

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A. MIRA-BEJAIA

Faculté de Technologie
Département du Génie Electrique



Mémoire de fin d'étude

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

Thème

**Carte de route probabiliste pour la planification de mouvements
d'un corps solide à 3D**

Préparé et présenté par :

ARAOUNE Thiziri.

ZATOUT Souad.

Composition du jury :

Mr. CHEBI Hocine

Président

Mr. HADDAR Hocine

Promoteur

Mr. Mendil Boubekour

Examineur

Année Universitaire : 2017/2018

Remerciement

On tient à exprimer nos vifs remerciements à notre promoteur « Haddar Hocine », pour son entière disposition, et ses judicieux conseils, sa patience et pour avoir proposé ce thème.

Et un remerciement chaleureux a monsieurs Blaid Amine et Tighzerth Lyes pour nous avoir aidé dans notre travail.

Nos remerciements s'adressent également à tous les membres du jury pour l'honneur qu'ils nous font en acceptant de juger notre travail.

Nos remerciements vont aussi à tous ceux et celles qui ont participé de près ou de loin à l'élaboration de ce mémoire.

Enfin, On tient à remercier tous nos amis et collègues pour leur soutien moral tout au long de cette préparation.

Dédicace

Je dédie ce modeste travail à :

Mes chers parents qui n'ont pas cessé de croire en moi et m'encourager et me soutenir dans les moments difficiles et qui ont prié Dieu pour moi, et qui ont su m'entourer de toute leur affection et amour.

Mes sœurs Meriem et Feryel et mon petit frère saadi et a toute la famille.

Ma binôme zatout Souad.

A tous mes amis.

Mes camarades de promotion : Lydia ,aicha, jugo, fafa, youyou, Imad.

Thiziri

Dédicace

Je dédie ce modeste travail à :

*Mes chers parents qui n'ont pas cessé de croire en moi ,
m'encourager et me soutenir dans les moments difficiles et qui ont
prié Dieu pour moi, et qui ont su m'entourer de toute leur affection et
amour.*

Ma sœur Rima

Mes frères Amirouche et Rafik.

Mes belles sœurs Dihia et Loula

Mes neveux Dilane et Ares.

Oualid mon amis de cœur.

Ma binôme Araoune Thiziri.

Manel et Khoukha

Mes copines de chambres mes amis et mes camarades de promotion.

Souad

Table des matières

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

Chapitre I : la robotique en générale

I.1. Introduction :	3
I.2. Historique :	3
I.3. Composant d'un robot :	5
I.4. Les types et domaines des robots :	5
I.4.1. Robot mobile :	5
I.4.2. Robot domestique :	6
I.4.3. Robot industriel :	6
I.4.4. Robots médicaux :	6
I.4.5. Les humanoïdes :	6
I.4.6. Les robots militaires :	7
I.5. Les caractéristiques des robots [6] :	7
I.6. Quelques notions sur la robotique :	7
I.6.1. Degrés de liberté :	7
I.6.2. Articulations :	7
I.6.3. Espace opérationnel :	8
I.6.4. Espace de travail :	8
I.6.5. Espace de configuration :	9
I.7. Orientation et matrices de rotation en trois dimensions :	10
I.8. Mouvement d'un corps rigide :	14
I.9. Modélisation géométrique :	14
I.9.1. Modèle géométrique directe MGD :	14
I.9.2. Modèle géométrique inverse MGI :	15
I.10. Modélisation cinématique directe et inverse [3] :	15
I.11. Modélisation dynamique [14] :	16

Chapitre II: Planification de mouvement et tests de collisions

II.1. Planification de trajectoire et planification du chemin :	18
II.1.1. Planification de trajectoire et planification de mouvement [11] :	19
➤ L'approche découplée :	19
➤ L'approche directe :	19
II.1.2. Résoudre le problème de planification de mouvement :	19
➤ Recherche basée sur la grille :	19

➤ Méthodes de décomposition des cellules :	19
➤ Planificateur de chemins aléatoires (RPP) :	20
➤ Rapidly Exploring Random Trees (RRT):	20
II.2. Test de collision :	21
II.2.1. Les polyèdres :	21
II.2.2 Problème du point dans le polyèdre :	22
II.2.3. Rayon ou segment se croisant contre le polyèdre convexe :	23
II.2.4. Détection de collision entre polyèdres convexes :	24
II.2.4.1. Détection d'intersection vide :	24
➤ Calcul de distance interpolyèdre	24
➤ Approche Gilbert-Johnson-Keerthi (GJK):	24
➤ Approche dobkin et kirkpatrick (DK) :	26
➤ Approche L-C voronoi marching :	26
II.2.4.2. Calcul d'interpénétration :	26
II.2.4.3. Construction de l'intersection :	27
 Chapitre III : PRM pour la planification de mouvement d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques	
III.1. Introduction :	28
III.2. Test de collision :	28
III.2.1. Théorème [20] :	28
III.3. Échantillonnage aléatoire de configurations :	32
III.3.1. Choix aléatoire d'une configuration selon une distribution uniforme :	33
III.4. Connexion de la PRM [10] :	36
III.4.1. Choix des k voisins les plus proches (k-nearest neighbors) :	36
III.4.2. Planificateur local [11] :	38
III.4.2.1. Méthode incrémentale :	38
III.4.2.2. Méthode binaire :	38
III.4.2.3. Rotation à s (rotate-at-s):	39
III.5. Phase de requêtes :	42
III.6. Traitement postérieur [11, 10] :	44
Conclusion :	49
Conclusion générale :	50

Figure (I.1) : Shakey.....	4
Figure (I.2) : PUMA.....	4
Figure (I.3) : le robot Nao.....	5
Figure (I.4) : Composant d'un robot.....	5
Figure (I.5) : robot à roue.....	6
Figure (I.6) : représentation d'une articulation rotoïde.....	8
Figure (I.7) : représentation d'une articulation prismatique.....	8
Figure (I.8) : L'espace de travail, les obstacles et le chemin entre q_{init} et q_{goal}	9
Figure (I.9) : les axes des repères (A) et (B).....	11
Figure (I.10) : les angles d'Euler [4].....	12
Figure (II.1) : Un chemin d'accès est une courbe dans l'espace de configuration libre C_{free} reliant $c(0)$ à $c(1)$	17
Figure (II.2) : Rapidly Exploring Random Trees : Etapes successives de construction de l'arbre de recherche dans l'espace des configurations.....	19
Figure (II.3) : (a) polyèdre convexe. (b) polyèdre concave face, bord, sommet.....	21
Figure (II.4) : L'intersection d'un rayon (ou segment) contre un polyèdre convexe (défini comme l'intersection d'un ensemble de demi-espaces) est l'intersection logique du rayon coupé contre tous demi-espaces.....	22
Figure (II.5) : Représentation de l'Algorithme GJK.....	24
Figure (II.6) : Différence de Minkowski de deux polyèdres convexes.....	24
Figure (II.7) : Illustration de l'approche DK.....	25
Figure (II.8) : Algorithme LC-Voronoi Marching.....	25
Figure (III.1) : Test de collision de deux polyèdres en utilisant la projection sur une droite. On voit que selon la droite perpendiculaire les deux intervalles obtenus par projection chevauchent ce qui ne permet pas de conclure sur la collision. Par contre selon la droite horizontale les deux intervalles obtenus par projection sont complètement disjoints ce qui permet de conclure que les deux polyèdres sont disjoints.....	28
Figure (III.2) : Test de collision en utilisant la projection sur des lignes perpendiculaires à deux arêtes une de chaque polyèdre. Si le test est négatif pour toutes ces combinaisons et pour toutes	

les combinaisons concernant les lignes perpendiculaires à une face de l'un des deux polyèdres alors les deux polyèdres sont en contact (en collision).....	29
Figure(III.3) : Polyèdre balayé pendant un mouvement de translation.....	31
Figure (III.4) : Représentation de 60 configurations hors collision choisies en utilisant une distribution uniforme.....	34
Figure(III.5) : Illustration de la méthode rotate-at-s pour la planification locale.....	38
Figure(III.6) : Cartes obtenues en utilisant deux méthodes de connexion : a) K-nearest neighbors ; b) Méthode de forêts.....	40
Figure(III.7) : Réponse à une requête de planification en utilisant une carte construite en utilisant l'échantillonnage uniforme de configurations la méthode de k-nearest neighbors pour la connexion.....	41
Figure (III.8) : Recherche des chemins entre deux nœuds (1 et 6) dans un graphe.....	42
Figure (III.9) : Extraction d'un arbre à partir d'un graphe contenant des cycles fermés.....	43
Figure (III.10) : Réponse à une requête de planification dans la carte obtenue par la méthode de pont pour le choix de configurations et la méthode de forêt pour la connexion.....	44
Figure (III.11) : Deux chemins parmi 20000 obtenus d'une PRM avec cycles fermés. Le chemin en ligne continue est le chemin le plus courts parmi les 20000.....	45
Figure (III.12) : a) La PRM obtenue en utilisant le test de pont avec échantillonnage uniforme et une connexion avec des cycles fermés.....	48

Introduction générale

Introduction générale :

L'évolution de la science de l'automatique a vu émerger le domaine de la Robotique.

Un robot est une machine équipée de capacités de perception, de décision et d'action, qui lui permettent d'agir de façon autonome dans son environnement.

L'utilisation des robots est présente dans plusieurs domaines : industriels, militaires, chirurgicales, ainsi que dans la vie quotidienne.

Un robot doté de capacité de perception et d'information sur son environnement doit pouvoir se mouvoir de façon autonome, tout en évitant les obstacles. Cette notion d'autonomie du robot représente une composante essentielle qui est la planification de mouvement.

La planification de mouvement consiste à trouver une séquence d'actions valide permettant de relier un état initial à un état final désiré. Cette séquence d'actions peut se définir comme une suite continue de situation géométrique occupée par le robot durant son déplacement appelé trajectoire. Un mouvement valide est un déplacement qui ne produit pas de collision dans un environnement encombré d'obstacles.

Un concept utile pour la planification de mouvement est un espace de configuration. Il est utile de décrire le robot comme une configuration, car il permet de trouver un chemin pour un point dans cet espace, ce qui permet de manipuler différents robots de la même manière.

Une solution rapide et fiable au problème de planification de mouvement est particulièrement importante dans la robotique. Malheureusement, les solutions exactes sont généralement trop lentes pour être utile dans la pratique. Pour surmonter ce problème, de nombreuses approches ont été étudiées et une solution importante consiste à utiliser des méthodes de planification de mouvement basées sur l'échantillonnage. Parmi ces méthodes on trouve les méthodes probabilistes. Leur champ d'application est très diversifiée, elle est utilisée dans la biologie du calcul, et employé dans le prototypage virtuel dans le domaine de la conception assisté par ordinateur...

Dans ce travail, nous avons étudié et programmé, sous MATLAB, les procédures nécessaires pour gérer une carte de route probabiliste d'un environnement à 3D dans lequel le robot et les obstacles ont des formes polyédriques.

Pour une bonne compréhension de notre thème, on a organisé notre mémoire comme suit :

Introduction générale

Dans le premier chapitre, nous présenterons une introduction à la robotique en général.

Dans le deuxième chapitre, nous aborderons le problème de la planification de mouvement et les tests de collisions qui représentent les parties les plus sollicitées du travail.

Dans le troisième chapitre, nous avons traité l'échantillonnage uniforme de configurations, l'échantillonnage basé sur les obstacles et celui qui utilise le test de pont pour l'échantillonnage à l'intérieur des passages étroits, enfin nous avons construit une carte de route. Nous portons les résultats dans notre mémoire.

Chapitre I :

La robotique en générale

Chapitre I : La robotique en générale

I.1. Introduction :

La robotique est la science qui s'intéresse aux robots. C'est un domaine multidisciplinaire qui touche à de nombreux secteurs de la vie, surtout l'industrie.

Le terme robotique a été utilisé pour la première fois en 1942 par l'écrivain de science-fiction Isaac Asimov dans son livre « Runaround » dans lequel il évoque les trois règles de la robotique suivante :

- 1- Un robot ne peut pas blesser un être humain ou, par inaction, permettre à un être humain de nuire, à moins que cela ne viole une des deux autres loi.
- 2- Un robot doit obéir aux ordres qui lui sont donnés par les êtres humains, sauf si ces ordres seraient en conflit avec la première loi.
- 3- Un robot doit protéger sa propre existence, à condition que cette protection n'entre pas en conflit avec une des lois précédentes [1].

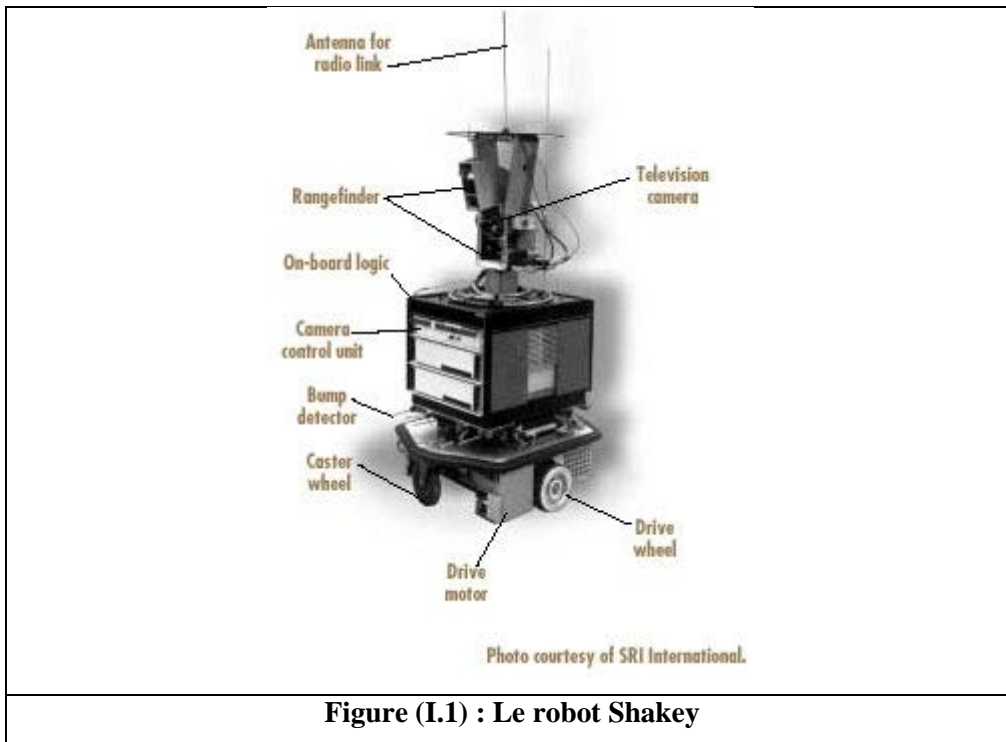
Selon l'organisation internationale de standardisation (ISO), un robot est un manipulateur multifonctions programmable et contrôlé automatiquement qui possède 3 degrés de liberté ou plus. Autrement dit, c'est un agent physique réalisant une variété de tâches complexes.

Pour que le robot réalise des tâches, il est obligé de bouger, alors le mouvement est la problématique commune à tous les robots. D'où l'importance de comprendre la description mathématique du mouvement en termes de déplacement, de vitesse et d'accélération, ce qui sera le fruit d'une bonne modélisation et planification de mouvement.

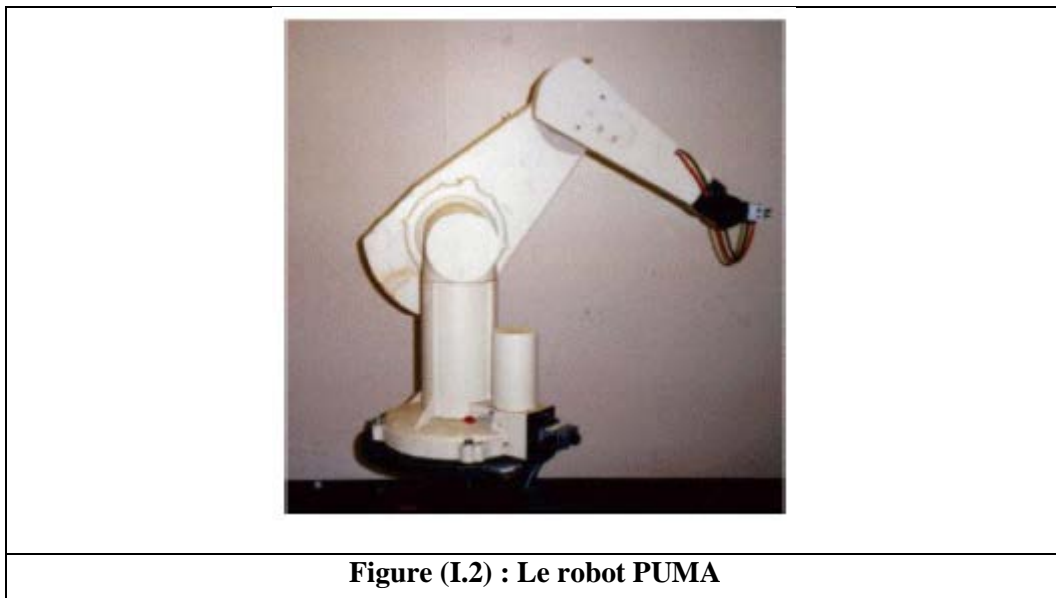
I.2. Historique :

Développement de la robotique [1] :

- En 1940 : Robots de type maître-esclave pour manipuler des charges radio-actives..
- En 1950, le couplage mécanique remplacé par des parties électrique et hydraulique
- En 1955, premier prototype pour une séquence programmée d'action « Programmed Articulated Transfert Device » par George DEVOL.
- En 1961 : Introduction du premier robot industriel capable de réaliser les tâches automatiquement dans une usine de General Motors (UNIMATE).
- Fin des années 60 : mise en place de « Shakey » premier robot mobile intégrant perception, planification et exécution.

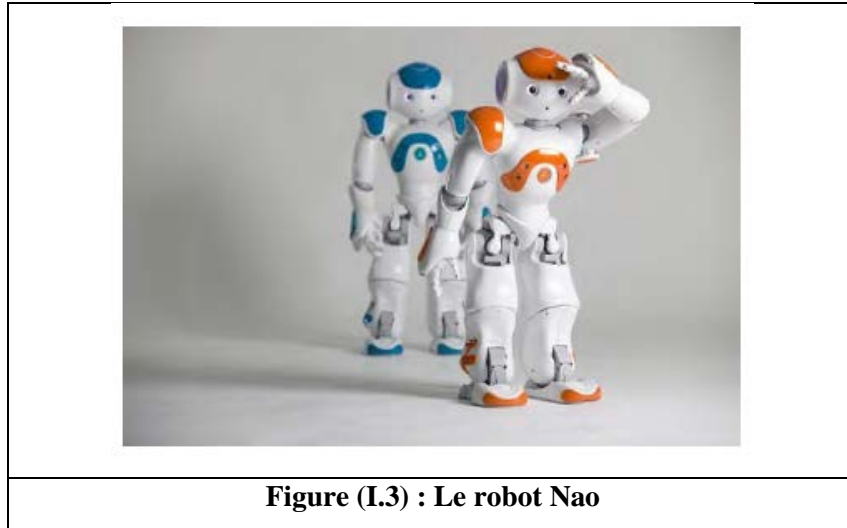


- 1970 : Stanford Cart
- En 1972 : Nissan ouvre la première chaîne de production complètement robotisée.
- En 1978 : PUMA (Programmable Universal Machine for Assembly) développé par General Motors.



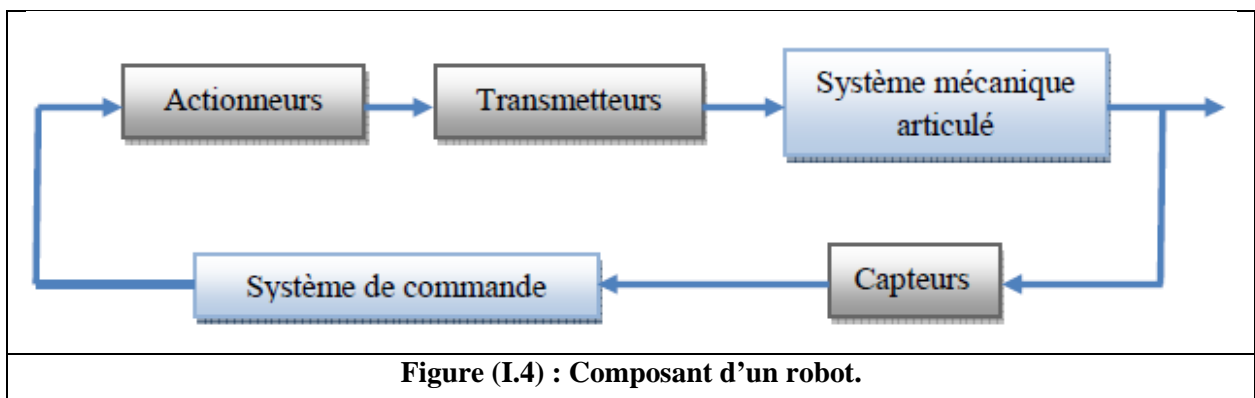
- 1980-1990 : développement important de l'intelligence artificielle et de la robotique

- 1990-2000 : la robotique mobile s'est développée, notamment la mise en place de la RoboCup.
- 2009 : démocratisation des robots, comme le robot Nao utilisé à la RobocupSoccer[2].



