



UNIVERSITE DE ABDERRAHMANE MIRA-BEJAIA.

FACULTE DE LA TECHNOLOGIE.

DEPARTEMENT De Génie Electrique.



## *Mémoire de fin d'étude*

**En vue de l'obtention du diplôme de master  
en électronique**

**Option: Automatique**

**THEME**

**PLANIFICATION DE TRAJECTOIRE OPTIMALE  
D'UN ROBOT MANIPULATEUR EN UTILISANT  
LE B-SPLINE ET UN CRITÈRE PHYSIQUE**

**Réalisé par :**

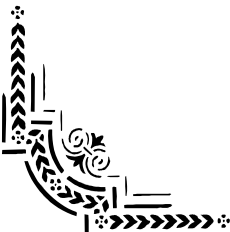
Bouguelane nabil.

Ouali farid.

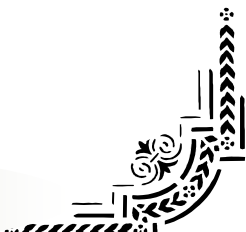
**Encadré par :**

Mr :Haddar hocine

**Promotion:2012/ 2013**



Année: 2012/2013



# Remerciement



Nos remerciements vont tout premièrement à *Dieu* tout puissant pour la volonté, la santé et la patience, qu'il nous a donnée durant toutes ces longues années.

Nous exprimons nos profondes gratitudees à *nos parents* pour leurs encouragements, leur soutien et pour les sacrifices qu'ils ont enduré.

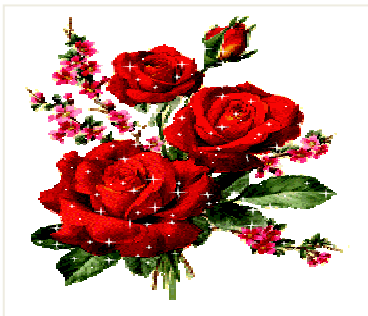
Nous tenons également à exprimer nos vifs remerciements à notre encadreur Monsieur " *Haddar Hocin* " pour avoir d'abord proposé ce thème, pour le suivi continuuel tout le long de la réalisation de ce mémoire et qui n'a pas cessée de nous donner ses conseils et remarques.

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

Nous tenons à remercier vivement tous ceux qui nous ont aidés de prés ou de loin à accomplir ce travail.

Nous remercions vont aussi à tous *les enseignants* du département d'électronique qui ont contribué à notre formation.

Enfin nous tenons à exprimer notre reconnaissance à tous *nos amis et collègues* pour leur soutien moral et matériel...



*Merci*



# DEDICACE



*A ceux qui m'ont donné la vie, symbole de beauté, de fierté, de sagesse et de patience.*

*A ceux qui sont la source de mon inspiration et de mon courage, à qui je dois de l'amour et la reconnaissance.*

- ⊙ *A ma très chère **Mère** et à mon cher **Père** AISSA;*
- ⊙ *A mes frères Karim, Mouloud et mes sœurs ;*
- ⊙ *A mon cher Ami et Co-binôme Nabil pour tous les moments de joies et de peines qu'on a passé ensemble, A sa Famille aussi ;*
- ⊙ *A tous mes amis surtout: Djillali, Zidan, Nassim, Bilal, Faro, sofiane, Sami, Fahim, Bachir, Faycel, Smail...*
- ⊙ *A tous mes collègues de promotion.*
- ⊙ *A tous les proches que j'ai mentionnés et les autres que j'ai oubliés veuillez m'excuser*

*Je vous remercie tous*



*Farid*



## *Dédicace*



*Je dédie ce modeste travail :*

*À Celle à qui mon cœur depuis sa naissance n'a pas pu éprouver qu'amour et reconnaissance, à celle qui a donné un sens à mon existence en m'offrant une éducation digne de confiance*

*À ma chère Mère.*

*À mon père, pour son amour et son soutien moral depuis mon enfance.*

*Père merci.*

*À tout les membres de ma petite famille ; mes sœurs : samira, fahima et dihia, a mes frères : samir et sofiane.*

*À mes grands peres et mes grandes meres.*

*À mes meilleurs amis :*

*nabil, Bachir, yassin, Nassim, mohand.*

*À mes amis de la cité universitaire A-Mira, et à toute les étudiant de ma spécialité, à tous qui m'ont connu et aidé de près et de loin dans la réalisation de ce travail.*



*nabil*

# SOMMAIRE

Liste des notations

Liste des abréviations

Liste des figures

## Introduction générale

### Chapitre I :

#### Modélisation des robots manipulateurs

I.1 Introduction.....	1
I.2 Historique.....	1
I.3 Définition d'un robot .....	1
I.4 Généralité .....	2
I.4.1 Degré de liberté .....	2
I.4.2 Articulation .....	2
I.4.2.1 Articulation rotoïde.....	3
I.4.2.2 Articulation prismatique .....	3
I.4.3 Espace articulaire .....	4
I.4.4 Espace opérationnel .....	4
I.4.5 Redondance .....	4
I.5 Composantes d'une cellule robotisée .....	4
I.5.1 Le mécanisme .....	4
I.5.2 La perception .....	4
I.5.3 La commande.....	5
I.5.4 L'interface homme-machine .....	5
I.5.5 Le poste de travail et les dispositifs péri-robotiques.....	5
I.6 Constitution mécanique du robot.....	5
I.7 Caractéristiques d'un robot .....	6

I.8 Modélisation géométrique .....	7
I.8.1 Modèle géométrique direct.....	7
I.2.1.1 Représentation d'un point dans l'espace .....	7
I.2.1.2 Représentation d'une direction .....	8
I.2.1.3 Matrice de translation .....	8
I.2.1.4 Matrice de rotation .....	9
I.2.1.5 Paramètre de Denavite-Hartenberg .....	10
I.2.2 Modèle géométrique inverse .....	12
I.9 Modélisation cinématique .....	12
I.9.1. Définition.....	12
I.9.2 Modèle cinématique direct.....	13
I.9.2.1 La matrice jacobienne.....	13
I.9.2.2 L'intérêt de la matrice jacobienne .....	13
I.9.2.3 Calcul de la matrice jacobienne par dérivation du MGD.....	13
I.9.3 Modèle cinématique inverse .....	14
I.10 Modélisation dynamique .....	14
I.10.1 Définition .....	14
I.10.2 Modèle dynamique direct .....	15
I.10.3 Modèle dynamique inverse .....	15
I.10.4 Le formalisme d'Euler-Lagrange .....	15
I.11 Conclusion.....	15

## **Chapitre II**

### **Planification de trajectoire et commande optimale**

II.1 Introduction.....	17
II.2 Génération de la trajectoire dans l'espace articulaire.....	17
II.2.1 La fonction polynomiale cubique.....	18
II.2.2 Polynôme de degré cinq .....	19
II.2.3 mouvement continu entre points via .....	21
II.2.4 Commande opérationnelle.....	23

