajectoire dans un environnement statique: méthode des champs de potentiels (MCP-S)*". Université d'Oran, 2012.
- [26]: Nicolas Lauzier. "Mécanisme de sécurité pour l'interaction physique humain-robot ". Thèse Doctorat, Université de Laval, Québec, 2011.

- [27]: Ali Mallem. " *Algorithmes d'aide à la navigation d'un robot mobile* ". Mémoire Master en Robotique, Université de Batna, 2011.
- [28]: F.Morbidi. " *Localisation et navigation de robot* ". UFR des Sciences, Département EEA M2 EEAI, parcours ViRob, 2014-2015.
- [20]: Khatib.O. " *Real-time obstacle avoidance for manipulators and mobile robots* ". In Robotics and Automation . Proceedings IEEE International Conference on, volume2, 1985.
- [30] : Howie Choset, Kevin Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia Kavraki & Sebastian Thrun . " *Principels of robot motion: Theory, Algorithms and implementation* ". MIT Press Cambridge, Massachusetts London, England, 2005.
- [31] : S. Kenouche. " *Algorithmes d'optimisation : application à des fonctions objectifs non-linéaires* ". Département des sciences de la matière- UMKB, 2017.
- [32] : Fabian Schwarzer, Mitul Saha ,and Jean-Claude Latombe. " *Adaptive Dynamic Collision Checking for Single and Multiple Articulated Robots in Complex Environments* ". Laboratoire de Robotique, Université de Stanford .
- [33] : Smaoun Ramdhan & Benlatreche Zakaria, " *Planification de trajectoires sans collision d'un robot manipulateur en utilisant de champs de potentiels artificiels*" Mémoire Master en Automatique et Systèmes, Université de Bejaia, 2019.
- [34] : J. Baraquand, B. Langlois & J.C. Latomb, " *Numerical Potential Field Techniques for Robot Path Planning*". IEEE Trans, Man, and Cybernetics, vol 22, n° 2, 1992
- [35] : J. Baraquand & J.C. Latomb, " *A Monte Carlo Algorithm for Path Planning with Many Degrees of Freedom*". IEEE, 1990.

L'objectif de ce travail est de présenter une approche de planification d'un système robotique dans un environnement connu à priori, en utilisant une méthode basée sur l'établissement des camps de potentiel artificiel numérique fictif pour la planification d'une trajectoire dans un environnement statique. L'optimisation d'une trajectoire est obtenue en utilisant une recherche randomisée, afin de palier aux problèmes de l'augmentation exponentiel de temps de calcul et le nombre de degrés de liberté.

The objective of this work is to present an approach for planning a robotic system in an environment known a priori, using a method based on the establishment of fictitious digital artificial potential camps for the planning of a trajectory in a static environment. The optimization of a trajectory is obtained using a randomized search, in order to overcome the problems of the exponential increase in computation time and the number of degrees of freedom.

الهدف من هذا العمل هو تقديم نهج لتخطيط نظام آلي في بيئة
معروفة مسبقاً، باستخدام طريقة تعتمد على إنشاء معسكرات رقمية وهمية
محتملة لتخطيط مسار في بيئة ثابتة. يتم تحسين المسار باستخدام البحث
العشوائي للتغلب على مشاكل الزيادة الأسية في وقت الحساب وعدد درجات
الحرية