I.3. Composant d'un robot :

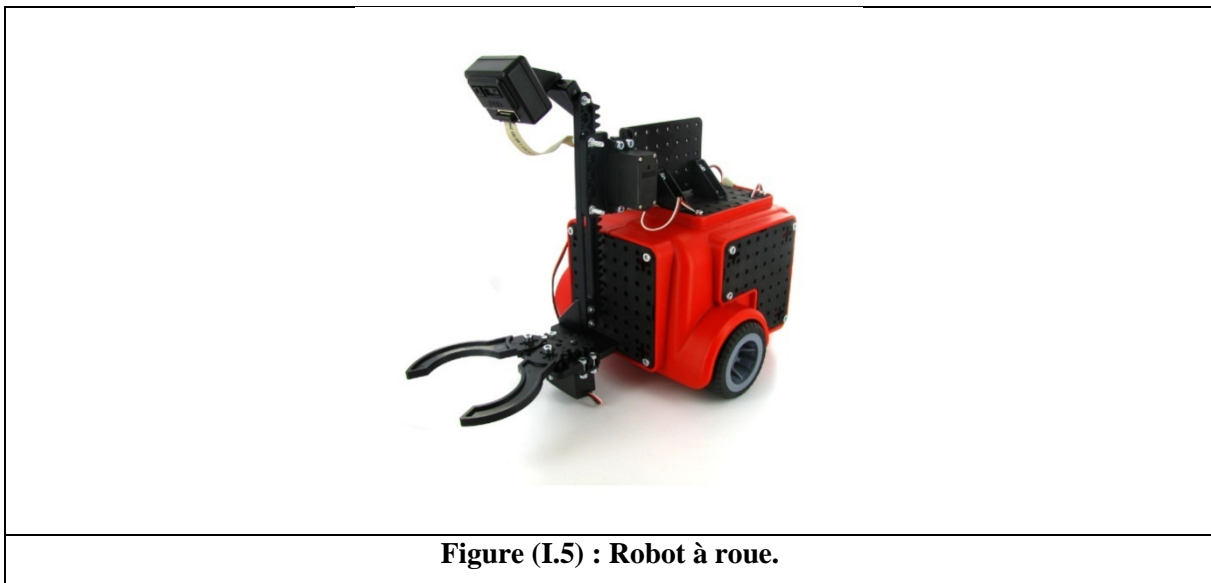
- Le système mécanique articulé exécute les tâches
- L'effort est produit par les actionneurs
- Transfert d'effort et adaptation du mouvement à travers les transmetteurs
- L'état du système articulé et son environnement sont donnés par les capteurs
- Système de commande qui est composé de calculateurs avec algorithmes de commande.



I.4. Les types et domaines des robots :

I.4.1. Robot mobile :

Le principe d'un robot mobile consiste à réaliser un mouvement en reliant un point source à un point destination. L'exécution de cette tâche nécessite des moyens de perception et de raisonnement par rapport à son degré d'autonomie [15]. On distingue quatre types de structure mécanique, dont les robots à roues, à chenilles, marcheurs, et les robots rampants, comme dans (figure(I.5)).



I.4.2. Robot domestique :

C'est des robots utilisés pour des tâches ménagères, par exemple la vaisselle, le repassage, et le nettoyage. Ces derniers parcourent toute une surface d'une pièce en évitant les obstacles réels tels que les meubles, ou des obstacles virtuels spécifiés par l'utilisateur tels que les escaliers.

I.4.3. Robot industriel :

Il est défini par l'ISO comme un contrôle automatique, reprogrammable, polyvalent, d'un manipulateur programmable dans trois ou plusieurs axes.

I.4.4. Robots médicaux :

Le robot médical est utilisé dans le cadre des applications thérapeutiques, il peut aussi servir de cobayes pour les apprentis médecins.

I.4.5. Les humanoïdes :

C'est des robots qui ont généralement une morphologie humaine. Ils évoquent la bipédie, la présence de deux bras et d'une tête. Ils sont capables de faire des choses que seuls les humains peuvent faire.

I.4.6. Les robots militaires :

Un robot militaire est une arme autonome ou contrôlée à distance, conçue pour des applications militaires.

I.5. Les caractéristiques des robots [6] :

La norme ISO 9946 spécifie les caractéristiques que les fabricants de robots doivent fournir. Parmi ces caractéristiques, on peut en mentionner quelques-unes qui permettent de choisir un robot en fonction de l'application envisagée :

- L'espace de travail : il s'agit de l'ensemble des situations de l'espace que l'organe terminal du robot peut atteindre. Il est défini par ses limites, imposées par le nombre de degrés de liberté, les débattements articulaires et par la longueur des segments du manipulateur ;
- La charge utile ou charge maximale transportable par le robot ;
- Les vitesses et accélérations maximales, qui conditionnent les temps de cycle ;
- Les performances (dont les critères sont précisés dans la norme ISO9283)
- La résolution : c'est la plus petite modification de la configuration du robot à la fois observable et contrôlable par le système de commande.

D'autres caractéristiques doivent être prises en compte : techniques (énergie, commande, programmation...) et commerciales (cout, maintenances...).

I.6. Quelques notions sur la robotique :

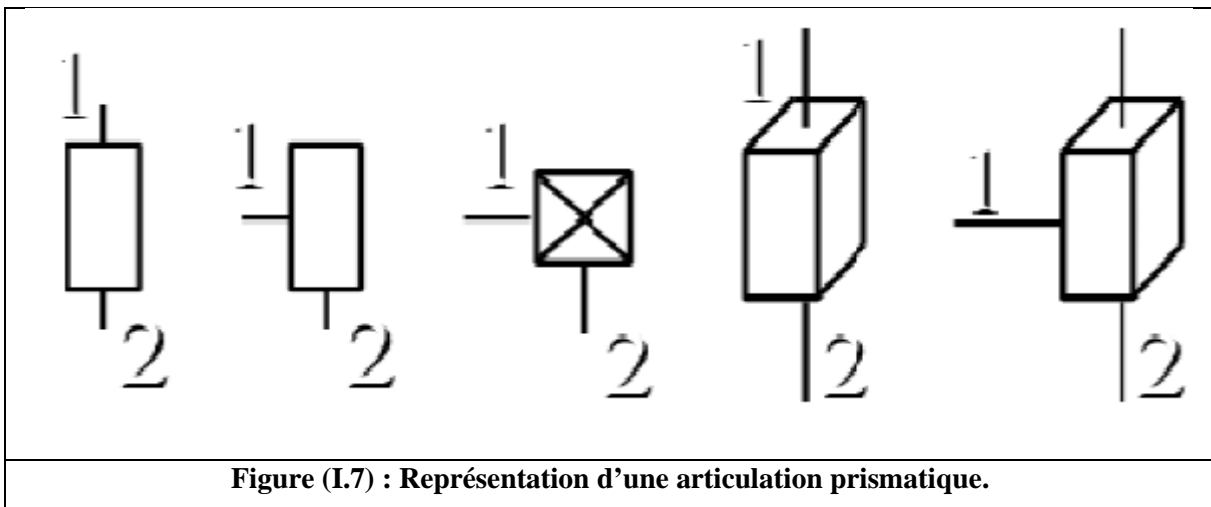
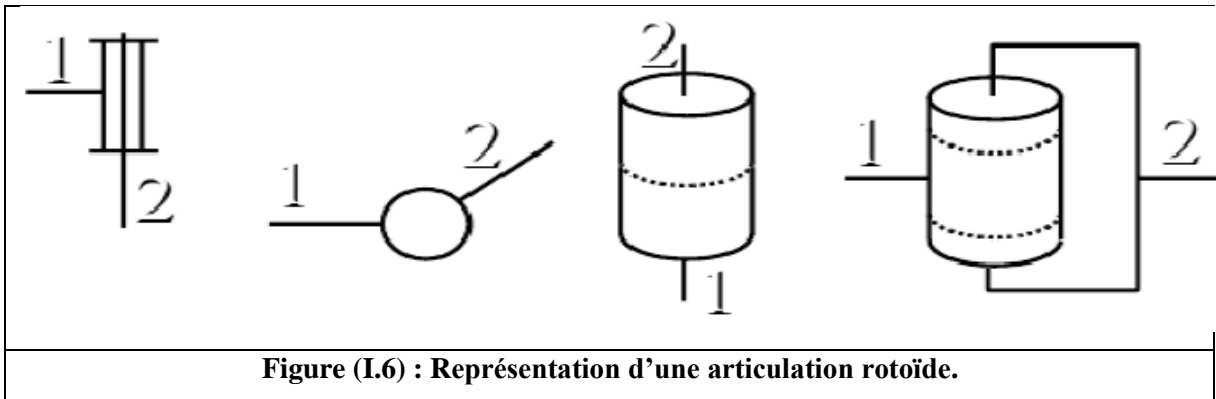
I.6.1. Degrés de liberté :

Le degré de liberté d'un robot manipulateur est égal au nombre de paramètres indépendants qui fixent la situation de l'organe terminal, il peut être en fonction de la configuration du robot.

I.6.2. Articulations :

Une articulation lie deux corps successifs en limitant le nombre de degrés de liberté (dl) de l'un par rapport à l'autre. Soit m le nombre de degrés de liberté résultant, encore appelé *mobilité de l'articulation*. La mobilité est telle que $0 < m < 6$.

Lorsque $m = 1$, ce qui est le cas le plus fréquent en robotique, l'articulation est soit rotoïde noté R (**figure (I.6)**), ou prismatique noté P (**figure (I.7)**) :



I.6.3. Espace opérationnel :

C'est celui dans lequel est représentée la situation de l'organe terminal. Sa dimension est égale au nombre de paramètres indépendants nécessaires pour décrire la situation de l'organe terminal dans l'espace. Dans l'espace tridimensionnel, ce nombre est de six (trois pour la position, et trois pour l'orientation).

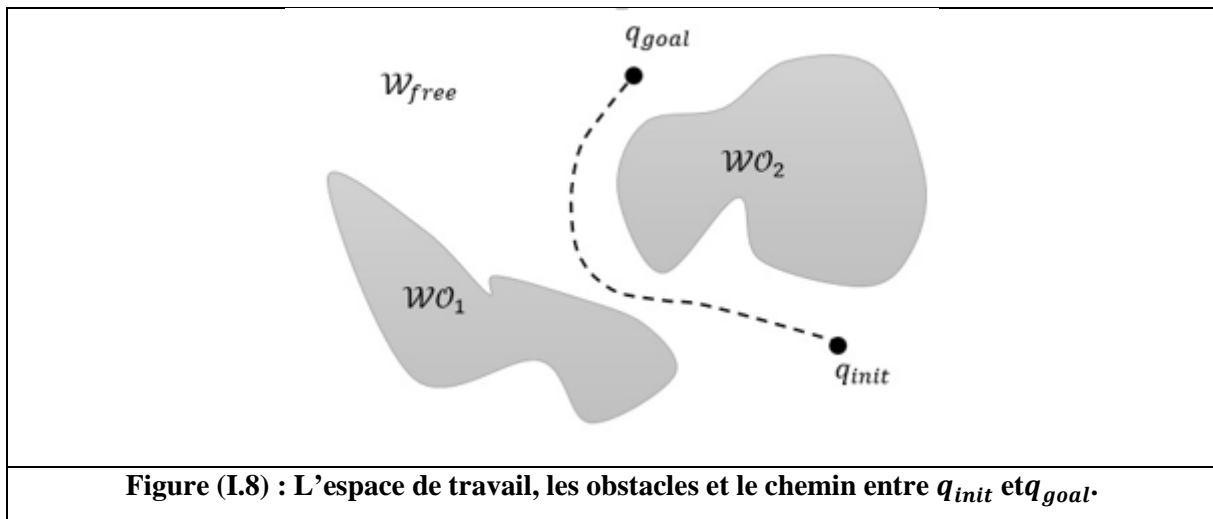
I.6.4. Espace de travail :

L'espace de travail $W = \mathbb{R}^N$ où $N = 2$ si le robot se déplace dans un plan, $N = 3$ s'il se déplace dans un espace tridimensionnel, est un environnement statique contenant des obstacles. Supposons que WO_i est le $i^{\text{ème}}$ obstacle tel que $1 \leq i \leq n$. L'espace de travail libre est l'ensemble de points exprimés par la fonction suivante :

$$W_{free} = W \setminus \bigcup_i WO_i \quad (I.1)$$

Où le \setminus est un opérateur de soustraction.

Le but est de trouver un chemin sans collision pour que le robot puisse passer d'une position et orientation initiale à une position et orientation de but [1, 9, 11].



I.6.5. Espace de configuration :

L'emplacement d'un robot est représenté comme une configuration. La configuration d'un robot est un ensemble minimal de paramètres qui sont nécessaires pour décrire son emplacement exact. Cet ensemble de toutes les configurations possibles pour le robot est appelé *espace de configuration*, et c'est là que la planification de mouvement se produit. Il est d -dimensionnel.

Dans chaque configuration q , le robot occupe un ensemble de points noté $R(q)$ dans un espace de travail W . Les n obstacles contenus dans W sont notés O_i ; $1 \leq i \leq n$; pour chaque O_i il y a une contrepartie CO_i dans l'espace de configuration C . Cet obstacle de l'espace de configuration peut être défini comme :

$$CO_i = \{q \in C \mid R(q) \cap O_i \neq \emptyset\}, \quad (I.2)$$

Ce qui signifie que CO_i est un ensemble de toutes les configurations où le robot R serait en collision avec l'obstacle O_i .

Dans le cas d'un robot à corps rigide, l'espace de configuration libre peut être défini comme :

$$C_{free} = C - \bigcup_{i=1}^n CO_i, \quad (I.3)$$

Et c'est un ensemble de configurations où le robot ne heurte aucun des obstacles [9].

Pour en savoir plus sur l'espace de travail et l'espace de configuration voir [1, 9, 11].

I.7. Orientation et matrices de rotation en trois dimensions :

La position d'un point dans l'espace par rapport à un repère peut être exprimée par ses coordonnées selon x,y et z :

$$M = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Mais pour un corps rigide, on a besoin de la position et l'orientation par rapport à un repère de référence (fixe), pour cela on associe un repère mobile à ce corps, et on étudie la position et l'orientation du repère mobile par rapport au repère fixe [3].

Soient un repère (A) fixe avec les axes (x,y,z) et les vecteurs unitaires associés, et le repère (B) mobile (rotatif) d'axes (x',y',z') ayant la même origine que (A).

Une rotation de (B) d'un angle α autour de l'axe ox du repère de référence (A), selon la figure I.2, peut être décrite comme suit :

$$(A) = R_B^A \cdot (B) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix} \cdot (B) \quad (I.4)$$

Avec R_B^A noté souvent par $R_{x,\alpha}$ est la matrice de rotation propre orthogonale avec :

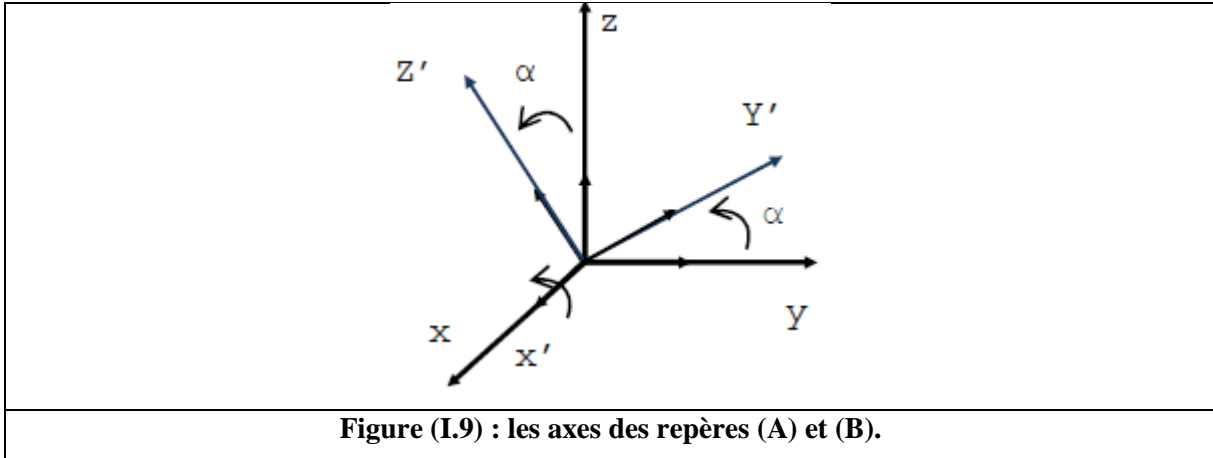
$$\det R_{x,\alpha} = 1 \quad (I.5)$$

$$R_{x,\alpha}^T R_{x,\alpha} = I \quad (I.6)$$

$$(R_{x,\alpha})^{-1} = (R_{x,\alpha})^T \quad (I.7)$$

Dont les éléments peuvent être obtenus par le produit scalaire : $R_B^A = \begin{bmatrix} x \cdot x' & x \cdot y' & x \cdot z' \\ y \cdot x' & y \cdot y' & y \cdot z' \\ z \cdot x' & z \cdot y' & z \cdot z' \end{bmatrix}$ (I.8)

Comme le produit scalaire de deux vecteurs unitaires est égale au cosinus de l'angle entre eux, $R_{x,\alpha}$ est appelée matrice des cosinus directeurs.



Les rotations autour des axes oy et oz donnent :

$$R_{y,\alpha} = \begin{bmatrix} c\alpha & 0 & s\alpha \\ 0 & 1 & 0 \\ -s\alpha & 0 & c\alpha \end{bmatrix} \quad (I.9)$$

$$R_{z,\alpha} = \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (I.10)$$

Remarque : selon la littérature, la matrice de rotation décrivant l'orientation de (B) par rapport à (A) est notée par : $R_A^B = {}^A_B R = {}^A R_B$.

L'ensemble de toutes les matrices réelles satisfaisant à la fois l'équation (I.5) et (I.6) est appelée l'ensemble de matrices orthogonales spéciales. En général, l'ensemble de toutes les matrices orthogonales $N \times N$ spécial est appelée SO (N) et l'ensemble de toutes les rotations dans l'espace tridimensionnel est appelé SO (3)[1].

Ces matrices de rotation ne servent pas seulement à exprimer l'orientation, mais peuvent aussi être utilisées pour transformer les coordonnées d'un point dans un repère vers un autre [13].

➤ **Rotation composée :**

Dans le cas où plusieurs rotations sont faites, la matrice de rotation totale est un produit de toutes ces matrices, et ceci en respectant deux règles :

Soit trois rotations successives : $R_{x,\alpha} \rightarrow R_{z,\theta} \rightarrow R_{y,\phi}$

- Si c'est une rotation du repère mobile (B) autour des axes du repère fixe (A), alors la matrice de rotation résultante correspond à une prémultiplication

Exemple :

Dans le cas des trois rotations : $R_{x,\alpha} \rightarrow R_{z,\theta} \rightarrow R_{y,\phi}$, la matrice de rotation composée est :

$$R = R_{y,\phi} \cdot R_{z,\theta} \cdot R_{x,\alpha}$$

- Si c'est une rotation du repère mobile (B) autour de ces propres axes, alors la matrice de rotation résultante correspond à une poste-multiplication.

Exemple :

Dans le cas des trois rotations : $R_{y,\phi} \rightarrow R_{z,\theta} \rightarrow R_{x,\alpha}$, la matrice de rotation composée est :

$$R = R_{y,\phi} \cdot I_3 \cdot R_{z,\theta} \cdot R_{x,\alpha}$$

➤ **Les angles d'Euler :**

Les angles d'Euler (φ , θ , ψ) sont de loin les paramétrisations de rotation les plus connues. Ils sont générés par trois rotations successives sur des axes indépendants [1].