II.3 Contrôleur de robot .....	23
II.4 Les actionneurs principaux.....	23
II.4.1 Les actionneurs électriques.....	23
II.4.1.1 Le moteur à courant continu (MCC).....	24
II.4.1.2 Le moteur synchrone.....	24
II.4.1.3 Le moteur asynchrone .....	24
II.4.1.4 Tableau comparatif des actionneurs électriques.....	25
II.4.2 Les actionneurs hydrauliques.....	25
II.4.3 Les actionneurs pneumatiques.....	25
II.4.4 Les réducteurs.....	25
II.5 Techniques de commande.....	26
II.6 Problème de la commande optimale .....	26
II.6.1 Rapport de la commande optimale.....	27
II.6.2 Contrôlabilité .....	28
II.6.3 Existence de trajectoires optimales .....	28
II.7 Conditions nécessaires d'optimalité .....	28
II.7.1 Equations d'Euler-Lagrange.....	28
II.7.2 Principe du minimum de pontryagin .....	29
II.8 Méthode de résolutions du problème de commande optimale.....	30
II.8.1 Méthodes indirectes .....	30
II.8.1.1 Méthodes en boucle fermée .....	30
II.8.1.2 Méthodes de commande en boucle ouverte .....	30
II.8.2 Méthodes directes.....	31
II.8.2.1 Méthodes de discrétisation .....	31
II.8.2.2 Méthodes de para-métrisation .....	31
II.9 Méthodes générales d'optimisation .....	33
II.9.1 Définition de l'optimisation.....	33
II.9.2 Optimisation locale.....	34
II.9.2.1 Méthode de descente .....	34
II.9.2.2 Algorithme génétique.....	34
II.9.3 Optimisation globale .....	34
II.9.3.1 Méthode des variations locales .....	34

II.9.3.1 Branch and bound .....	35
II.9.3.2 Branch and prune .....	35
II.9.4 Optimisation stochastique .....	36
II.9.4.1 Méthode adaptative .....	36
II.9.5 Considération de gradient .....	36
II.10 Description de la fonction <i>Fmincon</i> de MATLAB ... ..	38
II.11 Conclusion.....	39

## **Chapitre III**

### **Application sur un robot**

III.1 Introduction .....	40
III.2 Génération de la trajectoire par B-splines.....	40
III.3 Planification de trajectoire optimale avec chemin contraint et effort minimal .....	42
III.3.1 Modèle dynamique d'un bras manipulateur rigide à deux degrés de liberté... ..	44
III.3.1.1 Modèle dynamique d'un robot à 2 ddl sans charge sur l'effecteur .....	45
III.3.1.2 Modèle dynamique d'un robot à 2 ddl avec un poids ( $p_w$ ) sur l'organe terminal .....	48
III.3.2. La trajectoire initiale .....	49
III.3.3 Détermination des points de contrôle .....	50
III.3.4 Optimisation de la trajectoire .....	51
III.3.4.1 Approximation de la trajectoire par B-splines .....	51
III.3.4.2 Optimisation .....	51
III.4 Résultats de simulation .....	56
III.5 Conclusion .....	74

### **Conclusion générale**

Références bibliographiques



# LISTE DES NOTATIONS

$m$ (chap I)	Nombre de degrés de liberté
$R$	Articulation rotoïde
$P$	Articulation prismatique
$(P_x, P_y, P_z)$	Coordonnées cartésiennes
$p_t$	Facteur d'échelle
${}^i T_j$	Matrice de transformation
$Trans(a, b, c)$	Matrice de translation pure
$Rot(x, \theta)$	Matrice de rotation pure
$(x, y)$	Variables opérationnelles
$J$	Matrice jacobienne
$F$	Force
$M$	Masse
$w$	Vitesse angulaire
$c_I$	Tenseur d'inertie
$L$	Lagrange
$T$	Energie cinétique totale
$V$	Energie potentielle totale
$\Gamma$	Couple
$(q \text{ ou } \theta)$	Position articulaire
$(\dot{q} \text{ ou } \dot{\theta})$	Vitesse articulaire
$(\ddot{q} \text{ ou } \ddot{\theta})$	Accélération articulaire
$\theta_0$	Position initial
$\theta_f$	Position final
$t$	Temps
$P_j$	Vecteur de points de contrôle
$B_{j,K}$	La fonction de base de B-spline
$J_c$	Le coût
$\partial J_c$	Le gradient de coût
$p_w$	Poids
$t_f$	Temps final
$r_c$	Position du centre de masse
$\dot{r}_c$	Vitesse du centre de masse

# LISTE DES ABRÉVIATIONS

Abréviations	Signification
MGD	Modèle géométrique direct
MGI	Modèle géométrique inverse
MCD	Modèle cinématique direct
MCI	Modèle cinématique inverse
MDD	Modèle dynamique direct
MDI	Modèle dynamique inverse
MCC	Moteur à courant continue
PID	Proportionnel, intégrale, dérivé
HJB	Hamilton-Jacobi-Bellman
NLTPBP	A Non linear Two-point Boundary problem
BFGS	Broyden, Fletcher, Goldfarb, Shanno
SQP	Programmation quadratique séquentielle
ddl	Degrés de liberté
D-H	Denavite-Hartenberg



## Liste des figures

<b>Fig.I.1</b> Exemple explicatif pour la définition d'un degré de liberté.....	2
<b>Fig.I.2</b> représentation d'une articulation rotoïde.....	3
<b>Fig.I.3</b> représentation d'une articulation prismatique.....	3
<b>Fig. I.4</b> Structure ouverte simple.....	5
<b>Fig. I.5</b> Structure arborescentes.....	5
<b>Fig. I.6</b> Structure fermées.....	5
<b>Fig. I.7</b> Structure parallèles.....	6
<b>Fig. I.8</b> Représentation d'un point dans l'espace.....	8
<b>Fig. I.9:</b> translation pure.....	9
<b>Fig.I.10</b> rotation de $\theta$ autour de l'axe x.....	10
<b>Fig. I.11</b> Paramètres géométriques dans le cas d'une structure ouverte simple.....	11
<b>Fig. II.1</b> Génération d'une trajectoire polynomiale.....	19
<b>Fig. II.2</b> Génération d'une trajectoire polynomiale de degré cinq.....	20
<b>Fig. II.3</b> les notions utilisées Génération d'une trajectoire avec un mouvement continu entre points via.....	21
<b>Fig.II.4</b> Génération d'une trajectoire avec un mouvement continu entre points via.....	22
<b>Fig.II.5</b> Tableau comparatif des actionneurs électriques.....	25
<b>Fig.III.1</b> représentation d'un bras manipulateur à 2 ddl.....	45
<b>Fig.III.2</b> trajectoire initiale du robot.....	50
<b>Fig.III.3</b> explication de la fonction $F_{mincon}$ .....	52
<b>Fig.III.4</b> les positions articulaires $\theta_1$ et $\theta_2$ pour la trajectoire initiale.....	57
<b>Fig.III.5</b> les vitesses articulaires $\dot{\theta}_1$ et $\dot{\theta}_2$ pour la trajectoire initiale.....	57
<b>Fig.III.6</b> les accélérations articulaires $\ddot{\theta}_1$ et $\ddot{\theta}_2$ pour la trajectoire initiale.....	58
<b>Fig.III.7</b> les couples initiaux.....	59
<b>Fig.III.8</b> génération de la trajectoire initiale avec deux méthodes.....	60
<b>Fig.III.9</b> Variations des positions articulaires durant la trajectoire optimale.....	60
<b>Fig.III.10</b> les vitesses articulaires le long de la trajectoire optimale.....	60
<b>Fig.III.11</b> Les accélérations articulaires optimales.....	61
<b>Fig.III.12</b> Les couples articulaires durant la trajectoire optimale.....	61
<b>Fig.III.13</b> Le chemin optimisé de l'outil dans l'espace cartésien.....	62
<b>Fig.III.14</b> Variations des positions articulaires durant la trajectoire optimale.....	63