Ils décrivent l'orientation d'un corps solide par rapport à une référence fixe. La matrice de rotation résultante est donnée par :

$$R(\varphi, \theta, \psi) = R_{z,\varphi} R_{y,\theta} R_{z,\psi} \tag{I.11}$$

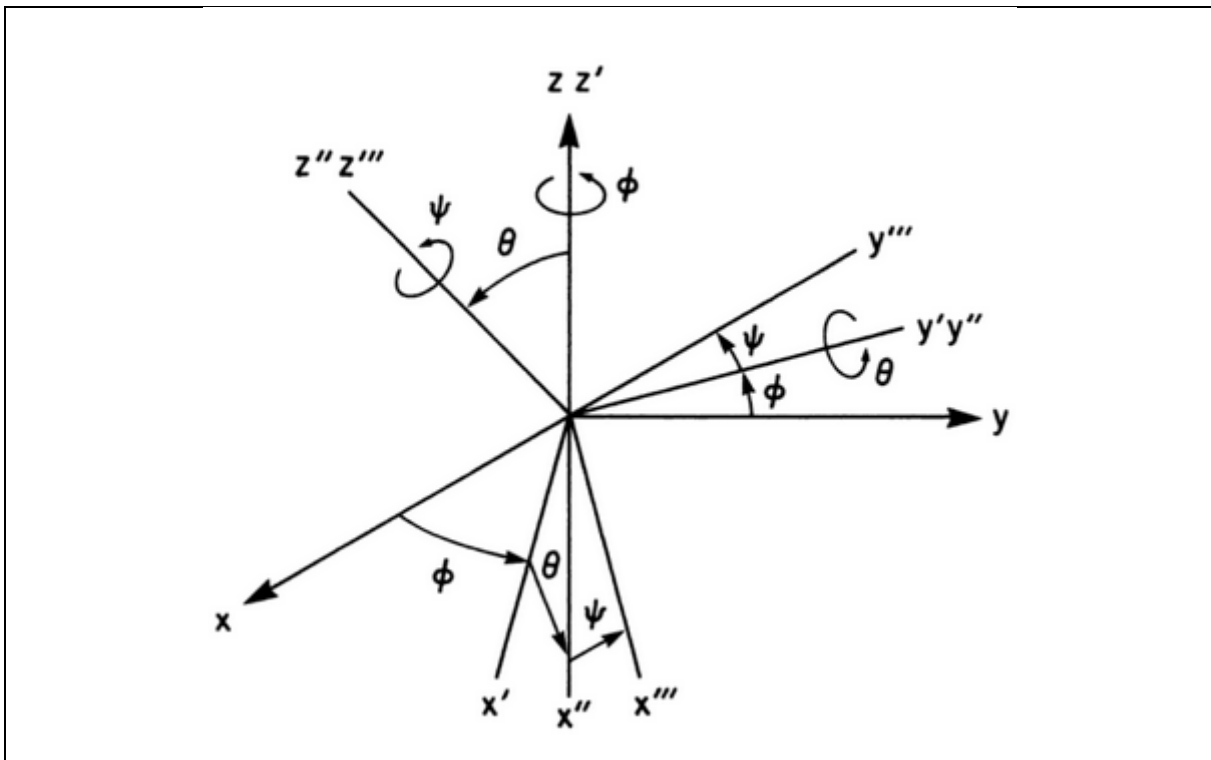


Figure (I.10) : les angles d'Euler [4].

➤ **Transformation Homogènes :**

Les matrices de transformations homogènes permettent de travailler avec des matrices de 4×4 incluant, à la fois, la rotation et la position.

$$T = \begin{bmatrix} R & P \\ N & 1 \end{bmatrix}; R \in SO(3); P \in \mathbb{R}^3 \quad (I.12)$$

Avec : $N = (0, 0, 0)$; P le vecteur de position ; R la matrice de rotation ; 1 le facteur d'échelle.

La matrice T est la matrice de transformation homogène, elle appartient à $SE(3)$, tel que :

$$SE(3) = \mathbb{R}^3 \times SO(3) \quad (I.13)$$

Pour plus de détails sur 1.7 voir [1] et [4].

I.8. Mouvement d'un corps rigide :

Charles'theorem[5] :

1. Chaque mouvement d'un corps rigide peut être considéré comme une translation dans l'espace et une rotation autour d'un point ;
2. Chaque déplacement spatial d'un corps rigide peut être affecté de façon équivalente par une rotation unique autour d'un axe et par une translation sur le même axe.

Dans la notation moderne, (1) est exprimée en disant que chaque point x dans un corps rigide peut être déplacé comme : $x' = Rx + b$ (I.14)

Où $R \in SO(3)$ est une matrice de rotation, et $b \in \mathbb{R}^3$ est un vecteur de translation. La paire $g = (R, b) \in SO(3) \times \mathbb{R}^3$ décrit à la fois le mouvement d'un corps rigide et la relation entre les cadres fixés dans l'espace et dans le corps. De plus, les mouvements caractérisés par une paire (R, b) peuvent décrire le comportement soit d'un corps rigide, soit d'un objet déformable qui a un mouvement de corps rigide pendant l'intervalle de temps pour lequel cette description est valide [1].

I.9. Modélisation géométrique :

Un robot manipulateur est constitué d'un ensemble d'organes reliés entre eux par des articulations. Les articulations peuvent être rotoïdes ou bien prismatiques, comme elles peuvent être plus complexes. Le modèle géométrique directe et inverse expriment la situation de l'organe terminal en fonction des variables articulaires et du mécanisme et inversement [6].

Pour un robot mobile dans le plan ou bien dans l'espace, son modèle géométrique est simple, car il n'y a que deux repères, un repère fixe et un repère mobile associé au robot. Mais pour les robots manipulateurs, le cas est différent [13].

I.9.1. Modèle géométrique directe MGD :

La modélisation des robots exige une méthode adéquate pour la description de leur structure. Plusieurs méthodes ont été proposées, telles que la convention de Denavit-Hartenberg(DH) et DH modifiée, et celle de Khalil-Kleinfinger (KK).

➤ **Convention de Denavit-Hartenberg :**

D-H propose une méthode matricielle permettant d'établir systématiquement un repère pour chaque corps d'une chaîne articulée afin de décrire les relations de translation et de rotation entre deux corps adjacents. 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 précède.

Pour établir les paramètres de DH, chacun des repères est établi en se basant sur trois règles :

1. L'axe Z_{i-1} est selon l'axe de mouvement de l'articulation i .
2. L'axe x_{i-1} est aligné entre les articulations i et $(i+1)$.
3. L'axe Y_i complète le repère i pour former un repère « main droite ».

Dans cette convention, le repère R_i est placé à la fin du corps C_i . Il est associé à l'articulation $i+1$.

Dans la convention DH modifiée (MDH), le repère R_i est placé au début du corps C_i . Il est associé à l'articulation.

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

$$T_{i-1}^i = T_{z,\theta_i} \cdot T_{z,d_i} \cdot T_{x,a_i} \cdot T_{x,\alpha_i} = \begin{bmatrix} c\theta_i & -c\alpha_i \cdot s\theta_i & s\alpha_i \cdot s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i \cdot c\theta_i & -s\alpha_i \cdot c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.15)$$

Avec : a_i : la longueur du corps ; α_i : angle d'articulation du corps ;

θ_i : Angle entre deux corps consécutifs ; d_i : la distance entre les deux corps.

I.9.2. Modèle géométrique inverse MGI :

Le MGI consiste à calculer les coordonnées articulaires qui amènent l'organe terminal dans une situation désirée, spécifiée par ses coordonnées opérationnelles [7].

On peut distinguer deux approches pour le calcul du MGI, approche analytique et approches numériques (itératives). Les méthodes analytiques sont les plus utilisées, vu la lenteur des méthodes itératives.

Le calcul du MGI se fait selon les méthodes suivantes : méthode de Paul et celle de Pieper.

Pour en savoir plus sur 1.6 voir [1], [6], [7].

I.10. Modélisation cinématique directe et inverse [3] :

Un repère mobile possède deux vitesses : linéaire et angulaire.

La vitesse linéaire est associée à un point mobile, tandis que la vitesse angulaire est associée à un repère rotatif. Ainsi, la vitesse linéaire d'un repère mobile est simplement la vitesse de son origine. La vitesse angulaire pour un repère mobile est liée à la dérivée temporelle de la matrice de rotation qui décrit l'orientation instantanée du repère.

L'idée utilisée pour calculer la dérivée d'une matrice est de multiplier cette matrice par une matrice spéciale que l'on appelle la matrice jacobéenne $J(q)$.

$$\dot{x} = J(q)\dot{q} \quad (\text{I.16})$$

$$\dot{q} = J^{-1}(q)\dot{x} \quad ; \quad \text{si } J(n \times n) \text{ est inversible} \quad (\text{I.17})$$

L'objectif du modèle cinématique direct est de décrire les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires, et l'objectif du modèle cinématique inverse est de calculer à partir d'une configuration donnée les vitesses articulaires qui assurent une vitesse opérationnelle imposée et pour ce faire on a besoin de calculer J^{-1} , ce qui pose quelques problèmes, tels que *la singularité* [14], et *la redondance* [14].

L'une des méthodes de calcul de la matrice jacobéenne est la méthode de *propagation des vitesses*, qui consiste à calculer les vitesses linéaires et angulaires en fonction des vitesses articulaires, et ce, en commençant de la base vers l'effecteur.

Si la matrice jacobéenne est carrée et non singulière, son inverse est calculé simplement, mais pour le cas où J n'est pas carrée (un degré de liberté de plus), on parle alors de la redondance, il y a alors possibilité de plusieurs solutions. Quelques méthodes sont proposées pour traiter ce problème, telles que : la pseudo-inverse avec un terme d'optimisation, les vecteurs de coordonnées supplémentaires. Et pour le cas où J est carrée, mais singulière, une solution consiste à augmenter le nombre de degrés de liberté du mécanisme.

I.11. Modélisation dynamique [14] :

La modélisation dynamique consiste en la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations des articulations. Le modèle dynamique est représenté par une relation de la forme :

$$\tau = f(q, \dot{q}, \ddot{q}, f_e), \quad (\text{I.18})$$

Avec :

- τ : le vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde(pivot) ou prismatique,
- q : le vecteur des coordonnées articulaires,
- \dot{q} : le vecteur des vitesses articulaires,
- \ddot{q} : le vecteur des accélérations articulaires.
- f_e : le vecteur représentant les forces et les moments extérieurs que le robot exerce sur l'environnement.

On peut appeler la relation (I.18) *le modèle dynamique inverse*, ou tout simplement le modèle dynamique.

Le modèle dynamique directe exprime les accélérations articulaires \ddot{q} en termes de positions q , vitesses \dot{q} et des couples τ . Par conséquent, il est représenté par la relation :

$$\ddot{q} = g(q, \dot{q}, \tau, f_e). \quad (\text{I.19})$$

Deux grandes approches existent pour calculer les équations dynamiques d'un robot. Celle de Newton-Euler [15-16], qui décrit le comportement dynamique d'un système en termes de forces et moments, et celle de Lagrange [17, 18, 19] qui le décrit en termes de travail et d'énergie.

Pour plus de détail consulter les références suivantes : [1, 6, 14].

Chapitre II :
Planification de mouvement et tests
de collisions

Chapitre II : planification de mouvement et tests de collisions

II.1. Planification de trajectoire et planification du chemin :

La trajectoire décrit le mouvement désiré d'un manipulateur dans l'espace multidimensionnel.

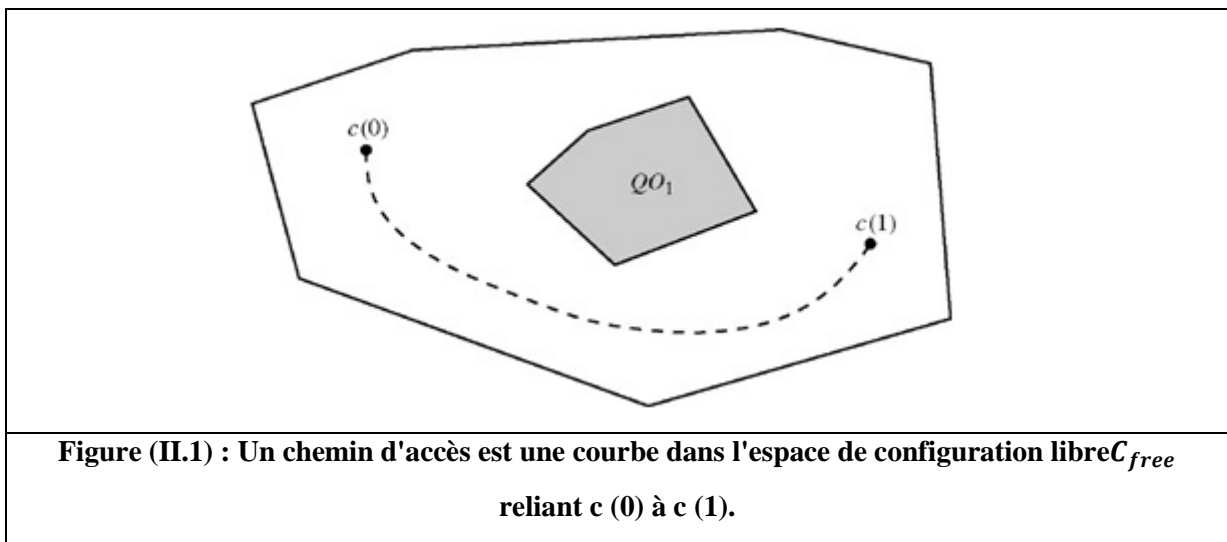
Elle fait référence à la variation de la position, de la vitesse et de l'accélération de chaque articulation, selon une fréquence allant de 60 à 2000 Hz [8]. La planification de la trajectoire se fait par un planificateur de mouvement qui calcule une fonction $q^d(t)$ de classe C^2 qui spécifie complètement le mouvement du robot pendant qu'il traverse le chemin.

➤ La différence entre la planification du chemin et la planification de la trajectoire :

Un chemin d'accès est une courbe continue sur l'espace de configuration. Il est représenté par une fonction continue qui mappe un paramètre Path, qui est pris généralement dans l'intervalle unitaire $[0,1]$, à une courbe dans C_{free} (figure II.1). La solution au problème de planification du chemin est une fonction continue $c \in C^0$ telle que lorsque le chemin est paramétré par le temps t , alors $c(t)$ est une trajectoire, et les vitesses et les accélérations peuvent être calculé par le planificateur de mouvement [10].

La fonction « c » est comme suit :

$$c: [0,1] \rightarrow C \text{ ou } c(0) = q_{start}, c(1) = q_{goal} \text{ et } c(s) \in C_{free} \forall s \in [0,1]$$



Dans la planification du mouvement on a besoin d'un concept qui est l'espace de travail et l'espace de configuration, qui sont bien définis dans (I.6.4 et I.6.5).

II.1.1. Planification de trajectoire et planification de mouvement [11] :

Le problème de planification de trajectoire pour un robot dynamique est de trouver des entrées de contrôle (force) $u(t)$ donnant une trajectoire $q(t)$ qui évite les obstacles. Cela peut être considéré comme un problème complet de « planification de mouvement », par opposition à un problème de « planification de trajectoire » qui ne demande qu'une courbe faisable $q(s)$ dans l'espace de configuration, sans référence à la vitesse d'exécution.

Il existe deux approches de planification de trajectoire pour un système dynamique :

➤ L'approche découplée :

Elle implique d'abord la recherche d'un chemin dans l'espace de configuration, puis la recherche d'un temps optimal de mise à l'échelle pour le chemin soumis aux limites de l'actionneur. Les détails de cette approche sont représentés dans [11].

➤ L'approche directe :

Dans cette approche, les méthodes de planification de la trajectoire se déroulent dans l'espace de l'état du système

II.1.2. Résoudre le problème de planification de mouvement :

Différentes approches ont été utilisées pour résoudre le problème de planification de mouvement. On essaye de décrire quelque'une de ces approches.

➤ Recherche basée sur la grille :

Une approche commune pour trouver un chemin pour un robot est d'utiliser une recherche basée sur la grille. L'idée est de diviser l'espace de configuration en une grille. Les cellules de la grille sont typiquement formées uniformément et dans le cas bidimensionnel ils peuvent être, par exemple, des carrés ou des hexagones. Chaque cellule peut être soit libre, soit bloquée. Les cellules libres sont celles qui n'ont pas d'obstacles et les cellules bloquées sont celles qui ont. Un robot occupe une cellule à la fois et peut se déplacer vers les cellules adjacentes. Il est possible de penser que les cellules libres forment un graphe et, par conséquent, le chemin entre deux cellules peut être trouvé avec un algorithme de recherche graphique [9].

➤ Méthodes de décomposition des cellules :

Elles ressemblent à une recherche de grille de base à certains égards. Au lieu de cellules uniformément formées, C_{free} est divisé en cellules qui peuvent toutes être différemment en forme et de taille. Tout comme dans la recherche de grille de base, deux cellules sont adjacentes

les unes aux autres si elles ont une frontière commune. Comme les cellules ne sont pas uniformément formées, les informations sur la contiguïté des cellules sont explicitement stockées dans un graphique. Le robot peut se déplacer librement à l'intérieur d'une cellule et, en se basant sur les informations du graphe de contiguïté, le robot peut aussi se déplacer vers d'autres cellules. Il est important qu'il soit très facile de trouver un chemin pour un robot entre deux configurations à l'intérieur d'une cellule.

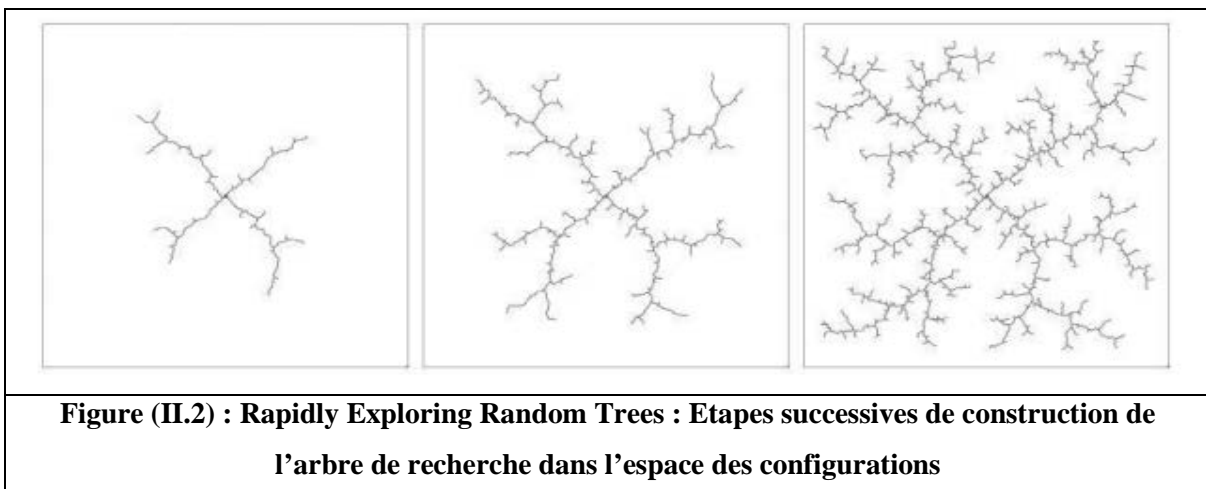
Par conséquent, les cellules doivent être de préférence convexes [9].

➤ **Planificateur de chemins aléatoires (RPP) :**

Utilise des champs potentiels pour trouver un chemin et si elle est bloquée dans un minimum local, il essaie d'utiliser des mouvements aléatoires pour l'échapper. L'inconvénient de la méthode était qu'il nécessitait de nombreux paramètres qui devaient être définis [9].

➤ **Rapidly Exploring Random Trees (exploration rapide de l'arbre aléatoire)(RRT):**

Représente certainement l'une des approches de planification de mouvement les plus répandues à l'heure actuelle. À partir d'une configuration initiale q_0 , l'espace de configuration du système est exploré en choisissant aléatoirement à chaque itération une nouvelle configuration q_{nv} non obstruée par les obstacles vers laquelle se diriger. La branche la plus proche de l'arbre déjà construit est alors déterminée puis étendue en direction de q_{nv} . En répétant le processus, l'espace de recherche est alors rapidement couvert, et un chemin vers toutes configurations de cet espace peut alors être facilement déterminé s'il en existe un (**Figure II.2**). Le but de la planification étant néanmoins d'atteindre une configuration finale q_f le processus essaie de déterminer un chemin liant la configuration la plus proche de l'arbre à q_f après un certain nombre d'itérations de l'expansion de l'arbre [15].



De nombreuses méthodes mentionnées dans cette section construisent une sorte de feuille de route. Une feuille de route est une approximation de C_{free} et de le construire au lieu d'essayer d'obtenir une représentation exacte peut être très utile dans les problèmes higher dimensional. Les planificateurs de cartes de route probabilistes essaient de faire exactement cela et ils sont étudiés plus en détail dans le chapitre suivant.

II.2. Test de collision :

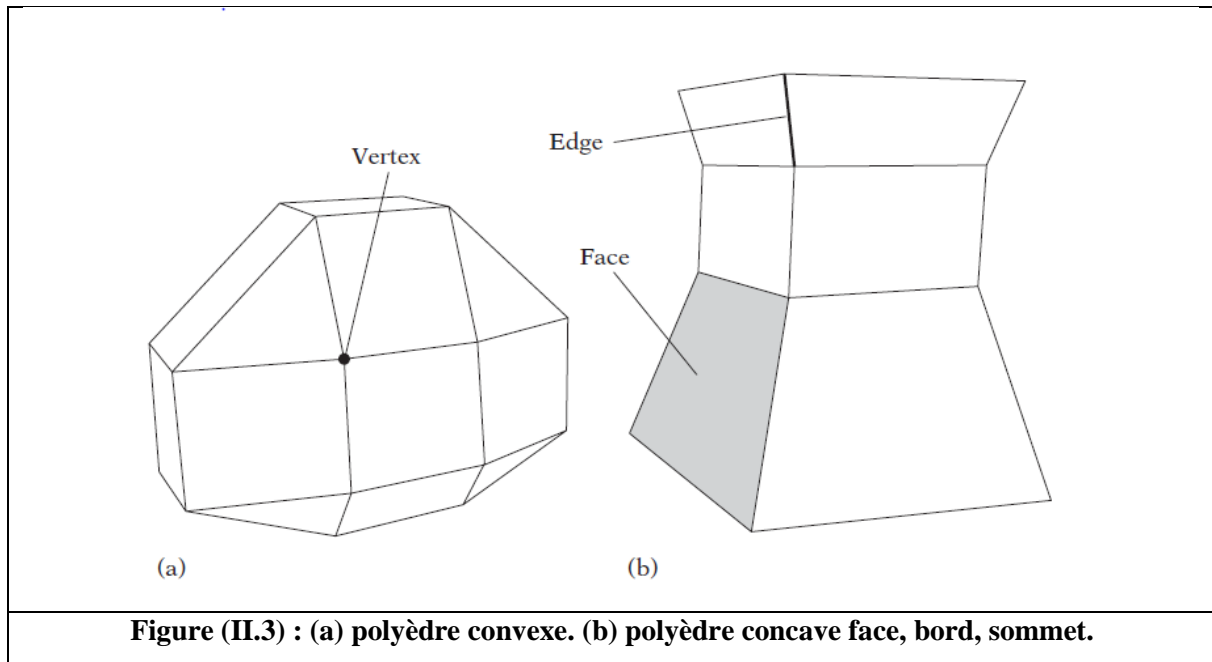
La détection de collision est fondamentale pour de nombreuses applications variées y compris jeux, simulation physique, la robotique. Par exemple pour les jeux informatiques la détection de collision assure que l'illusion du monde solide est entretenue, il empêche par exemple les personnages joueurs de traverser les murs, la détection de collision peut représenter un grand pourcentage du temps nécessaire pour compléter un cadre de jeux.

Dans cette deuxième partie du chapitre, nous allons parler de la détection et des méthodes de test de collision. Mais avant cela, pour sélectionner les algorithmes de détection de collision appropriés, il est important de prendre en compte les types de représentation géométrique qui seront utilisés pour la scène et ces objets.

II.2.1. Les polyèdres :

Un polyèdre est la contrepartie 3D d'un polygone c'est une région bornée et connectée de l'espace sous la forme d'un solide a multiples facettes la limite du polyèdre est composée d'un certain nombre de faces (plates) connecté. Comme pour les polygones, la limite du polyèdre divise l'espace en deux régions disjointes : l'intérieur et l'extérieur.

Un polyèdre P est dit convexe si et seulement si toutes les faces de P tous les sommets de P sont (strictement) du même côté de cette face. Un polyèdre convexe (borné) est également appelé en tant que polytope. Comme les polygones, les polytopes peuvent également être décrits comme l'intersection d'un nombre fini de demi-espaces.



II.2.2 Problème du point dans le polyèdre :

Dans cette section on décrit quelques différentes approches. Le test le plus simple pour tester si un point se trouve à l'intérieur d'un polyèdre donné est lorsque le polyèdre est convexe et donné implicitement en tant que volume d'intersection d'un certain nombre de demi-espaces dans ce cas le point se situe à l'intérieur du polyèdre s'il se trouve à l'intérieur de chaque demi-espace [16].

D'autres méthodes utiles se trouvent dans la littérature qui peuvent être utilisées à la fois pour les polyèdres convexe et concave on cite pour exemple :

- Si le polyèdre est donné comme un ensemble de sommets sans aucune information de connectivité, la méthode de GJK qui est écrite plus tard dans ce chapitre sert de confinement efficace des points approche. Seul le polyèdre convexe peut être donné dans ce format.
- Si le polyèdre est donné comme un maillage fermé, une approche consiste à d'abord construire une feuille solide Arbre BSP, ensuite compte tenu de cet arbre si une requête de point sur l'arbre se termine dans une feuille solide, le point est l'intersection du polyèdre (sinon il ne l'est pas) cette méthode fonctionne à la fois pour les polyèdres convexe et concave [16].
- Une méthode qui travaille avec un polyèdre donné soit comme un maillage ou comme une intersection des demi-espaces est de tirer un rayon à partir du point testé dans

n'importe quelle direction (généralement le long un axe majeur, tel que l'axe + x) et compte le nombre de faces intersectées. Si le nombre impair de faces est coupé, le point est à l'intérieur du polyèdre ; autrement, le point est en dehors de cela. Cette méthode aussi fonctionne à la fois.

Pour les polyèdres convexes et concaves [16].

II.2.3. Rayon ou segment se croisant contre le polyèdre convexe :

Pour tester le croisement d'un segment $S(t) = A + t(B - A), 0 \leq t \leq 1$, contre un polyèdre convexe une représentation pratique pour le polyèdre est de le décrire comme l'intersection d'un ensemble de demi-espaces. Pour un tel format, le segment intersecte le polyèdre s'il existe une valeur pour $t, 0 \leq t \leq 1$, pour laquelle $S(t)$ se trouve à l'intérieur de tous les demi-espaces.

Un moyen efficace de déterminer si c'est le cas consiste simplement à découper le segment contre chaque demi-espace, en coupant la partie du segment qui se trouve en dehors du demi-espace. Si pendant ce processus de découpage, le segment devient longueur nulle, le segment ne peut pas croiser le polyèdre et le processus peut s'arrêter avec un résultat de "pas d'intersection."

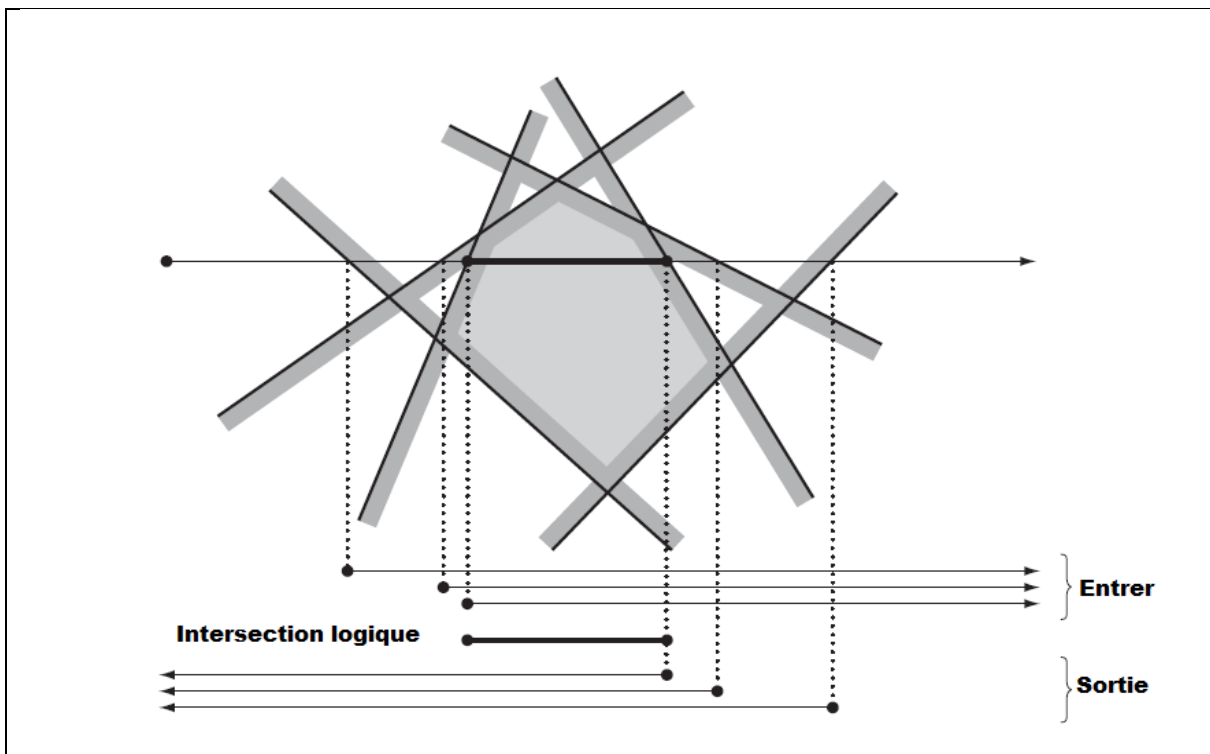


Figure (II.4) : L'intersection d'un rayon (ou segment) contre un polyèdre convexe (défini comme l'intersection d'un ensemble de demi-espaces) est l'intersection logique du rayon coupé contre tous demi-espaces.

II.2.4. Détection de collision entre polyèdres convexes :

II.2.4.1. Détection d'intersection vide :

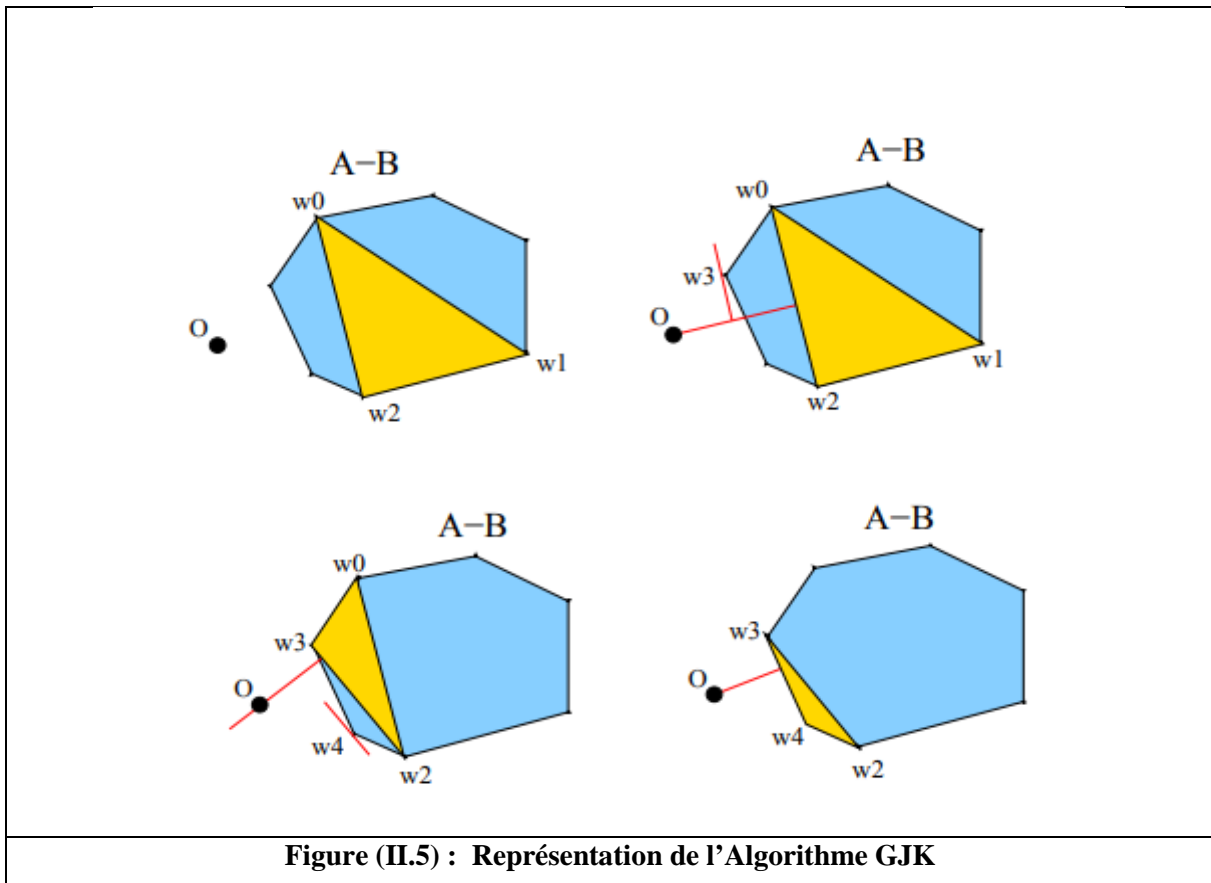
Les différents algorithmes permettant la détection d'intersection vide cherchent à séparer les deux polyèdres : soit on trouve un plan tel que les deux objets soient de part et d'autre du plan. Soit-on calcule une borne minimale de la distance les séparant.

➤ Calcul de distance interpolyèdre

Trois grandes approches ont été proposées, soit précisément, soit par borne intérieure, la distance entre deux polyèdres convexes.

- **Approche Gilbert-Johnson-Keerthi (GJK):**

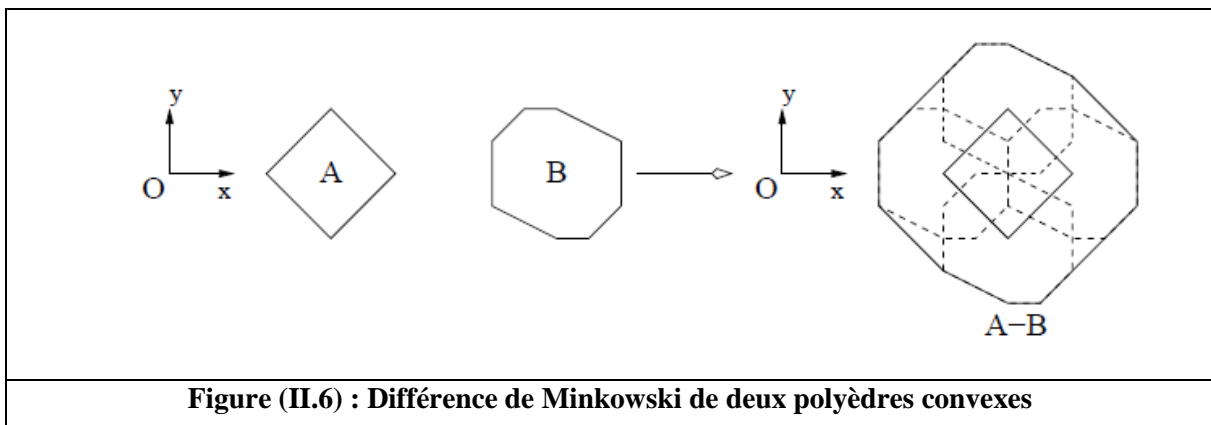
Elle se base sur la différence de Minkowski des polyèdres, que l'on approxime par des simplexes purs évitant sa construction explicite. Voir (**Figure II.5**). En cyan, la différence d'itération, on calcule la distance du simplexe en jaune par rapport à l'origine. Cette distance fournit une direction. On cherche alors le point support du polyèdre dans cette direction afin de former un nouveau simplexe



La différence Minkowski (notée M) est définie de la manière suivante [17] :

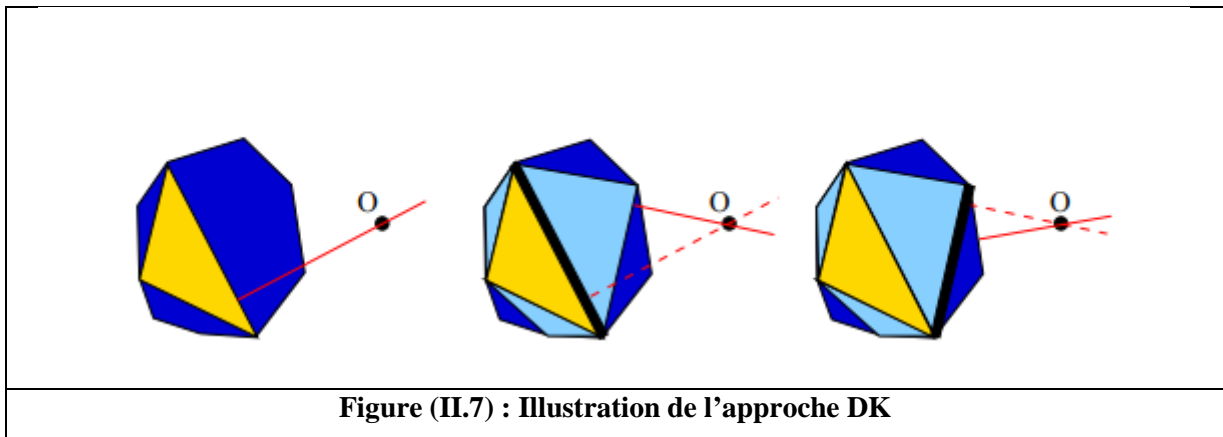
$$M = A - B = \{x - y \mid x \in A, y \in B\}$$

Cette différence est convexe lorsque les polyèdres A et B sont convexe L'intérêt de cette différence est que l'on ramène le calcul de la distance entre deux objets à la distance entre M et l'origine. De plus si l'origine se trouve à l'intérieur de M les objets sont en collision [17].



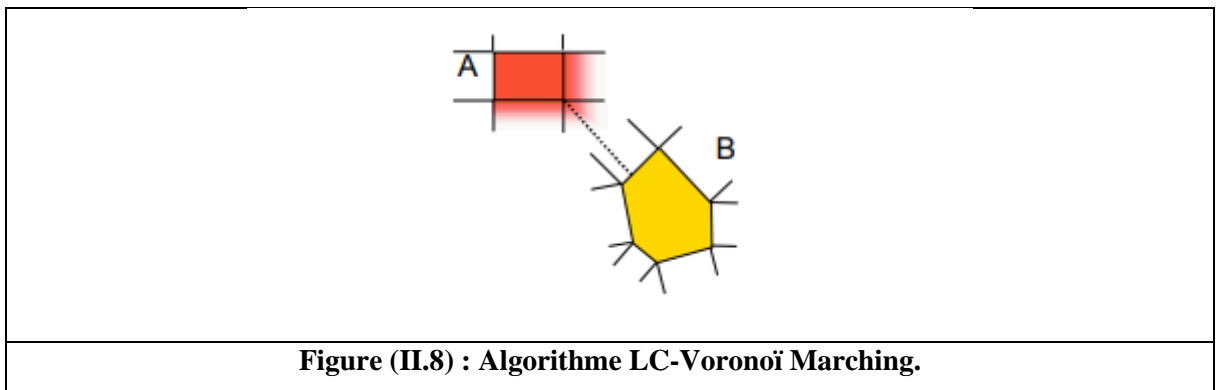
- **Approche dobkin et kirkpatrick (DK) :**

Cette approche consiste à construire une suite d'approximation de plus en plus grossière des polyèdres pour évaluer par itération la distance voire figure (**Figure II.7**) Au départ l'algorithme se base sur un simplexe approximant le polyèdre en exploitant certains de ces sommets ; à chaque étape on ajoute des sommets ; des sommets du polyèdre permettant d'étendre le simplexe de départ en direction de l'origine.



- **Approche L-C voronoï marching :**

Se base sur la découpe de l'espace environnant chaque polyèdre en régions de voronoï voir figure (**Figure.II.8**). On cherche le couple d'éléments de l'objet tel que l'un des éléments est dans la région de Voronoï associée à l'autre élément et réciproquement.



II.2.4.2. Calcul d'interpénétration :

Lorsque deux objets sont en collision il est nécessaire d'estimer le degré d'interpénétration de ces deux objets, pour cela on définit la profondeur d'interpénétration c'est la norme de la plus petite translation qui permet d'avoir les objets en contact (et seulement en contact) cette distance

peut être facilement obtenue avec la différence de Minkowski : c'est la plus petite distance entre M et l'origine [17]. Plusieurs algorithmes se basent sur le GJK pour la calculer ou étendent le voronoï Marching. Une autre approche a une complexité inférieure au cas linéaire, mais est moins robuste. Enn, une dernière méthode passe dans l'espace dual (espace des normales) pour trouver la direction de la translation.