<b>Fig.III.15</b> les vitesses articulaires le long de la trajectoire optimale.....	64
<b>Fig.III.16</b> Les accélérations articulaires optimales.....	64
<b>Fig.III.17</b> Les couples articulaires durant la trajectoire optimale.....	65
<b>Fig.III.18</b> Le chemin optimisé de l'outil dans l'espace cartésien.....	65
<b>Fig.III.19</b> Variations des positions articulaires durant la trajectoire optimale.....	67
<b>Fig.III.20</b> les vitesses articulaires le long de la trajectoire optimale.....	67
<b>Fig.III.21</b> Les accélérations articulaires optimales .....	68
<b>Fig.III.22</b> Les couples articulaires durant la trajectoire optimale.....	68
<b>Fig.III.23</b> Le chemin optimisé de l'outil dans l'espace cartésien.....	69
<b>Fig.III.24</b> Variations des positions articulaires durant la trajectoire optimale.....	70
<b>Fig.III.25</b> les vitesses articulaires le long de la trajectoire optimale.....	71
<b>Fig.III.26</b> Les accélérations articulaires optimales.....	71
<b>Fig.III.27</b> Les couples articulaires durant la trajectoire optimale.....	72
<b>Fig.III.28</b> Le chemin optimisé de l'outil dans l'espace cartésien.....	72

## **INTRODUCTION GÉNÉRALE**

---

## **Introduction générale :**

De nos jours, les applications des robots ne sont pas limitées, comme c'était le cas dans le passé, à remplacer l'humain dans des tâches industrielles répétitives. Le domaine d'applications des robots s'est beaucoup élargi, incluant l'exploration sous-marine et planétaire, réparation de satellites, désamorçage d'explosives, le travail dans des environnements radioactifs et même domestiques, les prothèses...

Dans les applications industrielles, les robots se présentent sous forme de chaînes mécaniques articulées commandées à partir de calculateurs. Les tâches qu'ils doivent réaliser sont effectuées en déplaçant un outil fixé à l'extrémité du robot selon une trajectoire déterminée.

La commande de ces robots pour effectuer ce genre de tâches est, généralement accomplie en trois étapes : planification de chemin, génération de la trajectoire est enfin la poursuite de cette trajectoire. Dans la première étape, on spécifie le chemin dans l'espace opérationnel (cartésien) ou dans l'espace de configuration pour déplacer l'outil vers une position de destination en évitant la collision avec des objets dans son espace de travail, ce chemin représente la position et l'orientation de l'outil sans prendre en considération le temps c'est-à-dire sans considérer les vitesses et les accélérations. Ensuite, dans la deuxième étape, on calcul l'historique temporelles de la position et de ses dérivée en prenant en considérations les limites physiques sur les différentes variables. Dans notre travail, nous nous intéressons à la première et la deuxième étape que nous allons appeler tout simplement planification de trajectoires.

Dans la littérature beaucoup de méthodes ont été proposées pour la planification de la trajectoire. Généralement cette dernière est paramétrée en utilisant différents techniques d'interpolation, ensuite, les paramètres sont calculés pour répondre à certaines exigences.

De point de vue engineering, l'utilisation d'une approche basée sur l'optimisation pour la génération de mouvement paraît comme la première chose raisonnable qu'on doit essayer. Le seul obstacle rencontré est la complexité des équations dynamiques des robots qui conduisent à un problème d'optimisation insoluble. En plus, ces méthodes d'optimisation utilisent le gradient de la fonction coût pour calculer le point optimal. Mais si on ne dispose pas d'expressions analytique de ce gradient, l'optimisation échoue, généralement, à converger vers une solution. En effet ces difficultés sont (au moins partiellement) responsables de l'intérêt porté aux réseaux de neurones, les algorithmes génétiques...

Cependant, les méthodes d'optimisations disponibles quand elles disposent de méthodes analytiques pour le calcul du gradient, permettent de trouver une solution au problème de planification de façon attractive.

Dans notre travail, nous avons représenté la trajectoire en utilisant le B-spline. Les points de contrôle de ce dernier sont les paramètres qu'il faut calculer pour obtenir la trajectoire optimale. Les valeurs initiales de ces paramètres sont choisies, au départ, à partir d'une trajectoire initiale non optimale. Le coût de performance, qu'il faut minimiser, compte pour l'effort des actionneurs, la maximisation du poids solvable par le robot et la minimisation de surcharge des actionneurs. Nous avons établie les expressions analytiques nécessaires au calcul du gradient de la fonction coût pour un robot à deux articulations rotoïdes. Pour l'optimisation nous avons fait usage de la fonction `fmincon` du toolbox optimisation de MATLAB.

Notre mémoire est organisé comme suit :

Chapitre I : consacré à la modélisation géométrique, cinématique et dynamique des robots qui est basée sur les formalismes de Denavit-Hartenberg et d'Euler-Lagrange. Et un ensemble de généralités et de notions de base sur la robotique.

Chapitre II : dans ce chapitre nous aborderons les techniques de génération des trajectoires et de planification des tâches dans l'espace opérationnel, avec un rappel sur la commande optimale en général et sur l'optimisation paramétrique.

Chapitre III : dans ce chapitre nous avons présentés les différents calculs effectués pour appliquer cette commande à un exemple (robot à deux degrés de liberté) ainsi que les résultats de simulation. Enfin, nous clôturons ce mémoire par une conclusion générale.



## **CHAPITRE I**

# **MODÉLISATION DES ROBOTS MANIPULATEURS**

---

## I.1 Introduction :

La *robotique* peut être définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles.

Jusqu'à présent, les robots industriels sont surtout employés pour servir de générateur de positions et d'orientations d'un outil dans l'espace. Pour cela, ils ne sont modélisés que par trois modèles: Le modèle géométrique, Le modèle cinématique et Le modèle dynamique. Un robot industriel peut-être vu comme un système mécanique poly articulé. Pour pouvoir faire exécuter une action où une tâche au robot manipulateur, il est nécessaire de modéliser ce système multi variable pour positionner les différents axes, par contre un modèle de déformation est une nécessité. Le but de la modélisation dans la robotique est de fournir les équations mathématiques qui permettent de programmer, commander et exécuter les tâches désirées.

L'utilisation de la paramétrisation de Denavit-Hartenberg (D-H) facilite la description géométrique du manipulateur, cette dernière nous permet d'aboutir au modèle cinématique et géométrique direct et inverse du robot. La même transformation offre une souplesse dans le calcul du modèle dynamique direct en utilisant le formalisme d'Euler-Lagrange.

## I.2 Historique [1]:

Le terme robot vient du Tchèque, il signifie travail forcé le mot sous son acception moderne, fut introduit par l'auteur tchèque Karel Capek dans son ouvrage R.U.R .qui date de 1921 en anglais Rossum's Universal Robots.