II.2.4.3. Construction de l'intersection :

La construction de l'intersection entre deux polyèdres est souvent très lourde et rarement nécessaire. En géométrie algorithmique, la construction de l'intersection de deux polyèdres convexes peut se faire avec une complexité linéaire, mais ce type d'algorithmes étant difficile à mettre en œuvre, d'autres approches sont proposées, notamment celles utilisant les partitions binaires de l'espace (BSP : Binary Space Partionning, voir [16]).

Chapitre III :

**PRM pour la planification de
mouvements d'un corps solide de
forme polyédrique dans une scène
contenant des obstacles polyédriques**

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

III.1. Introduction :

Dans ce chapitre, nous allons étudier l'implémentation de PRM. Nous commençons par le test de collision qui constitue la partie la plus sollicitée durant la construction de la PRM. L'approche que nous avons choisie est basée sur la recherche d'une droite sur laquelle les projections des deux corps convexes sont disjointes [20]. Ensuite nous traiterons l'échantillonnage de configurations qui est basé sur l'utilisation de distribution uniforme parfois combinée avec des opérations supplémentaires pour favoriser certaines propriétés. Pour la connexion de la PRM, nous avons considéré la méthode de k-voisins qui augmente le nombre de chemins obtenus en réponse à une requête, ainsi que la méthode de forêt qui permet d'éviter le calcul prohibitif pendant la phase de recherche qu'introduit la méthode précédente.

Nous allons présenter les différentes techniques que nous avons programmées sous MATLAB et nous allons donner les résultats de simulation pour différents cas. Dans ce travail nous considérons un robot sous forme polyédrique en mouvement dans une scène qui contient des obstacles qui sont aussi à leurs tours polyédriques. La supposition est raisonnable, car, quelle que soit la forme du corps, on peut le border par un polyèdre convexe de volume proche. En plus, l'utilisation de polyèdres facilite le test de collision pendant le mouvement qui constitue la partie la plus difficile pour un planificateur probabiliste.

III.2. Test de collision :

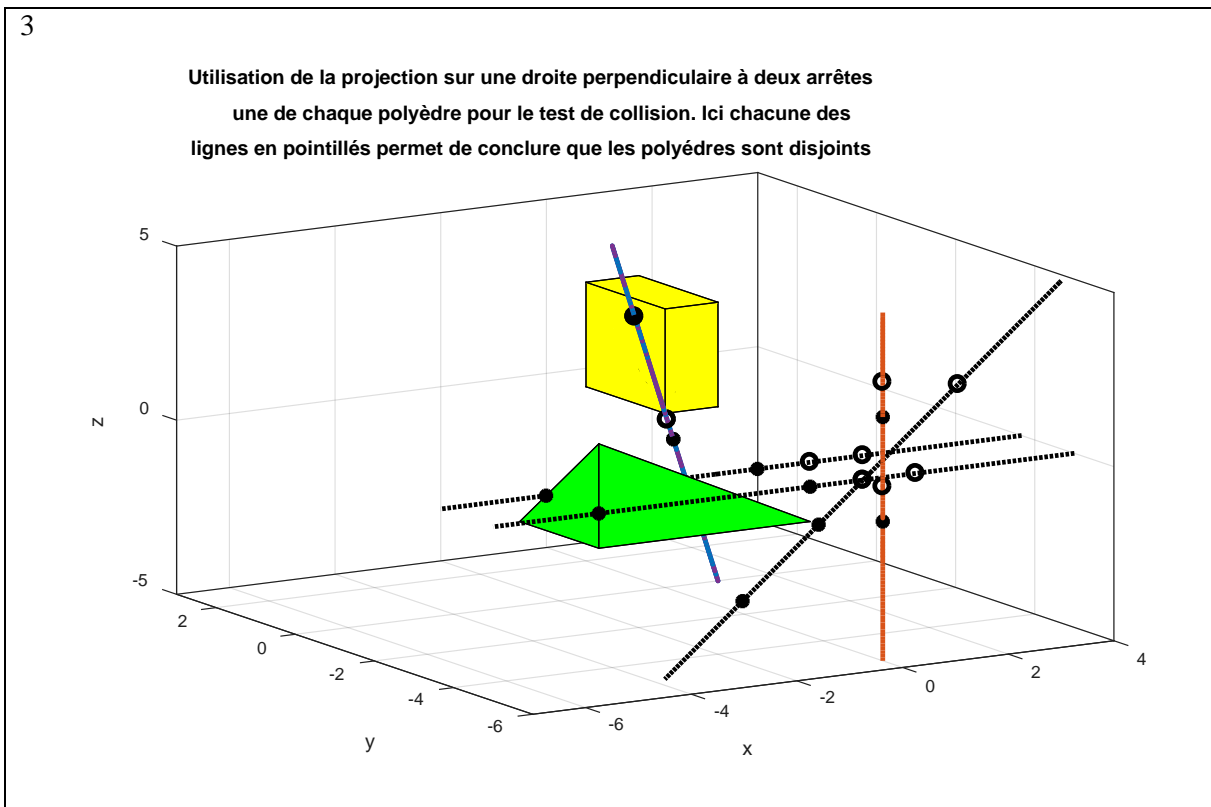
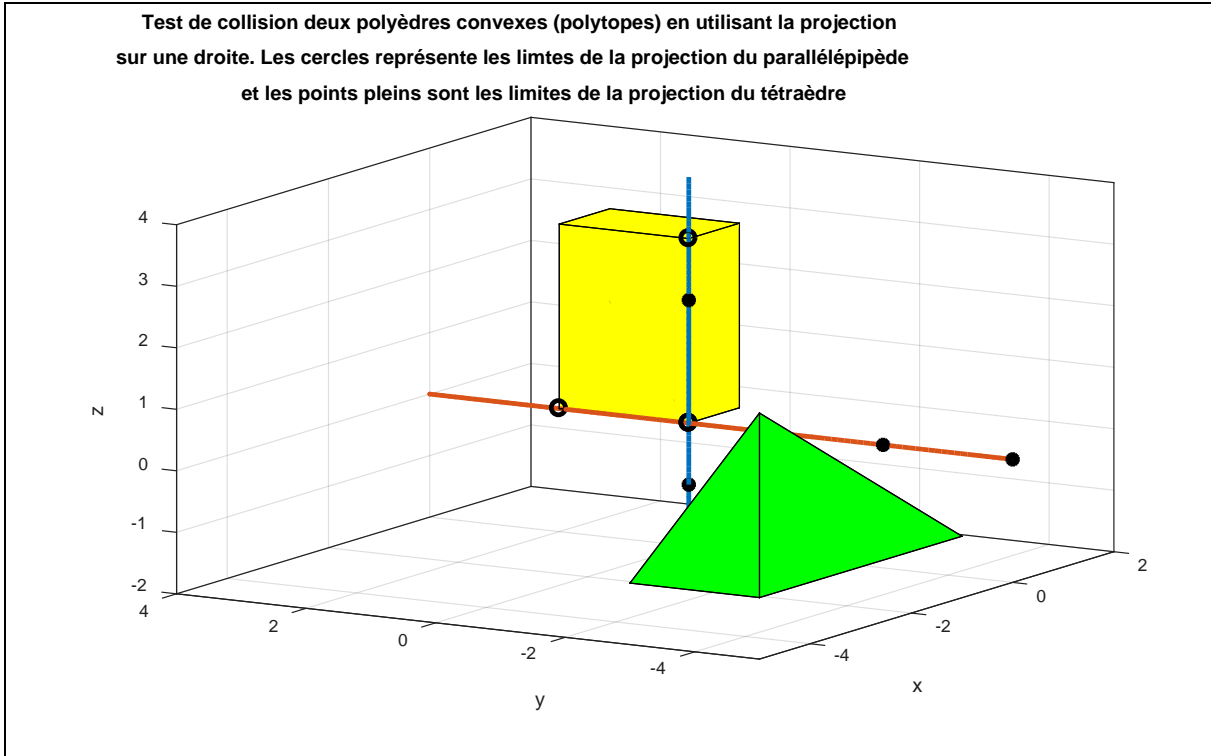
III.2.1. Théorème [20] :

Deux polyèdres convexes (polytopes) sont disjoints **si et seulement s'il existe** une droite perpendiculaire à l'une des faces de l'un des deux polyèdres telle que la projection des deux polyèdres sur cette droite donne deux intervalles disjoints. Ou qu'il existe une droite perpendiculaire en même temps à un segment du premier polyèdre et à un segment du deuxième polyèdre telle que les projections des deux polyèdres sur cette droite donnent deux intervalles disjoints.

Pour accomplir le test de collision en utilisant ce théorème, on teste les projections des deux polyèdres sur les axes spécifiés l'un après l'autre est on arrête le test dès qu'un axe qui donne

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

deux projections disjointes est trouvé. Sinon on continue jusqu'à ce qu'on termine avec tous les axes après quoi une collision est détectée.



Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

Pour déterminer la droite perpendiculaire à une face du polyèdre, on choisit trois sommets $A_1, A_2, et A_3$ parmi les sommets qui déterminent la face ensuite on prend le point A_1 et le vecteur normal à la face suivant :

$$N_{face} = A_1A_2 \otimes A_1A_3, \quad \otimes : \text{est le produit vectoriel} \quad (\text{III-1})$$

Pour déterminer la droite perpendiculaire à deux segments $A_1B_1 et A_2B_2$ du polyèdre, qui est pour le cas général unique (sauf pour le cas où $A_1B_1 et A_2B_2$ sont parallèles), on calcule les deux points $x_1 et x_2$ de ces segments qui appartiennent à la droite perpendiculaire.

$$\begin{aligned} x_1 \in A_1B_1 & \quad \therefore A_1x_1 = t_1A_1B_1 \\ x_2 \in A_2B_2 & \quad \therefore A_2x_2 = t_2A_2B_2 \\ x_1x_2 \perp A_1B_1 & \quad \therefore x_1x_2 \cdot A_1B_1 = 0 \\ x_1x_2 \perp A_2B_2 & \quad \therefore x_1x_2 \cdot A_2B_2 = 0 \end{aligned} \quad (\text{III-2})$$

Ceci donne un système de huit équations avec comme inconnues les composantes des vecteurs $x_1 et x_2$ en plus des nombres $t_1 et t_2$. Si on dénote par $v_1 et v_2$ les vecteurs $A_1B_1 et A_2B_2$ respectivement, alors :

$$\underbrace{\begin{bmatrix} -I_{3 \times 3} & 0_{3 \times 3} & v_1 & 0_{3 \times 1} \\ 0_{3 \times 3} & -I_{3 \times 3} & 0_{3 \times 1} & v_2 \\ -v_1 & v_1 & 0 & 0 \\ -v_2 & v_2 & 0 & 0 \end{bmatrix}}_p \begin{bmatrix} x_1 \\ x_2 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} -A_1 \\ -A_2 \\ t_1 \\ t_2 \end{bmatrix} \quad (\text{III-3})$$

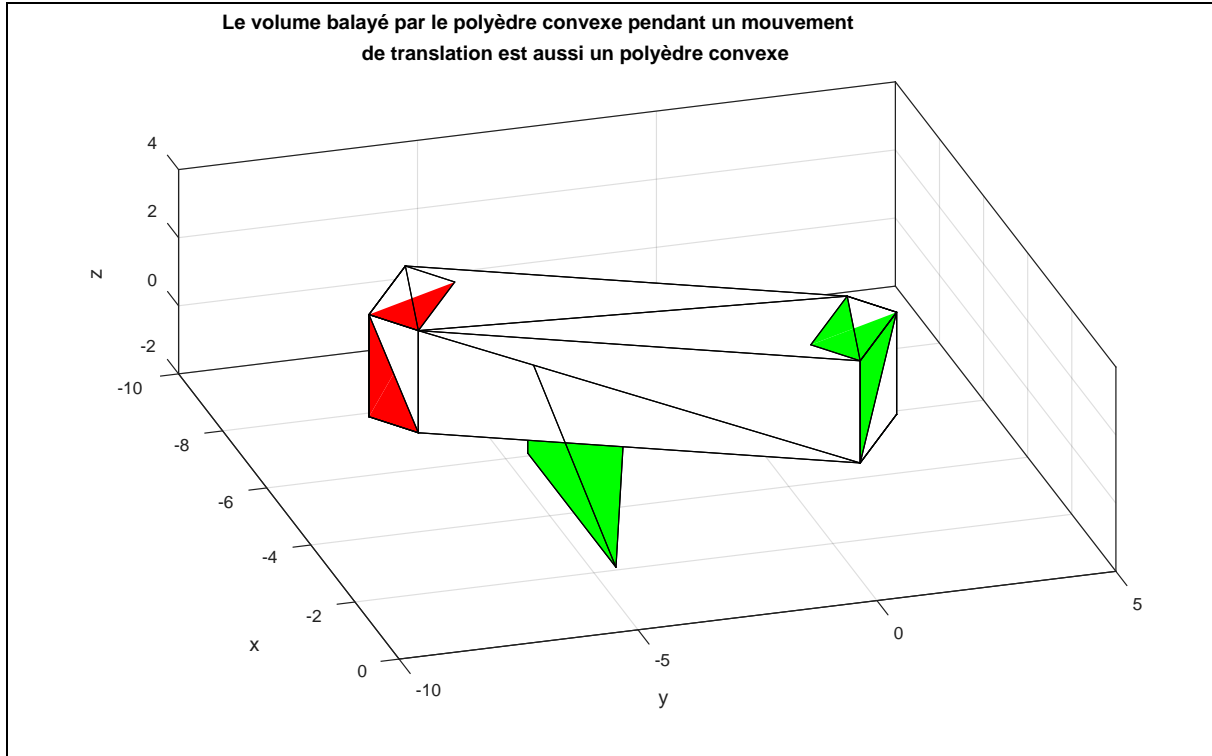
La matrice p est toujours non singulière sauf pour le cas particulier où $v_1 et v_2$ sont parallèles ou l'un d'eux est nul.

La résolution de ce système d'équations donne les points $x_1 et x_2$ et par suite la droite perpendiculaire.

L'une des propriétés importantes pour le test de collision concernant le mouvement de polyèdres est que le volume balayé par un polyèdre convexe durant un mouvement de translation est aussi un polyèdre convexe. Ce polyèdre balayé peut être obtenu en cherchant le polyèdre convexe formé par l'ensemble des sommets avant le mouvement et celui après le mouvement. Les positions des sommets du polyèdre après le mouvement sont obtenues en appliquant la transformation homogène aux coordonnées des sommets avant le mouvement ceci est due aux propriétés du solide.

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

Sous MATLAB, on peut obtenir le volume balayé en utilisant l'instruction « convhull ». La figure suivante représente un volume balayé durant un mouvement de translation.



III.3. Échantillonnage aléatoire de configurations :

Sans doute le choix de la stratégie d'échantillonnage peut jouer un rôle très important dans l'amélioration des performances de la PRM [10, 11]. Ceci a été observé dans les premières publications sur la PRM qui ont suggéré des mécanismes pour la génération de configurations selon une distribution non uniforme. L'augmentation de la densité d'échantillonnage dans une région de l'espace hors collision s'appelle « Importance Sampling ».

L'échantillonnage uniforme (selon une distribution uniforme) est la technique la plus simple à implémenter. Il a l'avantage que les cas particulièrement difficiles ne peuvent pas vaincre le planificateur. Cependant, l'inconvénient est que le temps d'exécution peut changer beaucoup entre différentes requêtes. Néanmoins, l'échantillonnage uniforme a donné des résultats très satisfaisants pour beaucoup de cas impliquant des robots avec un nombre élevé de degrés de liberté.

Il y a des cas où l'échantillonnage uniforme a des performances très réduites. Souvent, ceci est dû à la présence de passages étroits à travers lesquels il est nécessaire de passer pour répondre

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

à des requêtes de planification. Différentes techniques d'échantillonnage existent pour ce genre de problème : échantillonnage près des obstacles, échantillonnage à l'intérieur des passages étroits, échantillonnage basé sur la visibilité...

III.3.1. Choix aléatoire d'une configuration selon une distribution uniforme :

L'échantillonnage de la position selon une distribution uniforme est facile à accomplir, car il suffit de choisir les trois coordonnées x , y et z selon une distribution uniforme chacune. Dans MATLAB, ceci est réalisé en utilisant l'instruction `rand()`.

Le cas à trois dimensions avec un corps solide implique en plus l'orientation. L'orientation peut être représentée de différentes façons (matrice de rotation, quaternion, Angles d'Euler, Axe-angle...). Dans ce cas, l'échantillonnage de la position se fait toujours de la même façon. Cependant, pour l'orientation, il a été montré que le choix selon une distribution uniforme de certains paramètres de l'orientation ensuite l'obtention des autres par le calcul n'obéit pas toujours à une distribution uniforme.

Ainsi, pour l'échantillonnage d'une rotation selon une distribution uniforme, nous avons adopté la méthode de [19]. On commence par choisir 3 nombres selon une distribution uniforme ξ_0 , ξ_1 et ξ_2 . A partir de ces trois nombres on calcul le quaternion qui représente cette rotation comme suit :

$$Q = \begin{bmatrix} \sqrt{1-\xi_0} \sin(2\pi\xi_1) & \sqrt{1-\xi_0} \cos(2\pi\xi_1) & \sqrt{\xi_0} \sin(2\pi\xi_2) & \sqrt{\xi_0} \cos(2\pi\xi_2) \end{bmatrix}^T = [w \ x \ y \ z]^T \text{ avec } w^2 + x^2 + y^2 + z^2 = 1 \quad (\text{III-4})$$

La représentation Axe/Angle équivalente est liée à ce quaternion comme suit :

Si l'axe de rotation qui est unitaire est : $K = [k_x \ k_y \ k_z]$ et l'angle de rotation est θ on a [8] :

$$K = \frac{[x \ y \ z]}{\sqrt{x^2+y^2+z^2}}; \quad \theta = 2 * \arccos(w) \quad (\text{III-5})$$

On peut aussi faire l'opération inverse. À partir de la représentation axe-angle, on peut obtenir le quaternion équivalent comme suit [8] :

$$w = \cos\left(\frac{\theta}{2}\right), \quad x = k_x \sin\left(\frac{\theta}{2}\right), \quad y = k_y \sin\left(\frac{\theta}{2}\right), \quad z = k_z \sin\left(\frac{\theta}{2}\right), \quad (\text{III-6})$$

À partir du quaternion unitaire ainsi obtenu, on peut calculer la matrice de rotation comme suit [8] :

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

$$R = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2(xy - zw) & 2(xz + yw) \\ 2(xy + zw) & 1 - 2x^2 - 2z^2 & 2(yz - xw) \\ 2(xz - yw) & 2(yz + xw) & 1 - 2x^2 - 2y^2 \end{bmatrix} \quad (\text{III-7})$$

La transformation inverse d'une matrice de rotation vers un quaternion est donnée par [8] :

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}; \quad w = \frac{1}{2} \sqrt{1 + r_{11} + r_{22} + r_{33}}; \quad \begin{cases} x = \frac{r_{32} - r_{23}}{4w} \\ y = \frac{r_{13} - r_{31}}{4w} \\ z = \frac{r_{21} - r_{12}}{4w} \end{cases} \quad (\text{III-8})$$

Pour obtenir la matrice de rotation à partir de la représentation axe-angle, on applique [8] :

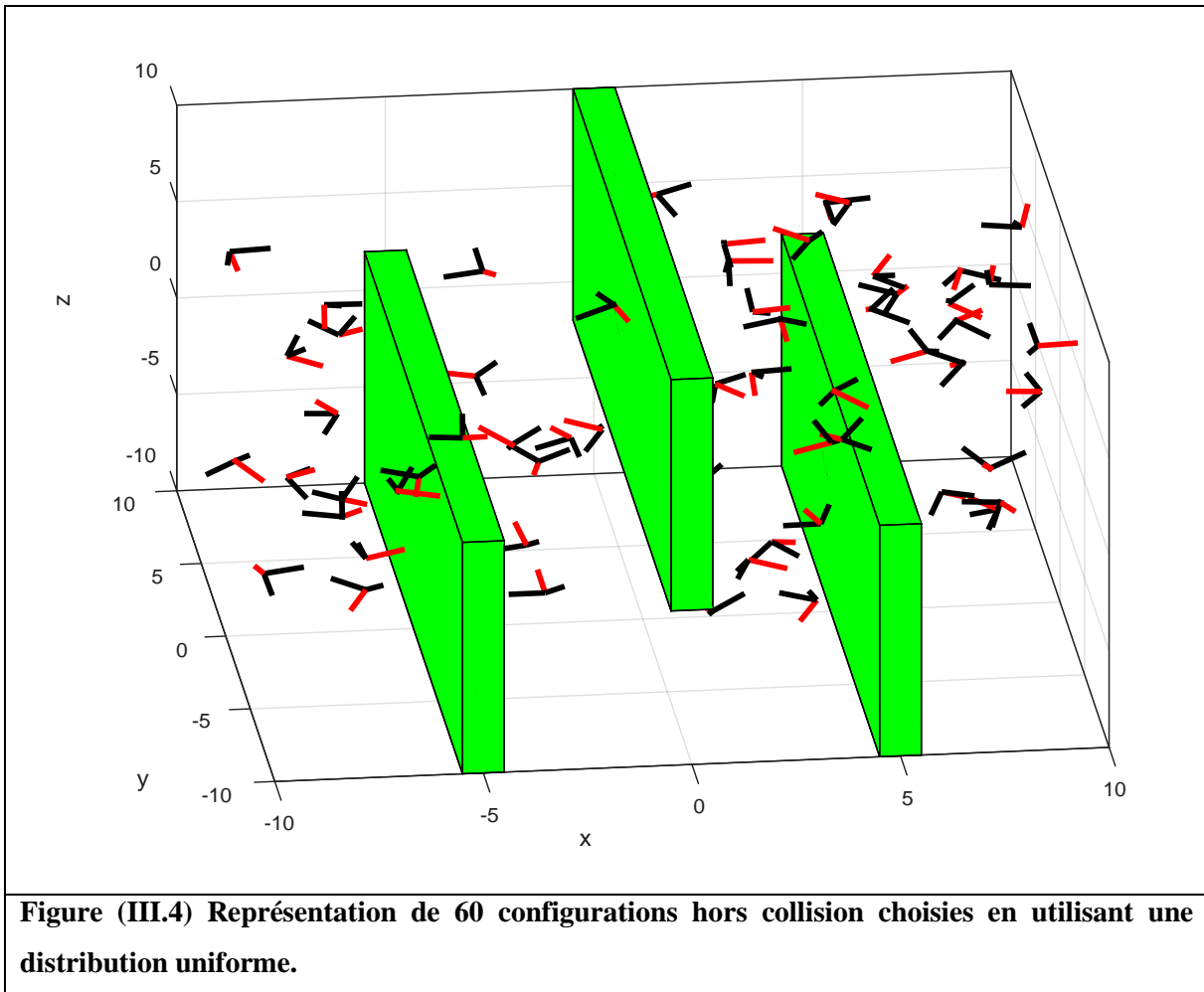
$$R_K(\theta) = \begin{bmatrix} k_x k_x v(\theta) + c\theta & k_x k_y v(\theta) - k_z s\theta & k_x k_z v(\theta) + k_y c\theta \\ k_x k_y v(\theta) + k_z s\theta & k_y k_y v(\theta) + c\theta & k_y k_z v(\theta) + k_x c\theta \\ k_x k_z v(\theta) - k_y s\theta & k_y k_z v(\theta) + k_x s\theta & k_z k_z v(\theta) + c\theta \end{bmatrix} \text{ avec } \begin{cases} c\theta = \cos(\theta) \\ s\theta = \sin(\theta) \\ v\theta = 1 - \cos(\theta) \\ K = [k_x \quad k_y \quad k_z] \end{cases} \quad (\text{III-9})$$

9)

L'opération inverse est obtenue comme suit [8] :

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \rightarrow \theta = \arccos\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right); \quad K = \frac{1}{2 \sin(\theta)} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \quad (\text{III-10})$$

Dans ce travail, nous avons représenté chaque configuration de la carte de route probabiliste par les trois coordonnées de l'origine du repère attaché au polyèdre représentant le robot. Pour l'orientation, nous avons utilisé les 4 composantes du quaternion représentant l'orientation du repère attaché au robot par rapport au repère de la station. Cette représentation rajoute un nombre supplémentaire, mais elle simplifie aussi les calculs. Chaque configuration est identifiée par son numéro. Un nombre entier est utilisé pour spécifier le nombre de nœuds auxquelles cette configuration est connectée par un chemin hors collision. La figure suivante représente 60 configurations choisies de façon aléatoire en utilisant une distribution uniforme. Sur le graphe chaque configuration est représentée par le repère attaché au robot. L'axe x de ce repère est représenté en rouge.



Il faut noter que le choix de configurations selon une distribution uniforme quoiqu'il forme une méthode puissante pour la construction de la PRM (Il arrive toujours à capturer la connectivité de l'espace hors collision si on considère un nombre suffisant de configurations) [11,10], ils existent des cas pathologiques où ceci devient très difficile à garantir. Un exemple est le cas des passages étroits. Si un passage étroit existe et qu'il est nécessaire de passer par ce passage pour répondre à des requêtes, alors des configurations doivent être choisies dans une région très limitée. Différentes techniques existent pour accommoder ce problème. Dans ce travail nous avons programmé deux techniques.

La première est appelée échantillonnage basé sur les obstacles (OBB : Obstacle Based sampling) [11]. Elle consiste à choisir aléatoirement selon une distribution uniforme une configuration q_{in} . Si q_{in} est hors collision, elle est abandonnée. Sinon, si q_{in} est en collision alors une direction aléatoire est générée et le planificateur local est utilisé pour trouver le long

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

de cette direction une configuration q_{out} qui est hors collision. Ensuite une méthode de bisection est utilisée pour rapprocher q_{out} près d'obstacle. La configuration obtenue après cette procédure est rajoutée à la carte et q_{in} q_{out} sont abandonnées.

La deuxième méthode que nous avons aussi programmée s'appelle méthode par pont [11]. Elle consiste à choisir aléatoirement selon une distribution uniforme deux configurations q_1, q_2 qui sont en collision. La configuration centrale est vérifiée pour collision et si elle est hors collision elle est rajoutée à la carte. Pour le cas où le robot est un corps solide, la configuration contient la position et l'orientation, ainsi le calcul de configurations centrales ne sera pas évident. Pour les trois coordonnées de la position, il suffit de prendre la moyenne entre les deux. Pour les quatre coordonnées du quaternion représentant l'orientation, nous avons utilisé l'algorithme de [Kuffner] pour les calculs.

III.4. Connexion de la PRM [10] :

Un autre aspect important dans la construction de la PRM, est celui de la sélection des paires de configurations pour lesquelles un chemin hors collision les connectant sera recherché en utilisant un planificateur local. L'objectif est de choisir les configurations pour lesquelles le planificateur local a plus de chance de réussir. Une façon de faire est d'utiliser le planificateur local pour connecter chaque configuration aux k configurations qui lui sont les plus proches (k nearest neighbors). La logique derrière ceci est que les configurations voisines conduisent à des connexions courtes qui ont plus de chances d'être hors collision.

III.4.1. Choix des k voisins les plus proches (k -nearest neighbors) :

Le problème que nous allons traiter ici est comment choisir les configurations voisines. Pour le cas d'un robot ponctuel où l'orientation n'est pas prise en considération, il suffit de prendre la distance euclidienne entre deux positions comme une fonction distance qui permettra de décider quelles sont les configurations qu'il faut considérer. Pour généraliser ceci au cas où le robot est un corps solide (représenté ici par un polyèdre), la fonction distance (qu'on appelle aussi métrique) doit être choisie de telle sorte qu'elle reflète la probabilité que le planificateur local échoue à trouver un chemin hors collision entre les deux configurations. Une solution alternative est de partager la fonction distance en deux parties. Une partie concerne la distance entre les deux configurations qui est due à la translation et la deuxième concerne la différence dans l'orientation. Par exemple si le vecteur X représente la translation entre deux

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

configurations et la matrice R représente la différence dans l'orientation $((X, R) \in SE(3))$, alors [18] :

$$dist(q', q'') = w_t \|X' - X''\| + w_r f(R', R'') \quad (III-11)$$

Est une métrique pondérée avec comme composante de translation $\|X' - X''\|$ qui est la norme euclidienne standard et une fonction scalaire positive qui donne typiquement une mesure approximative de la distance entre les deux rotations dans $R', R'' \in SO(3)$. La rotation est mise à l'échelle par rapport à la translation en utilisant les poids w_t et w_r . Une bonne sélection de la fonction $f(R', R'')$ est la longueur de la courbe géodésique [le plus grand arc sur le cercle unitaire dans l'espace 4D entre deux quaternions] entre $R', R'' \in SO(3)$.

Une difficulté avec cette approche réside dans la sélection des poids. De plus, son extension aux cas de systèmes articulés n'est pas facile. Quelques choix pour la fonction $f(R', R'')$ sont :

Pour le d'angle d'Euler $(\theta_1, \phi_1, \eta_1), (\theta_2, \phi_2, \eta_2)$:

$$f(R', R'') = \rho_r = w_r \sqrt{\Delta(\theta_1, \theta_2)^2 + \Delta(\phi_1, \phi_2)^2 + \Delta(\eta_1, \eta_2)^2} \quad (III-12)$$

$$\text{Avec} \quad \Delta(\theta_1, \theta_2) = \begin{cases} \theta_2 - \theta_1 + 2\pi & \text{si } (\theta_2 - \theta_1) < -\pi \\ \theta_2 - \theta_1 & \text{si } -\pi < (\theta_2 - \theta_1) < \pi \\ \theta_2 - \theta_1 - 2\pi & \text{si } (\theta_2 - \theta_1) > \pi \end{cases} \quad (III-13)$$

La même expression est utilisée pour $\Delta(\phi_1, \phi_2), \Delta(\eta_1, \eta_2)$.

Pour le cas où l'orientation est représentée en utilisant des quaternions Q_1, Q_2 [18] :

$$f(R', R'') = \rho_r = w_r (1 - \|\lambda\|) \quad \text{ici } 0 \leq \rho_r \leq w_r \quad (III-15)$$

Avec λ obtenu à partir du produit scalaire des deux quaternions $\lambda = Q_1 \cdot Q_2$. Le cas où deux quaternions représentent la même rotation est pris en charge ici en prenant la valeur absolue $\|\lambda\|$ car la différence est seulement dans le signe de λ .

III.4.2. Planificateur local [11] :

Pour déterminer la trajectoire [hors collision] à prendre entre deux configurations voisines q_i, q_j de la carte de route, nous utilisons un planificateur local. Ce dernier doit être simple et rapide pour minimiser le temps de calcul, car c'est la partie qui consomme le plus de temps de calcul. Ceci est dû au fait que le planificateur local doit tester toutes les connexions des nœuds voisins pour collision et ceci même pour les connexions non valides [en collision]. Le planificateur

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

local doit retourner un résultat vrai si une telle trajectoire hors collision existe ou faux dans le cas contraire. Il doit aussi être déterministe, car la trajectoire ne sera pas enregistrée. Plusieurs planificateurs locaux existent parmi lesquels on peut citer les planificateurs à temps optimal avec leurs variantes, les planificateurs basés sur des champs de potentiels [attractif vers la destination et répulsif des obstacles], segment linéaire ($q = q_i + (q_j - q_i)t$ avec $t \in [0,1]$), rotate-at-s qui sera expliqué en détail après. Le test de collision durant ce déplacement peut se faire de différentes façons.

III.4.2.1. Méthode incrémentale :

Dans cette méthode, on fait des petits pas le long de la trajectoire entre q_i et q_j et on vérifie la configuration obtenue après chaque pas pour collision. Ainsi, à partir d'une configuration q_i on se déplace vers l'une des configurations directement voisines. Par exemple, pour un robot ponctuel dans un espace 3D, les configurations directement voisines de q_j sont $q_j + \{(-\varepsilon, 0, \varepsilon)^3 - \{(0,0,0)\}\}$. Une définition analogue pour un robot manipulateur à n degrés de liberté est $q_j + \{(-\varepsilon, 0, \varepsilon)^n - \{(0,0, \dots, 0)\}\}$. Une façon de choisir le pas ε est de le choisir en fonction du déplacement maximal parmi les déplacements de tous les points du robot pendant le mouvement élémentaire, ou bien choisir ε de façon indépendante ensuite amplifier le volume du robot par un rapport ε .

III.4.2.2. Méthode binaire :

La logique derrière cette méthode est que : vu que les configurations q_i et q_j sont hors collision alors il en serait de même pour les configurations qui leur sont proches et par conséquent la configuration située au milieu a plus de chance à être en collision. Ainsi, on commence par tester cette configuration contre la collision, et si elle l'est, il n'y a pas de chemin entre la paire q_i, q_j et l'on aura rapidement trouvé la réponse. Si la configuration du milieu est hors collision, alors on teste les deux configurations situées au milieu des deux moitiés contre la collision selon la même logique. On continue de cette façon en testant à chaque fois 2^n Configurations jusqu'à ce qu'une collision soit détectée où que l'on atteigne une certaine tolérance concernant le pas.

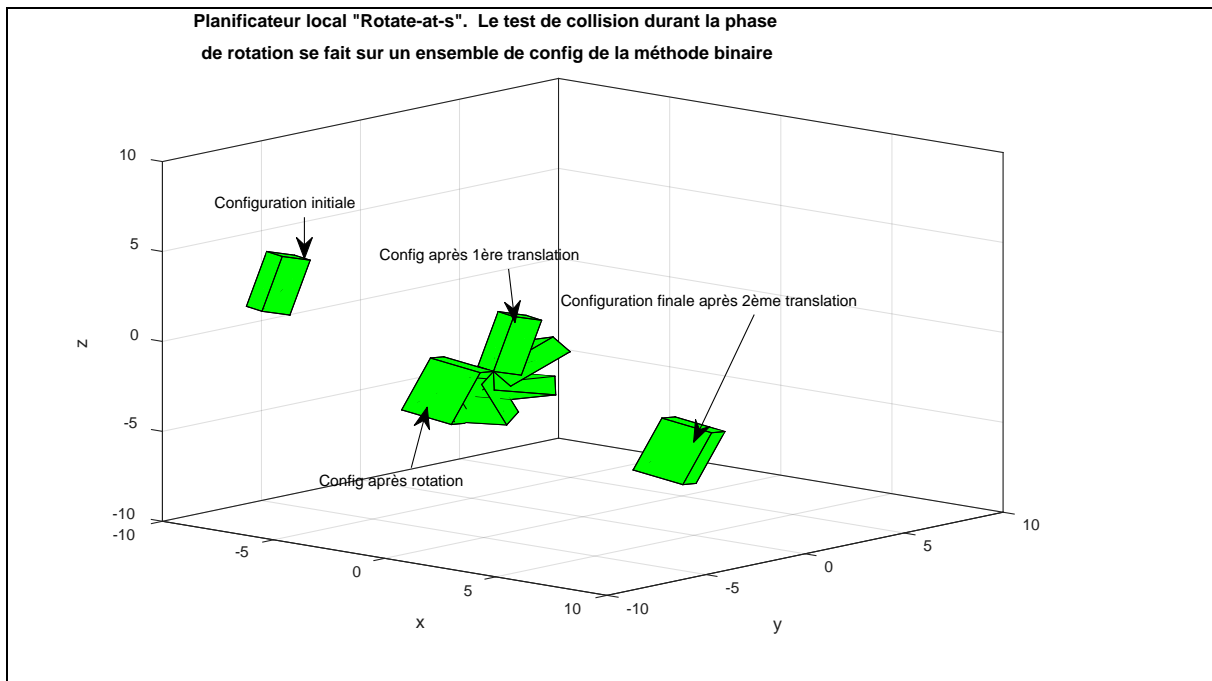
III.4.2.3. Rotation à s (rotate-at-s):

Ce planificateur commence par un mouvement de translation jusqu'à un certain point entre q_i et q_j généralement le point milieu, ensuite, il fait un mouvement de rotation (correspondant à la différence d'orientation entre les configurations q_i et q_j) et enfin un mouvement de

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

translation pour atteindre q_j . Dans ce travail, nous avons utilisé ce planificateur à cause de sa simplicité et, car il est adapté à la supposition que nous avons fait pour le robot. Pour le test de collision, nous avons procédé comme suit :

Pour les mouvements de translation, nous avons considéré le volume convexe balayé par le robot durant ce mouvement de translation qui est lui aussi un polyèdre. Ce dernier doit être hors collision. En utilisant la procédure de test de collision entre polyèdres que nous avons déjà introduite, on obtient un test exact. Le test du mouvement de rotation contre la collision a été fait en utilisant la méthode binaire. La figure suivante illustre ce planificateur.



Ayant choisi les k -nearest voisins d'une configuration q_i , une façon de faire est de tenter de connecter q_i à chacun des k configurations par le planificateur local. Cette technique produira une carte dense qui contiendra des cycles fermés. L'avantage ici est que les chemins obtenus durant la phase de requêtes sont presque optimaux. Malheureusement, la recherche dans ce cas devient très couteuse en termes de temps de calcul.

Une approche alternative est d'éviter de rajouter des connexions qui introduisent des cycles fermés dans le graphe vu que les deux configurations sont déjà connectées à travers un autre chemin. Ainsi l'ajout de l'arc qui ferme le cycle n'améliore pas la connectivité de la carte. Ceci

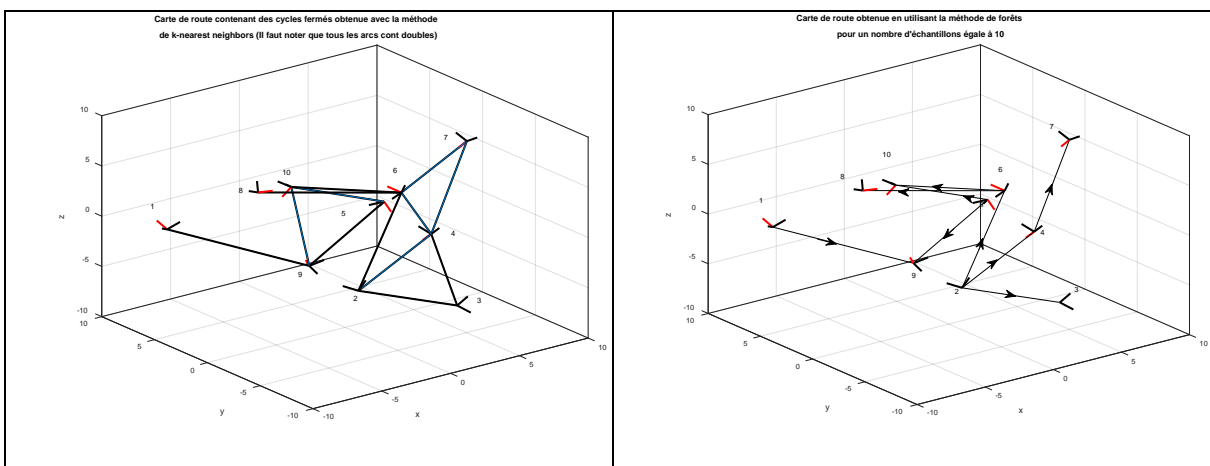
Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

suggère la connexion de la configuration q_i au plus une seule fois à chaque composante connectée du graphe G . La définition suivante donne une formulation plus précise [10].

$$N(c) = \{q \in (V - \{q_i\}) \mid [D(q_i, q) \leq \text{maxdist}] \wedge [\forall q' \in (V - \{q_i, q\}) : \text{connected}(q', q) \Rightarrow (D(q_i, q) < D(q_i, q'))]\}$$

Ainsi, dans chaque composante connectée du graphe G , le nœud le plus proche de q_i est un voisin s'il vérifie la condition $[D(q_i, q) \leq \text{maxdist}]$. A chaque fois que le planificateur local réussit à trouver un chemin entre deux configurations, les composantes connectées du graphe sont dynamiquement mises à jour. La méthode qui ajoute les arcs selon ce principe est appelée la méthode de forêt, car elle produit des graphes qui sont formés d'un ensemble d'arbres. L'inconvénient de cette technique est que les chemins obtenus durant la phase de requête sont parfois inutilement longs. Cet inconvénient peut être éliminé, en appliquant des méthodes de lissage soit durant la construction de la carte ou au chemin particulier obtenu en réponse à une requête de planification

Dans ce travail, nous avons programmé les deux méthodes de connexion. Nous donnons dans la figure qui suit les cartes obtenues en appliquant les deux méthodes au même ensemble de configurations. Il est clair que la méthode donnera un seul chemin en réponse à une requête de planification et que la méthode de forêt donnera plusieurs solutions (plusieurs choix disponibles).



Figure(III.6) : Cartes obtenues en utilisant deux méthodes de connexion : a) K-nearest neighbors ; b) Méthode de forêts.