Le concept de Robot industriel fut breveté en 1954, par G.C. Devol (Brevet U.S. N°=2988237) ; ce brevet décrit la réalisation d'un bras mécanique asservi, capable d'effectuer des tâches industrielles.

## I.3 Définition d'un robot [3] [2]:

Le Petit Larousse définit un robot comme étant un appareil automatique capable de manipuler des objets, ou d'exécuter des opérations selon un programme fixe ou modifiable.

La définition générique stipule qu'un Robot est une machine physique qui modifie matériellement son environnement pour atteindre le but qui lui est fixé : la tâche désirée.

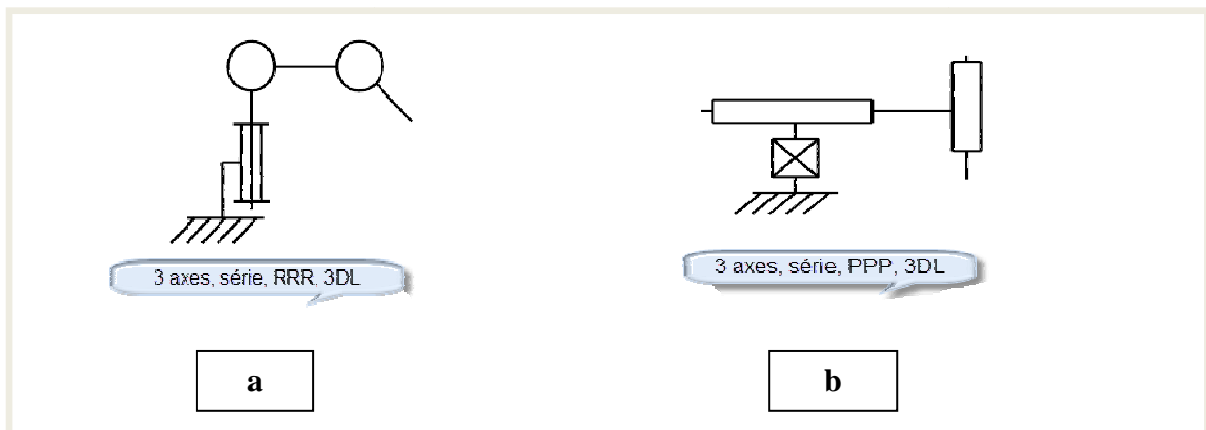
D'après l'Association Française de 'NORmalisation(A.F.NOR)', un robot est un manipulateur commandé en position, reprogrammable, polyvalent, à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des dispositifs

spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un ou plusieurs bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel.

## I.4 Généralités [2]:

### I.4.1 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 fonction de la configuration du robot. Cela sera illustré dans l'exemple suivant en figure. **Fig.I.1:**



**Fig.I.1** : Exemple explicatif pour la définition d'un degré de liberté : (a) bras à 3 degré de liberté RRR, (b) bras à 3 degré de libertés PPP

La figure **Fig. I.1. (a)** représente un bras comportant 3 liaison *rotoïde*, pour la **Fig.I.1.(b)**, le bras comporte aussi le même nombre d'articulations, mais combinées différemment, ce qui lui permet de faire bouger l'organe terminal dans le sens de  $Y$  (grâce à l'articulation  $P3$  et  $P1$ ) et de  $X$  (grâce à l'articulation  $P2$ ).

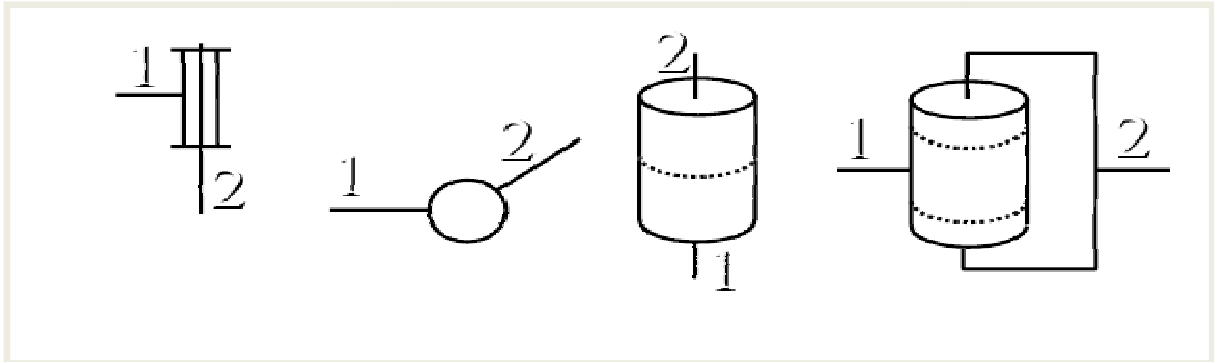
### I.4.2 Articulations :

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*, soit *prismatique*.

**I.4.2.1 Articulation rotoïde:**

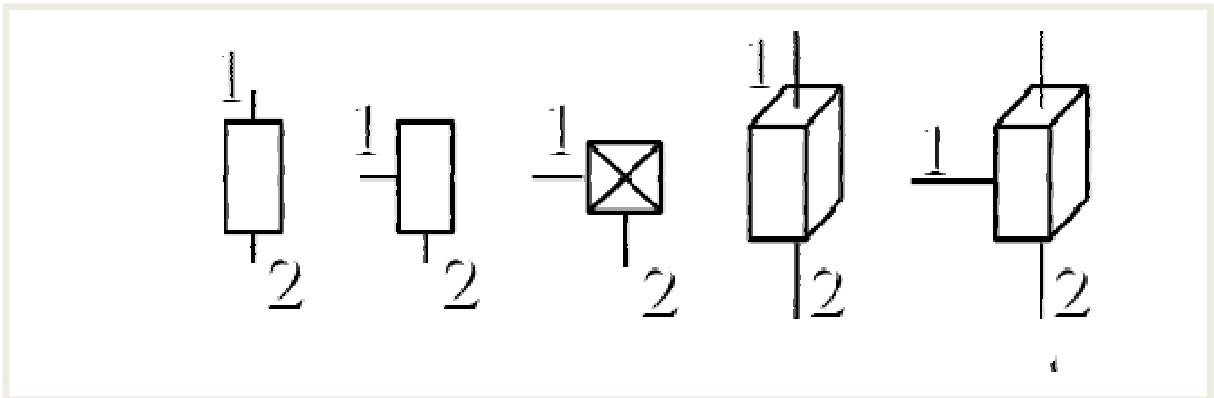
Il s'agit d'une articulation de type pivot réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe. L'articulation rotoïde est représentée par le symbole de la figure **Fig.I.2** et est notée **R**.



**Fig.I.2** Représentation d'une articulation rotoïde.

**I.4.2.2 Articulation prismatique:**

Il s'agit d'une articulation de type glissière réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe. La figure **Fig.I.3** donne sa représentation symbolique. On la note **P**.



**Fig.I.3** Représentation d'une articulation prismatique.

**Remarque :** on peut créer des liaisons de mobilité supérieure à 1 en combinant des articulations simples. Une rotule, par exemple, est obtenue avec trois articulations *rotoïdes* d'axes concourants.