En utilisant la procédure de recherche de chemin dans une carte de route (qui sera expliquée dans ce qui suit) pour obtenir un chemin entre les configurations 1 et 6 et en utilisant les deux

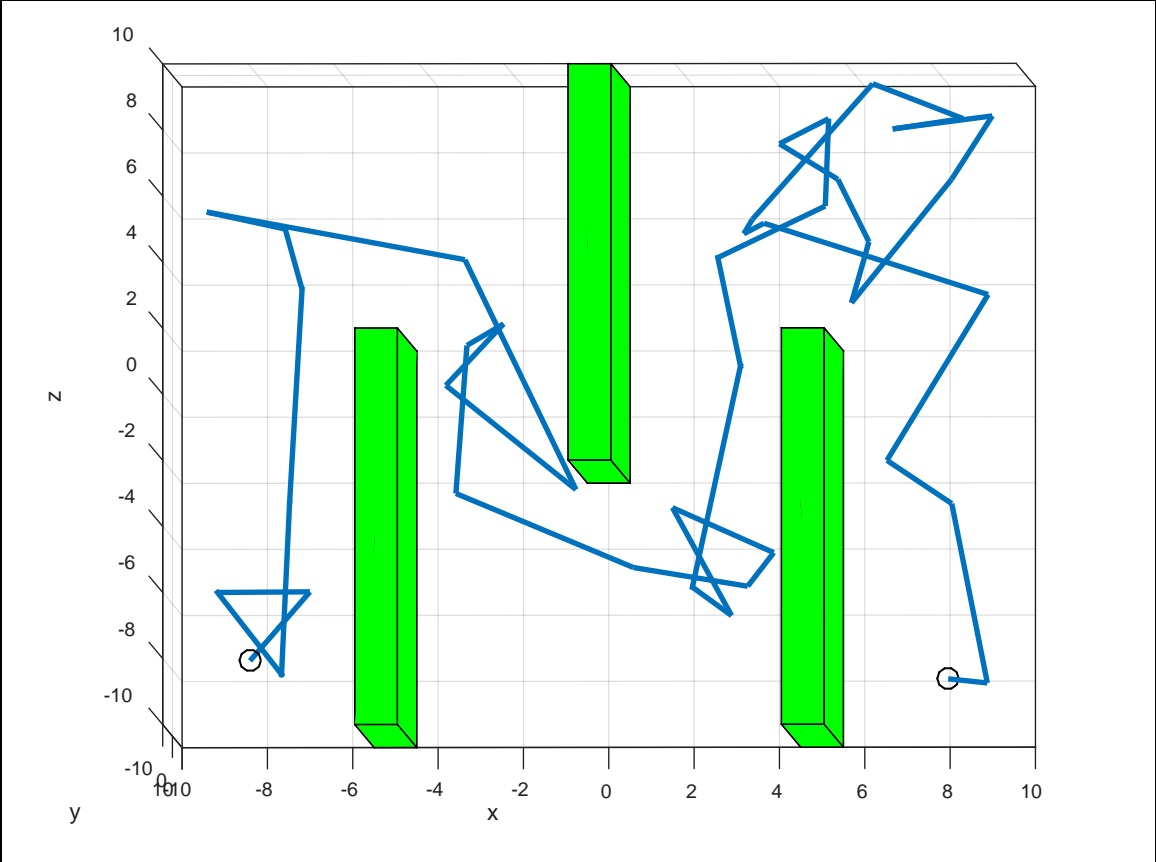
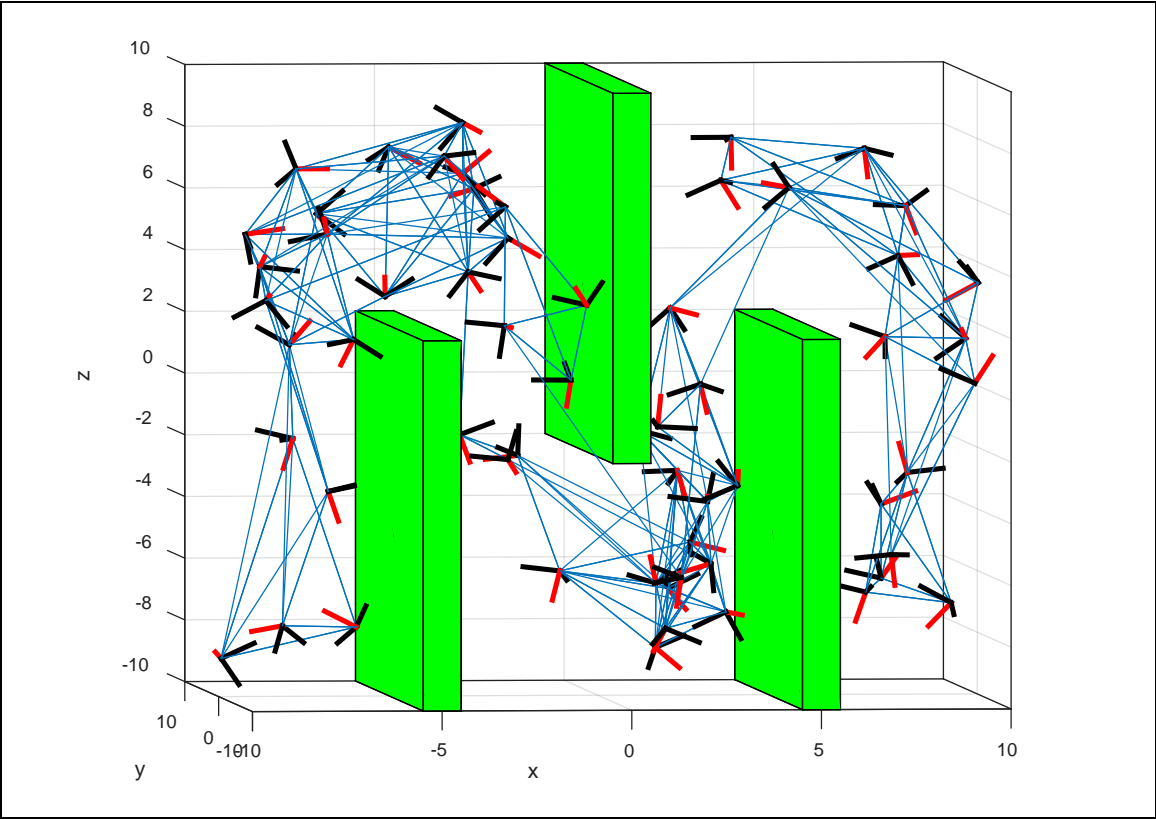
Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

cartes, nous avons obtenu les deux chemins $1 \rightarrow 9 \rightarrow 5 \rightarrow 10 \rightarrow 6$ pour la carte construite avec la méthode de k-nearest neighbors et le chemin unique (évidemment) $1 \rightarrow 9 \rightarrow 10 \rightarrow 6$ pour la carte construite avec la méthode de forêts. On remarque que le chemin $1 \rightarrow 9 \rightarrow 10 \rightarrow 6$ est le meilleur parmi les deux obtenus, mais il n'a pas été obtenu par la méthode de forêts. Il faut remarquer que pour obtenir un chemin à partir de la carte obtenue avec la méthode de forêt il a fallu rajouter les arcs de sens inverse à la carte.

Nous avons aussi effectué la recherche de chemins entre les configurations 1 et 3 dans les deux cartes. 12 solutions ont été obtenues à partir de la carte prenant en considération les cycles et 1 seul à partir de la carte sous forme d'arbre.

Pour montrer l'inconvénient de la méthode incluant des cycles fermés nous considérons l'exemple suivant qui consiste en une carte construite en utilisant la méthode de k-nearest neighbors (incluant des cycles fermés) et comportant 50 configurations choisies avec une distribution uniforme. Non avons fixé le nombre maximal des voisins les plus proches à 8 et la distance maximale à 40. Pour calculer les chemins possibles on s'est contenté des 50 premiers chemins obtenus parmi lesquels celui donné dans la figure. Il faut noter que le calcul de tous les chemins possibles est prohibitif (après plus de 45 min le programme n'a pas terminé sachant qu'il arrive à calculer 50 chemins de façon presque instantanée).

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques



Figure(III.7) : Réponse à une requête de planification en utilisant une carte construite en utilisant l'échantillonnage uniforme de configurations la méthode de k-nearest neighbors pour la connexion.

III.5. Phase de requêtes :

Durant la phase de requête (exploitation de la PRM), on donne une configuration initiale q_{init} et une configuration cible q_{goal} , ensuite on connecte ces deux configurations à la carte à travers leurs k-configurations voisines qui vérifient la condition sur la distance et enfin on fait une recherche dans le graphe pour un chemin qui lie les deux configurations. Différentes techniques de recherche dans un graphe existent. Dans ce travail, nous avons utilisé la recherche en profondeur (depth first search). Cette technique peut être facilement expliquée sur un graphe en arbre. On commence par le nœud de racine (qui est ici q_{init}), ensuite un de ses descendants directs. À partir de ce dernier, on prend aussi un descendant et l'on continue de cette façon jusqu'à ce qu'on arrive soit au nœud cible q_{goal} ou à un nœud sans descendant (feuille). Si on rencontre un nœud sans descendants (feuille), alors on revient vers l'arrière d'un niveau (le nœud parent) et on choisit un autre descendant non visité à travers duquel on continue la recherche jusqu'à ce qu'on arrive à q_{goal} ou à un nœud sans descendant. Ce processus est répété jusqu'à ce qu'on arrive à q_{goal} ou que tous les nœuds du graphe soient visités.

A titre d'illustration, nous avons appliqué l'algorithme de recherche de chemin au graphe simple de la figure suivante pour trouver les chemins entre les nœuds 1 et 6 et nous avons obtenu les résultats associés. Il faut noter que tous les arcs sont doubles (dans les deux sens). Ainsi, le programme donne tous les chemins possibles entre la configuration initiale (nœud 1) et la configuration finale (nœud 6) ce qui permet de choisir la meilleure solution parmi les différentes possibilités. Il faut remarquer que le fait que le graphe contient des cycles fermés a conduit à plusieurs solutions ce qui ne serait pas le cas si le graphe était un arbre ou une forêt.

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

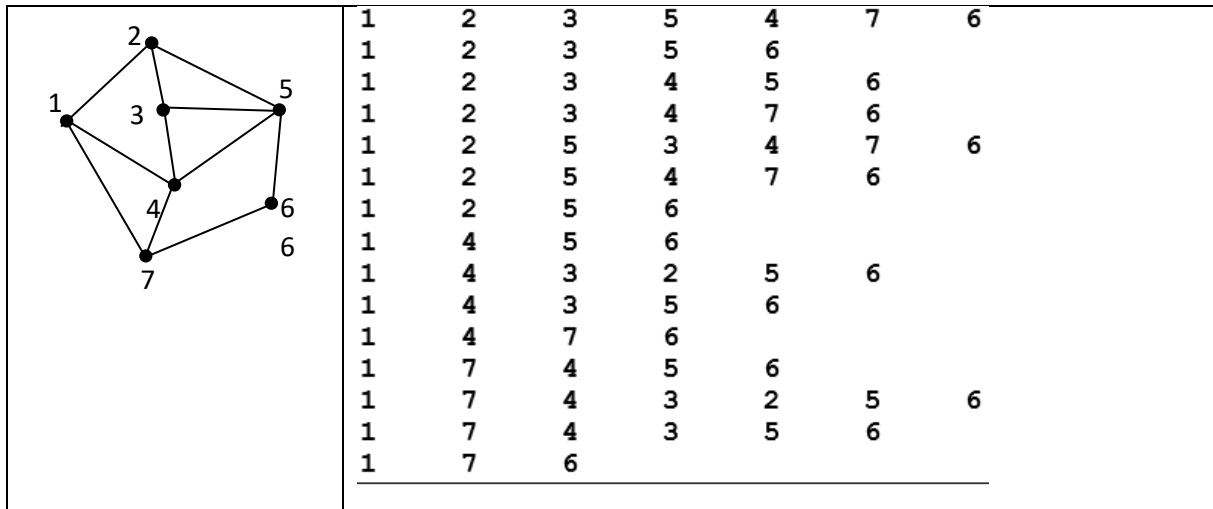


Figure (III.8) : Recherche des chemins entre deux nœuds (1 et 6) dans un graphe. A) Le graphe ; B) L'ensemble des chemins possibles (Résultat de recherche). Le graphe est spécifié par l'ensemble des arcs :

E' = 1 1 1 2 2 2 3 3 3 4 4 4 4 5 5 5 5 6 6 7 7 7
 2 4 7 3 5 1 2 5 4 1 5 3 7 2 3 4 6 5 7 1 4 6

La même technique peut être utilisée pour convertir un graphe contenant des cycles à une représentation sous forme d'arbre. La figure qui suit représente l'arbre obtenu à partir du graphe précédent.

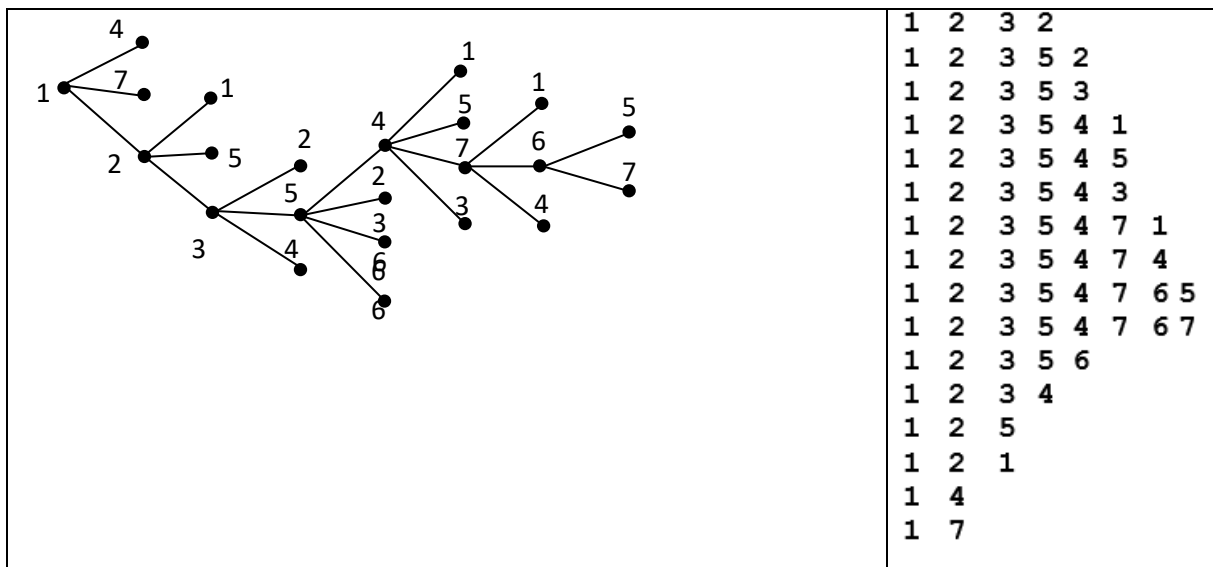


Figure (III.9) : Extraction d'un arbre à partir d'un graphe contenant des cycles fermés

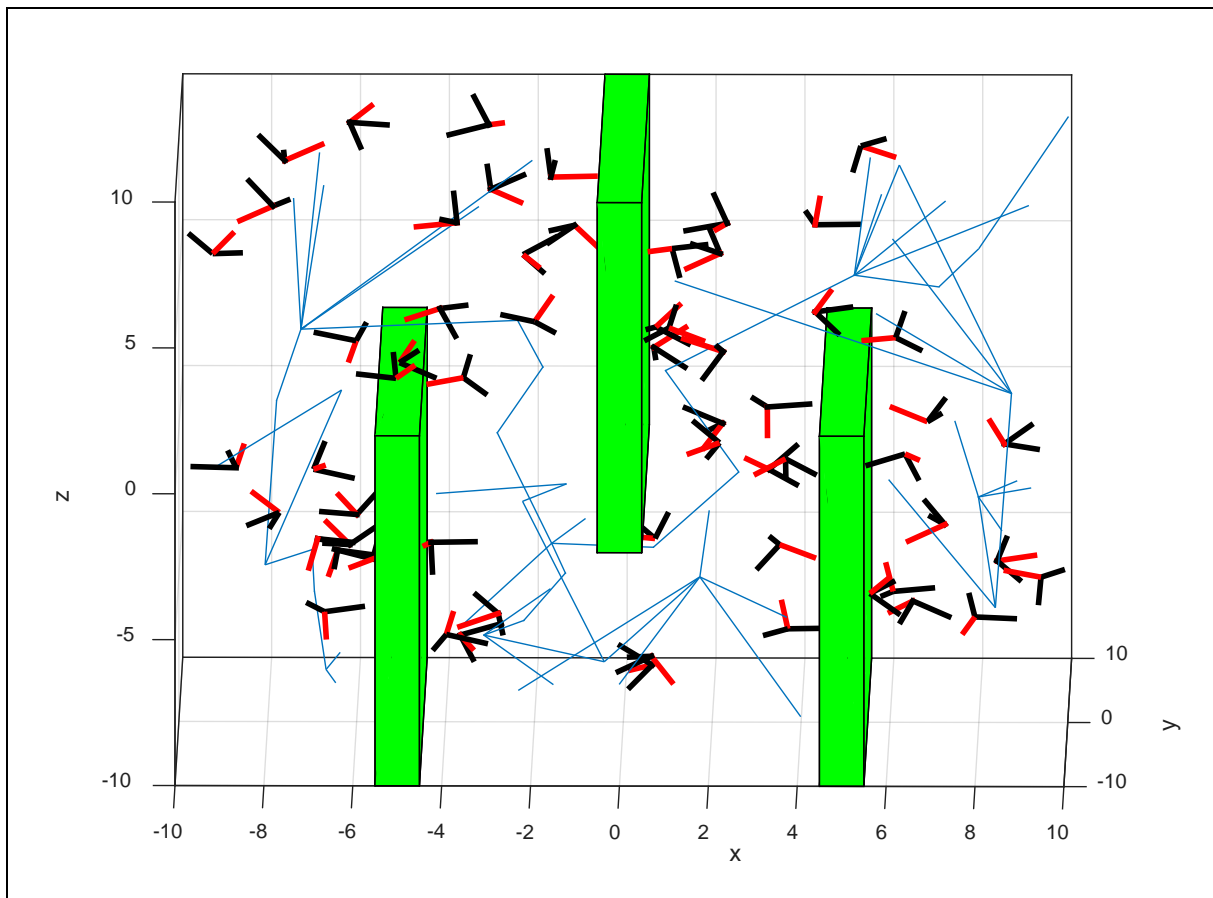
Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

Dans l'exemple qui suit, nous avons construit la carte en utilisant la méthode de pont pour le choix des configurations et la méthode de forêt pour la connexion. Nous avons aussi utilisé 50 configurations, avec un nombre maximum de voisins égal à 8. Pour la distance maximale entre voisins, nous avons choisi 40. La figure représente aussi la réponse à une requête de planification qui a comme configuration initiale q_{38} et comme configuration finale q_{50} .

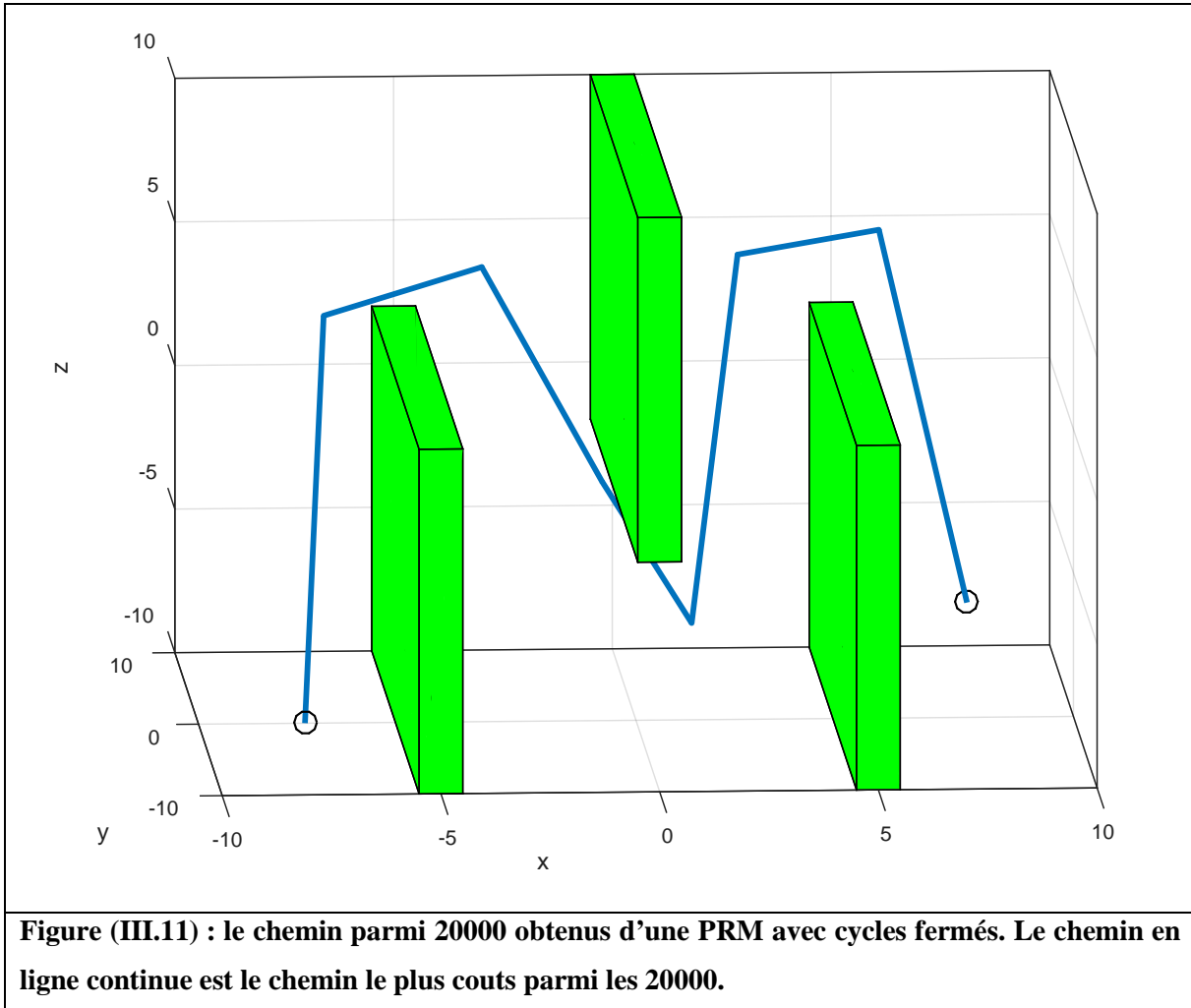
Si on examine le chemin obtenu, on remarque qu'il contient beaucoup de coin avec des angles aigus. C'est d'ailleurs une propriété des chemins qu'on obtient à partir d'une PRM. Pour améliorer les chemins obtenus à partir d'une PRM, une étape supplémentaire après le calcul de la réponse à une requête de planification s'impose.

III.6. Traitement postérieur [11, 10] :

Dans cette étape certaines opérations sont appliquées sur le chemin obtenu entre q_{init} et q_{goal} . Ces opérations améliorent la qualité du chemin obtenu selon certains critères (minimisation de distance, lissage...).



Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques



Au lieu de chercher le chemin le plus court, un autre critère peut être celui de chercher un chemin avec courbure lisse. Une approche possible est d'utiliser une courbe d'interpolation (exemple splines) ensuite prendre les configurations obtenues de la PRM comme des points d'interpolation de la courbe. Dans ce cas, le test de collision est effectué le long de la courbe, jusqu'à ce qu'une courbe qui remplit la propriété de lissage et l'évitement d'obstacle soit trouvée.

Nous avons implémenté l'approche gourmande dans notre programme et nous l'avons appliqué à un cas avec passage étroit. Nous avons obtenu 20000 chemins possibles entre q_{init} et q_{goal} . Nous avons pris le chemin le plus court entre eux ensuite nous lui avons appliqué l'opération d'élagage. La figure suivante montre le résultat obtenu. On remarque que le chemin a été nettement amélioré par élagage de point de vue distance et de point de vue nombre de points aux coins avec angle aigu.

Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

A titre de comparaison, nous avons traité le même cas en utilisant une PRM construite en utilisant une distribution uniforme seulement. On remarque que la PRM ne contient aucune configuration à l'intérieur du passage étroit et la réponse à la requête de planification été négative, c'est-à-dire qu'aucun chemin n'a été trouvé.

Dans cet exemple nous avons utilisé les mêmes valeurs pour les paramètres de la PRM qui sont :

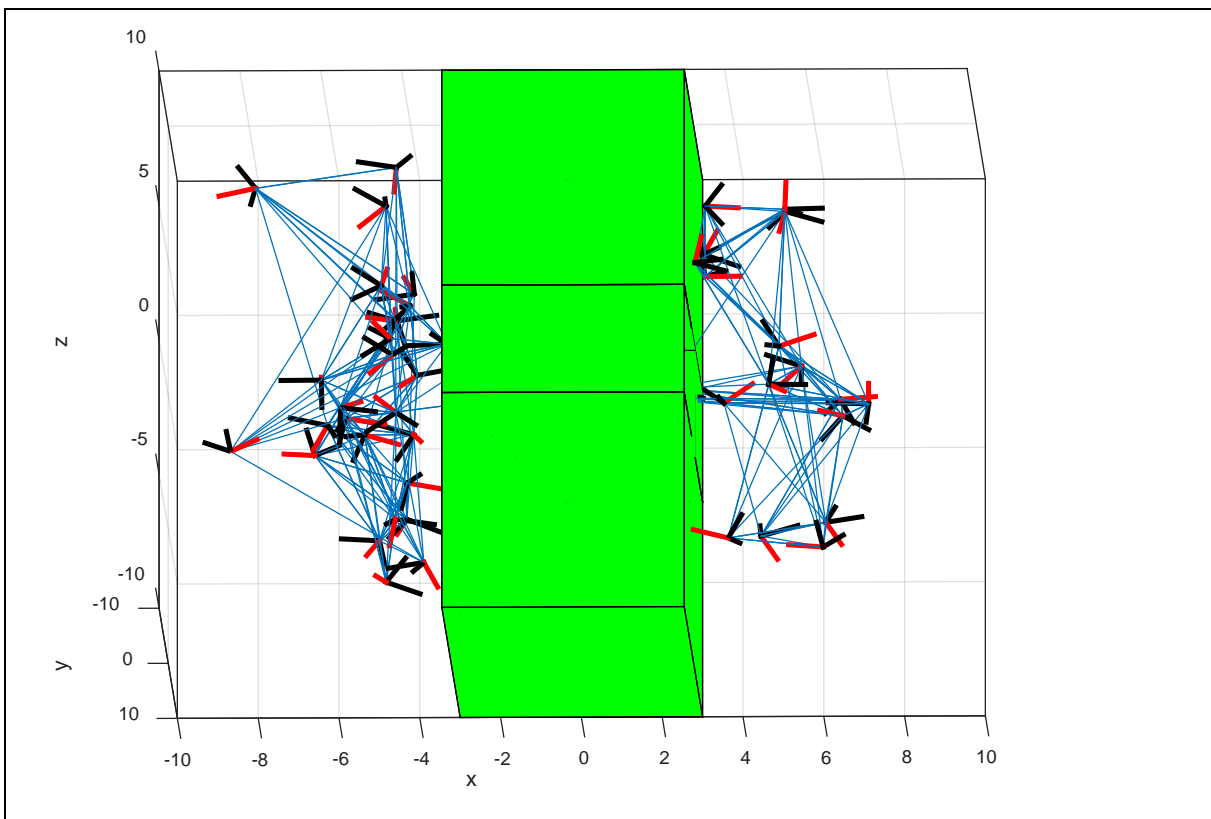
Nombre de configurations = 50

Nombre maximal de voisins = 8

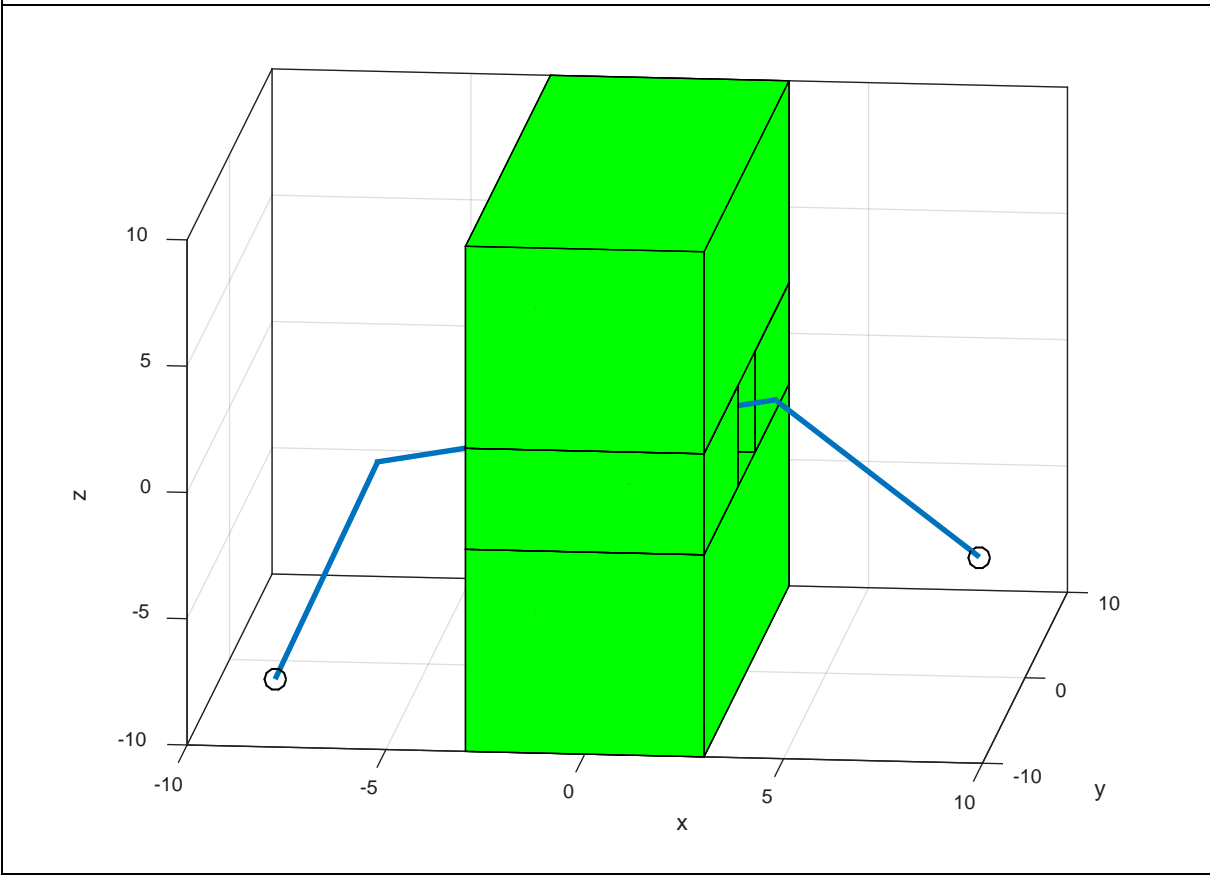
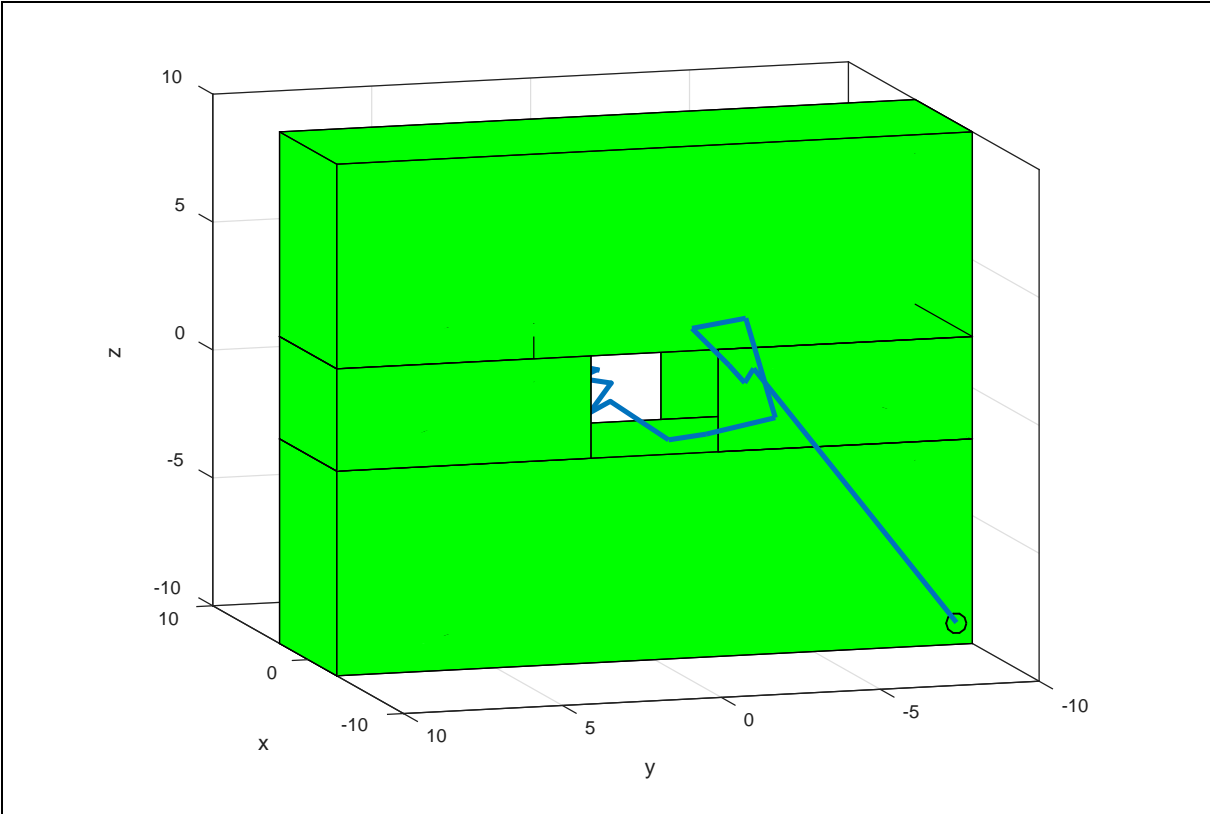
Distance maximale considérée en formant les k voisins = 40

Nombre de niveaux considérés pour le test de mouvement de rotation contre la collision = 6

Les poids de la rotation et de la translation dans la métrique : $w_t = 0.8$, $w_r = 0.2$.



Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques



Chapitre III : PRM pour la planification de mouvements d'un corps solide de forme polyédrique dans une scène contenant des obstacles polyédriques

Figure (III.12) : Utilisation du test du pont pour l'échantillonnage à l'intérieur de passage étroit
a) La PRM obtenue en utilisant le test de pont avec échantillonnage uniforme et une connexion avec des cycles fermés. b) Chemin obtenu à partir de la PRM et son traitement par élagage

Conclusion :

Dans ce chapitre, nous avons appliqué les différentes techniques de construction et d'exploitation d'une PRM. Nous avons programmé sous MATLAB les différents outils inhérents à l'utilisation de PRM (échantillonnage, connexion, recherche dans un graphe...). Nous avons fait des applications à des cas de planification en considérant que l'objet en mouvement est un corps solide sous forme polyédrique. Ceci impose de gérer en plus de la position l'orientation du corps. Nous avons simulé différentes techniques concernant l'échantillonnage, la connexion des configurations, la recherche dans un graphe et le traitement postérieur. Nous avons montré que pour certains cas pathologiques (passages étroits) la méthode de l'échantillonnage et de la connexion de la PRM doit être dotée de procédures spéciales pour garantir une réponse positive aux requêtes de planification.

Conclusion générale

Conclusion générale :

La planification de mouvement est le problème de la robotique où il est un élément fondamental des robots autonomes. La résolution de ce problème réside dans la construction d'une carte de route probabiliste (PRM Probabilistic Roadmap Method), qui a fait l'objet de notre travail.

Pour ce faire, nous avons considéré un corps solide sous forme polyédrique convexe se déplaçant dans une scène remplie d'obstacles.

Dans la construction de notre PRM, nous avons utilisé différentes techniques concernant l'échantillonnage, la connexion des configurations, la recherche dans un graphe et le traitement postérieur.

Le planificateur de la carte de route probabiliste a démontré le potentiel énorme des méthodes basées sur l'échantillonnage [11], cependant dans certains cas comme le cas des passages étroits la méthode de l'échantillonnage et de la connexion de la PRM doivent être dotées de procédures spéciales pour garantir une réponse positive aux requêtes de planification.

Malgré, la simplicité de l'implémentation de la PRM, elle est capable de traiter des robots avec de nombreux degrés de liberté et avec de nombreuses contraintes différentes.

References bibliographies:

[1] T. H. Kurfess, Robotics and Automation Handbook, CRC Press, 2005.

[2] <https://perso.liris.cnrs.fr/laetitia.matignon/index/coursL1robotique.pdf>

[3] BELAID Amine. Planification de chemin par carte de route probabiliste : Cas d'une scène à deux dimensions et robots ponctuel ou polygonal. Automatique. Lieu de soutenance : Université ABDREHMANE Mira de Bejaia, 2015/2016, 59 pages.

[4] Richard P. Paul, Robot Manipulators, MIT Press, 1981.

[5] Chasles, M., Note sur les propriétés générales du système de deux corps semblables entre eux et places d'une manière quelconque dans l'espace ; et sur le déplacement fini ou infiniment petit d'un corps solides libre. Férussac, Bulletin des Sciences Mathématiques, 14, 321-326, 1830.

[6] E. D. W. KHALIL, Modélisation, identification et commande des robots, Hermès, 1999.

[7] <http://www.irccyn.ec-nantes.fr/~martinet/MoCom/inversekinematicsinfench.pdf>

[8] John J. Craig, "Introduction to Robotics, Mechanics and Control" 3rd edition, Prentice Hall 2005.

[9] RANTANEN Mika. Improving Probabilistic Roadmap Methods for Fast Motion Planning. Tampere, 2014, 163.

[10] L. E. Kavraki, P. Svestska, J. C. Latombe, M. H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces", IEEE Trans on Robotics and Automation, 12 (4) pp 566-580, June 1996.

[11] Howie Choset, Kevin Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia Kavraki, Sebastian Thrun, "Principles of Robot Motion, Theory, Algorithms, and Implementation" The MIT Press 2005.

[12] BOUGUELANE Nabil. Planification de trajectoire optimale d'un robot manipulateur en utilisant le B-Spline et un critère physique. Automatique. Lieu de soutenance : Université ABDREHMANE Mira de Bejaia, 2012/2013, 74 pages.

[13] Cours B.MENDIL.

[14] <https://tel.archives-ouvertes.fr/tel-00982969/document>

- [15] MECHALIKH Youcf. Développement d'algorithmes d'évitement d'obstacles statiques et dynamiques. Automatique. Lieu de soutenance : Université Kasdi Merbah-Ouargla, 2013, 63.
- [16] C. Ericson, Real Time Collision Detection, Elsevier, 2005.
- [17] J. Dequidt. L. Grisoni. P. Meseure et al. Détection de collision entre objets rigides convexes autonomes. N°CFAO et informatique graphique,2003, 13.
- [18] J. J. Kuffner, "Effective Sampling and Distance Metrics for 3D Rigid body Path Planning", IEEE Int Conf on Robotics and Automation, 2004.
- [19] Ken Shoemake, "Uniform Random Rotations", Graphics GEMS III edited by David Kirk, Academic Press 1992.
- [20] S. Gottschlk, M. C. Lin, D. Manocha, "OBBTree, A Hierarchical Structure for Rapid Inerference detection", Computer Graphics, 30, pp 171-180, 1996.

Résumé :

L'objectif de notre mémoire est de réaliser un planificateur de chemin basé sur la carte de route probabiliste, pour un robot modélisé par un polyèdre dans un environnement statique en 3D encombré d'obstacle, On a commencé notre travail par une généralisation sur la robotique suivie de la formulation du problème de planification de mouvement et quelques méthodes de test de collision présentent dans la littérature.

Pour la construction de la carte on est passé par les étapes suivantes : l'échantillonnage selon une distribution uniforme, échantillonnage basé sur les obstacles (OBB), et la méthode par pont, où les deux dernières méthodes sont utiles pour les passages étroits, Nous avons aussi étudié trois Méthodes des planificateurs locaux : méthodes incrémentales, méthodes binaire et rotation à s (rotat-at-s).

Pour la connexion entre les configurations nous avons utilisé deux méthodes qui sont : k-nearest neighbor et la méthode de forêts. Ensuite vient la phase de requêtes ou nous avons travaillé avec la recherche en profondeur (depth first search).

Enfin, nous avons utilisé le traitement postérieur dans lequel nous avons amélioré le chemin obtenu.

Abstract:

The objective of our work is to realize a path planner based on the probabilistic road map, for a robot modelled by a polyhedron in a static 3d environment obstructed by obstacle, we started our work by a generalization on Robotics followed by the formulation of the problem of motion planning and some methods of collision test present in the literature.

For the construction of the map one went through the following steps: sampling according to a uniform distribution, sampling based on obstacles (OBB), and the method by bridge, where the last two methods are useful for narrow passages, We also studied three methods of local planners: incremental methods, binary methods, and rotation at s (Rotat-at-s).

For the connection between the configurations, we used two methods that are K-nearest neighbor and the forest method. Then comes the query phase where we have worked with depth first search.

Finally, we used the posterior treatment in which we improved the path obtained.

Les mots clés : Polyèdre, collision, planification de mouvement, échantillonnage, configuration