### I.4.3 Espace articulaire :

L'espace articulaire d'un robot est celui dans lequel est représentée la situation de tous ses corps. La solution la plus simple consiste à utiliser les variables ou coordonnées articulaires. Sa dimension  $N$  est égale au nombre de variables articulaires indépendantes et correspond au nombre de degrés de liberté de la structure mécanique.

### I.4.4 Espace opérationnel :

L'espace opérationnel est celui dans lequel est représentée la situation de l'organe terminal (on considère donc autant d'espaces opérationnels qu'il y a d'organes terminaux). Sa dimension est égal 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 placer un point du corps en un point quelconque de cet espace et trois pour orienter ce corps de façon quelconque).

### I.4.5 Redondance :

Un robot est redondant lorsque le nombre de degrés de liberté de l'organe terminal est inférieur au nombre de degrés de liberté de l'espace articulaire (nombre d'articulations motorisées). Cette propriété permet d'augmenter le volume du domaine accessible et de préserver les capacités de déplacement de l'organe terminal en présence d'obstacles ( $N < 6$ ).

**Remarque :** pour une structure comportant plusieurs organes terminaux, la redondance s'évalue en comparant les nombres de degrés de liberté de l'espace articulaire agissant sur chaque organe terminale et le nombre de degré de liberté de l'espace opérationnel correspondant.

## I.5 Composantes d'une cellule robotisée [2]:

### I.5.1 Le mécanisme :

Ayant une structure plus au moins proche de celle du bras humain, il permet de remplacer ou de prolonger son action (le terme manipulateur exclue implicitement les robots mobiles). Sa motorisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés.

### I.5.2 La perception :

Elle permet de gérer la relation entre le robot et son environnement. Les organes de perception sont des capteurs dits "*proprioceptifs*" lorsqu'ils mesurent l'état interne du robot, et "*extéroceptifs*" lorsqu'ils recueillent des informations sur l'environnement.

**I.5.3 La commande :**

Elle synthétise les consignes des asservissements pilotant les actionneurs. A partir de la fonction de perception et des ordres de l'utilisateur, elle permet d'engendrer les actions du robot.

**I.5.4 L'interface homme-machine :**

A travers laquelle l'utilisateur programme les tâches que le robot doit exécuter;

**I.5.5 Le poste de travail les dispositifs péri-robotiques :**

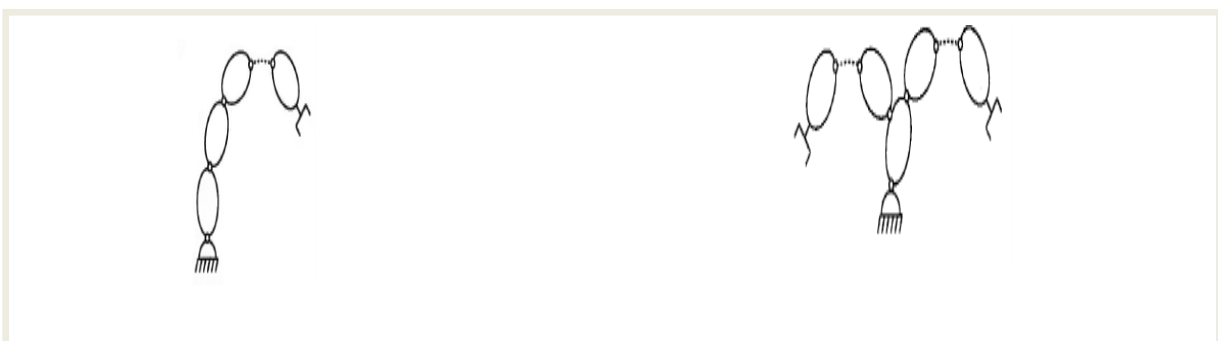
Ils constituent l'environnement dans lequel évolue le robot.

**I.6 Constitution mécanique du robot [3] :**

Un robot manipulateur est constitué par deux sous-ensembles distincts, un (ou plusieurs) organes terminaux et une structure mécanique articulée :

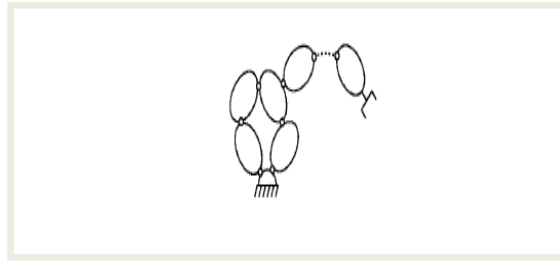
Sous le terme organe terminal, on regroupe tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression ...) ou à les transformer (outils, torche de soudage, pistolet de peinture...). Il s'agit donc d'une interface permettant au robot d'interagir avec son environnement.

Le rôle de la structure mécanique articulée 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, assemblés par des liaisons appelées articulations. Les chaînes peuvent être soit ouvertes simples (**Fig. I.4**), soit arborescentes (**Fig. I.5**), soit fermées (**Fig. I.6**).



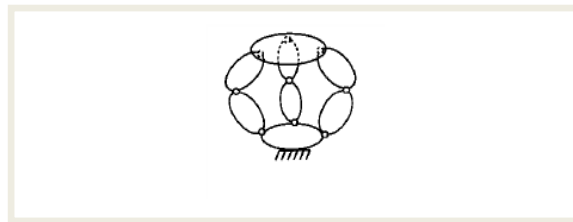
**Fig. I.4** Structure ouverte simple.

**Fig. I.5** Structure arborescentes.



**Fig. I.6** Structure fermées.

La figure (**Fig. I.7**) montre une génération différente des robots – les robots parallèles – dans laquelle l'organe terminal est relié à la base du mécanisme par plusieurs chaînes parallèles.



**Fig. I.7** Structure parallèles.

### I.7 Caractéristiques d'un robot [2] :

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 envisagées :

- 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 essentiellement 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 **ISO 9283**)
- la résolution : c'est la plus petite modification de la configuration du robot à la fois observable et contrôlable par le système de commande.

Cependant, pour une tâche donnée d'autre caractéristique doivent être prises en compte : techniques (énergie, commande, programmation ..... ) et commerciales (coût, maintenances...)

## I.8 Modélisation géométrique [2] :

Dans la modélisation géométrique on s'intéresse au mouvement sans tenir compte des forces qui le provoquent. Elle s'intéresse à l'étude de la géométrie du robot en vue de décrire ses paramètres géométriques : position et orientation.

### I.8.1 Modèle géométrique direct [2] :

Le modèle géométrique directe (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du robot en fonction de ses coordonnées articulaires.

Le (MGD) peut être représenté par la relation :

$$X = f(q) \quad (\text{I.1})$$

où :

$X$  : Le vecteur des coordonnées opérationnelles ;

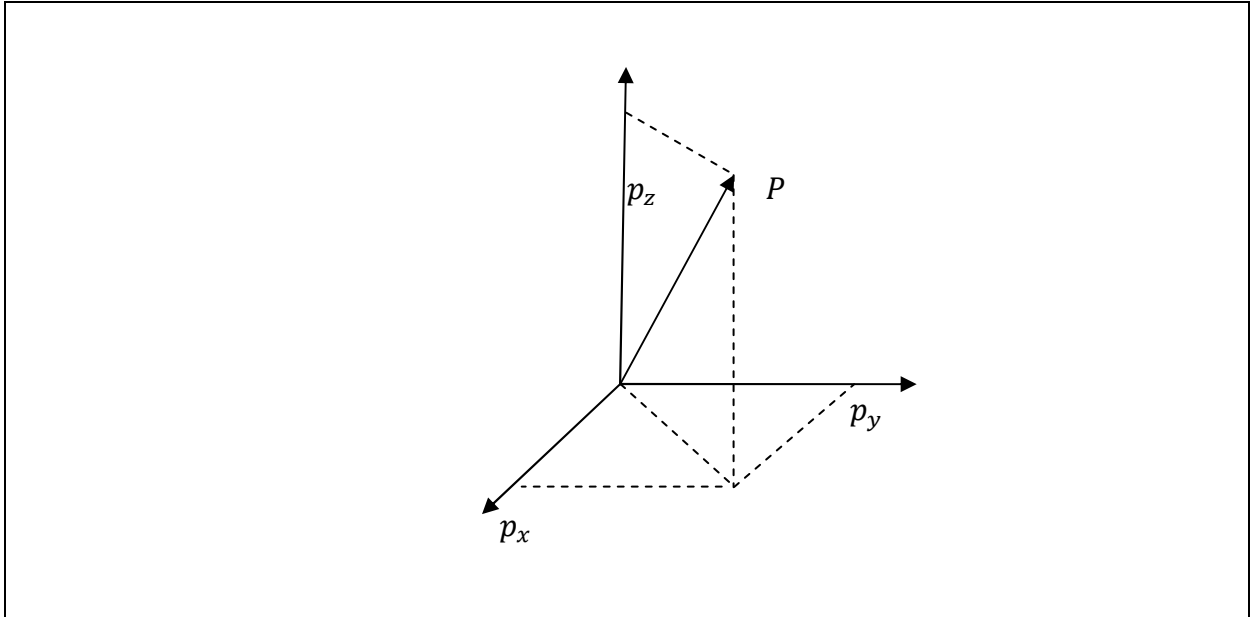
$q$  : Le vecteur des coordonnées articulaires ;

#### I.8.1.1 Représentation d'un point dans l'espace [2] :

Soit  $P$  un point de coordonnées cartésiennes  $p_x, p_y, p_z$  (fig.1.4). On appelle coordonnées homogènes du point  $P$ , les termes  $w \times p_x, w \times p_y, w \times p_z$  ou 'w' est un facteur d'échelle égale à 1 en robotique. On présente alors les coordonnées homogènes d'un point par le vecteur :

$$P = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$





**Fig. I.8** Représentation d'un point dans l'espace.

### I.8.1.2 Représentation d'une direction [2] :

La représentation d'une direction (vecteur libre) se fait aussi par quatre composantes, la quatrième étant nulle, indique un point à l'infini. Si l'on note  $u_x, u_y, u_z$  les coordonnées cartésiennes d'un vecteur unitaire  $u$ , en coordonnées homogènes on écrit :

$$u = \begin{bmatrix} u_x \\ u_y \\ u_z \\ 0 \end{bmatrix} \quad (\text{I.2})$$

### I.8.1.3 Matrice de translation [2] :

Soit de repère  $R_0(O_0, x_0, y_0, z_0)$ ;  $R_1(O_1, x_1, y_1, z_1)$  et soient  $p_x, p_y, p_z$  les coordonnées de l'origine  $O_1$  du repère  $R_1$  dans le repère  $R_0$ . Lorsque les axes des deux repère  $R_0$  et  $R_1$  sont en parallèle, alors la transformation de A vers B est appelée "matrice de translation" et donnée par :

$$A = \text{Trans}(p_x, p_y, p_z) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{I.3})$$

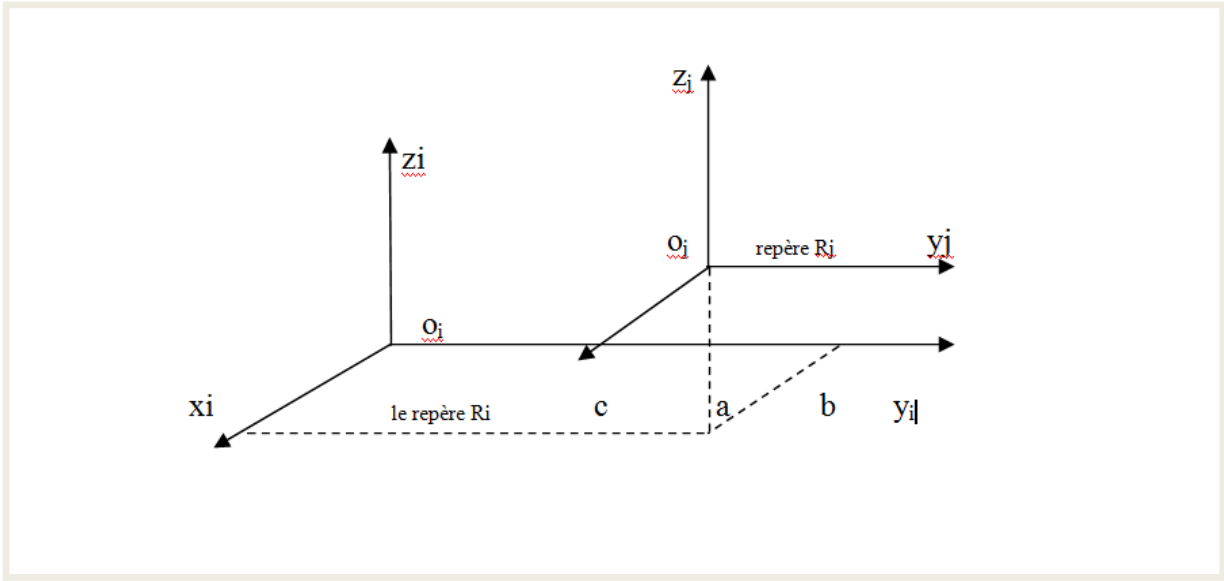


Fig. I.9: translation pure

**I.8.1.4 Matrice de rotation :**

Soient deux repères  $R_0$  et  $R_1$  précédemment définis, lorsque leurs origines coïncident ( $O_0 = O_1$ ), et que le repère  $R_1$  effectue une rotation d'un angle  $\theta$  autour d'un ou plusieurs axes de  $R_0$  alors cette rotation peut être décomposée en rotations élémentaires de différents angles autour des axes de  $R_0$ , la matrice de transformation est obtenue par le produit de ces trois matrices de rotation :

- Rotation d'un angle  $\theta_1$  autour de  $x_0$  :

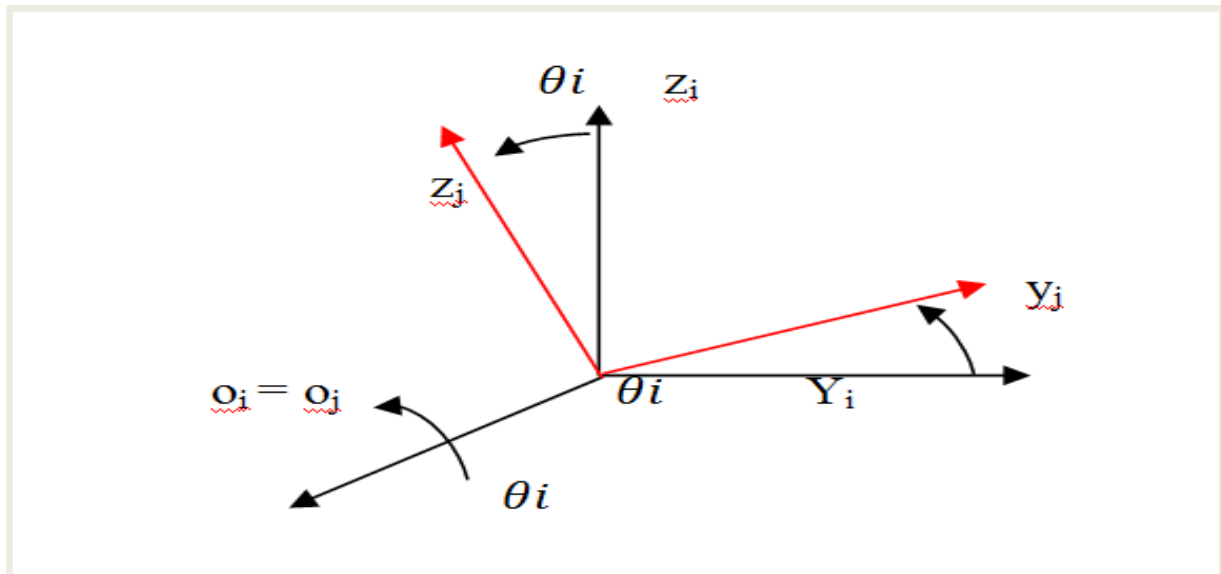
$$Rot(x_0, \theta_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_1 & -\sin \theta_1 & 0 \\ 0 & \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{I.4}$$

- Rotation d'un angle  $\theta_2$  autour  $y_0$  :

$$Rot(y_0, \theta_2) = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{I.5}$$

- Rotation d'un angle  $\theta_3$  autour de  $z_0$  :

$$Rot(z_0, \theta_3) = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{I.6}$$



**Fig.I.10** rotation de  $\theta$  autour de l'axe  $x$

#### I.8.1.5 Paramètre de Denavit-Hartenberg [10] :

La modélisation des robots de façon systématique et automatique exige une méthode adéquate pour la description de leur morphologie. Plusieurs méthodes et notation ont été proposées pour ça. Mais la plus répandue est celle de *Denavit-Hartenberg*, mais cette méthode développée pour des structures ouvertes simples, présente des ambiguïtés lorsqu'elle est appliquée sur des robots ayant des structures fermées ou arborescentes.

Une structure ouverte simple est composée de  $(n + 1)$  corps notés  $C_0, \dots, C_n$  et de  $n$  articulations. Le corps  $C_0$  désigne la base du robot et le corps  $C_n$  le corps qui porte l'organe terminal. L'articulation  $j$  connecte le corps  $C_j$  au corps  $C_{j-1}$ .

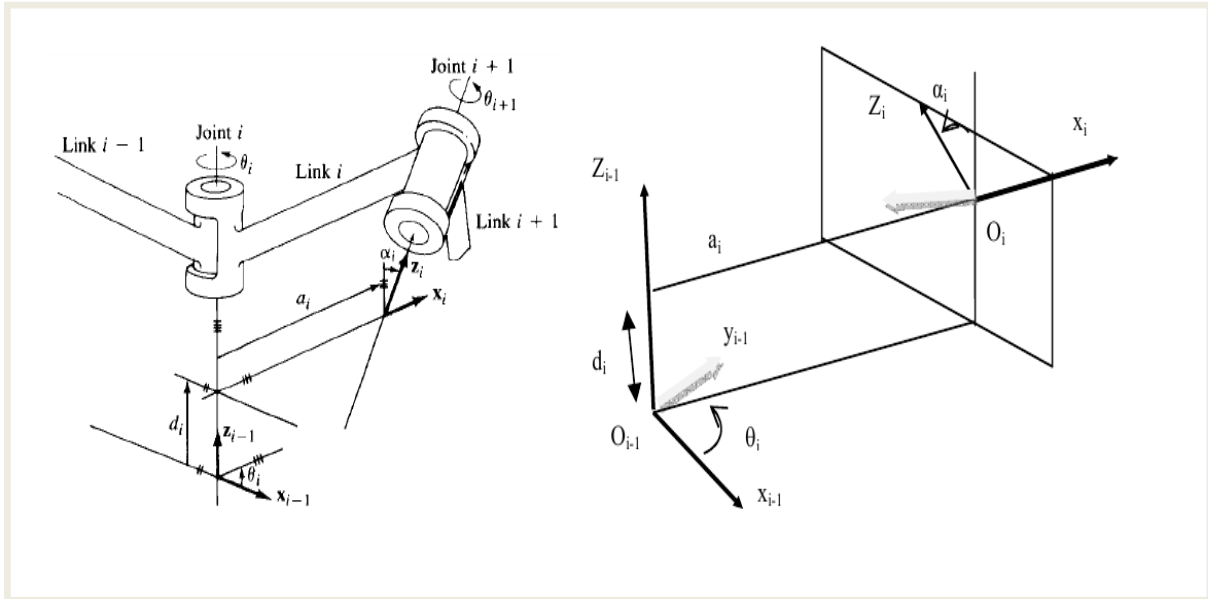
La méthode de description est fondée sur les règles et conventions suivantes :

- Les corps sont supposés parfaitement rigides.
- Les articulations sont considérées comme idéales (pas de jeu mécanique, pas d'élasticité).
- Articulations soit prismatiques, soit rotoïdes.
- Le repère  $R_j$  est lié au corps  $C_j$ .
- L'axe  $z_j$  est porté par l'axe de l'articulation  $j$ .
- L'axe  $x_j$  est porté par la perpendiculaire commune aux axes  $z_j$  et  $z_{j+1}$ .

Si les axes  $z_j$  et  $z_{j+1}$  sont parallèles ou colinéaires le choix de  $x_j$  n'est pas unique : des considérations de symétrie ou de simplicité permettent alors un choix rationnel.

Le passage du repère  $R_{j-1}$  au repère  $R_j$  s'exprime en fonction des quatre paramètres géométriques suivants :

- $\alpha_j$ : Angle entre les axes  $z_{j-1}$  et  $z_j$  correspondant à une rotation autour de  $x_{j-1}$  ;
- $d_j$  : Distance entre  $z_{j-1}$  et  $z_j$  le long de  $x_{j-1}$  ;
- $\theta_j$ : Angle entre les axes  $x_{j-1}$  et  $x_j$  correspondant à une rotation autour de  $z_j$  ;
- $r_j$ : Distance entre  $x_{j-1}$  et  $x_j$  le long de  $z_j$ ;



**Fig. I.11** Paramètres géométriques dans le cas d'une structure ouverte simple.

La variable articulaire  $q_j$  associée à la  $j$ -ième articulation est soit  $\theta_j$ , soit  $r_j$ , selon que cette articulation est rotoïde ou prismatique, ce qui se traduit par la relation :

$$q_j = \delta'_j \cdot \theta_j + \delta_j \cdot r_j \tag{I.7}$$

Avec:

- $\delta_j = 0$  si l'articulation est rotoïde ;
- $\delta_j = 1$  si l'articulation est prismatique ;
- $\delta'_j = 1 - \delta_j$  ;

La matrice de transformation définissant le repère  $R_j$  dans le repère  $R_{j-1}$  est donnée par :

$${}^{j-1}T_j = Rot(x, \alpha_j) \times Trans(x, d_j) \times Rot(z, \theta_j) \times Trans(z, r_j) \tag{I.8}$$

$${}^{j-1}T_j = \begin{bmatrix} \cos \theta_j & -\sin \theta_j & 0 & d_j \\ \cos \alpha_j \cdot \sin \theta_j & \cos \alpha_j \cdot \sin \theta_j & -\sin \alpha_j & -r_j \times \sin \alpha_j \\ r_j \times \cos \alpha_j & \sin \alpha_j \cdot \sin \theta_j & \sin \alpha_j \cdot \cos \theta_j & \cos \alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{I.9}$$

où  $Rot(u, \alpha)$  et  $Trans(u, d)$  sont des matrices homogènes (4x4) représentant respectivement une rotation et une translation  $d$  le long de  $u$ .

### I.8.2 Modèle géométrique inverse [10] [2] :

Le modèle géométrique direct d'un robot permet de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires. Le problème inverse (MGI) consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminale.

Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue ce que l'on appelle le modèle géométrique inverse (MGI). Le (MGI) peut être représenté par la relation :

$$q = f^{-1}(X) \quad (\text{I.10})$$

On peut distinguer trois méthodes du calcul du MGI :

- La méthode de Paul, qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels ;
- La méthode piper, qui permet de résoudre le problème pour les robots à six degrés de liberté possédant trois articulations rotoïdes d'axes concourants ou trois articulations prismatiques ;
- La méthode générale de Raghavan et Roth, donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égal à 16.

Lorsqu'il n'est pas possible de trouver une forme explicite du modèle géométrique inverse, on peut calculer une solution particulière par des procédures numériques.

## I.9 Modélisation cinématique [5] :

### I.9.1 Définition :

Le modèle cinématique est, littéralement, un modèle des vitesses. Il exprime les relations entre les vitesses articulaires de chaque joint et les vitesses cartésiennes d'un point de la chaîne cinématique, généralement l'organe terminal. Ce modèle est donc un modèle par accroissements élémentaires: chaque variation élémentaire de la valeur d'une articulation implique une variation de position de l'organe terminal, et inversement. Lorsque ces variations infinitésimales sont exprimées par rapport au temps, on peut les considérer comme des vitesses.

Le modèle cinématique permet donc non seulement de compléter éventuellement le modèle géométrique en tenant compte des vitesses, mais aussi de remplacer le modèle

géométrique: en agissant par accroissements successifs, on peut se déplacer d'un point donné à un autre.

Le modèle cinématique possède une propriété essentielle: il est une différentiation du modèle géométrique. Il est donc une linéarisation du système d'équations non linéaires représentant le modèle géométrique. Par conséquent on peut toujours facilement obtenir les transformations inverses puisqu'elles proviennent de l'inversion d'un problème linéaire.

### I.9.2 Modèle cinématique direct [6] [5] :

Le modèle cinématique direct (MCD) d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires. Il est noté :  $\dot{x}$

$\dot{x} = J(q) \dot{q}$                       Où  $J(q)$  désigne la matrice jacobienne de dimension  $(m \times n)$  du mécanisme, égale à  $\frac{\partial x}{\partial q}$  et fonction de la configuration articulaire  $q$ . La même matrice jacobienne intervient dans le calcul du modèle différentiel direct qui donne les variations élémentaires  $dx$  des coordonnées opérationnelles en fonction des variations  $dq$ , soit :

$$dx = J(q) dq \quad (I.11)$$

#### I.9.2.1 La matrice jacobienne :

L'outil principalement utilisé pour traiter le problème de la cinématique des robots est la matrice jacobienne. Elle représente un opérateur permettant de lier les vitesses des corps d'un robot exprimées dans différents espaces vectoriels.

#### I.9.2.2 L'intérêt de la matrice jacobienne :

- elle est à la base du modèle cinématique inverse, permettant de calculer une solution locale des variations articulaires  $dq$  connaissant les variations opérationnelles,  $dx$
- en statique, on utilise le jacobien pour établir la relation liant les efforts exercés par l'organe terminal sur l'environnement aux forces et couples des actionneurs ;
- elle facilite le calcul des singularités et de la dimension de l'espace opérationnel accessible du robot.

#### I.9.2.3 Calcul de la matrice jacobienne par dérivation du MGD :

Le calcul de la matrice jacobienne peut se faire en dérivant le MGD,  $X = f(q)$  à partir de la relation :

$$J_{ij} = \frac{\partial f_i(q)}{\partial q_j} \quad i = 1, \dots, m ; j = 1, \dots, n \quad (I.12)$$

$J_{ij}$  est l'élément  $(i, j)$  de la matrice jacobienne  $J$

Cette méthode est facile à mettre en œuvre pour les robots à deux ou trois degrés de liberté, mais elle devient de plus en plus compliquée dans le cas du manipulateur à six degrés de liberté.

### I.9.3 Modèle cinématique inverse [5] :

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. On peut déterminer la différentielle articulaire  $dq$  correspondante à une différentielle des coordonnées opérationnelles spécifiée  $dX$ . Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires. La mise en œuvre peut être faite de façon analytique ou numérique :

- la solution analytique a pour avantage de diminuer considérablement le nombre d'opérations, mais on doit traiter séparément tous les cas singuliers.
- les méthodes numériques sont plus générales, la plus répandue étant fondée sur la notion de pseudo inverse : les algorithmes traitent de façon unifiée les cas réguliers, singuliers et redondants. Elles nécessitent un temps de calcul relativement important. On peut écrire le modèle cinématique inverse sous la forme

$$\dot{q} = J^{-1} \dot{X} \quad (\text{I.13})$$

## I.10 Modélisation dynamique [1] :

### I.10.1 Définition :

Le modèle dynamique est la relation entre les couples (et/ou les forces) appliqués aux actionneurs avec les positions, vitesses et accélérations articulaires. On représente le modèle dynamique inverse par une relation de la forme :

$$\Gamma = f(q, \dot{q}, \ddot{q}, f_e) \quad (\text{I.14})$$

avec :

$\Gamma$  : vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique ;

$q$  : vecteur des positions articulaires ;

$\dot{q}$  : vecteur des vitesses articulaires ;

$\ddot{q}$  : vecteur des accélérations articulaires ;

$f_e$  : vecteur représentant l'effort extérieur (forces et moment) qu'exerce le robot sur l'environnement ;

**I.10.2 Modèle dynamique direct :**

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

$$\ddot{q} = g(q, \dot{q}, \Gamma, f_e) \quad (\text{I.15})$$

Parmi les applications du modèle dynamique direct, on peut citer :

- La simulation qui utilise le modèle dynamique direct ;
- Le dimensionnement des actionneurs ;
- L'identification des paramètres inertiels et des paramètres de frottement du robot ;

**I.10.3 Modèle dynamique inverse :**

On convient d'appeler que le modèle dynamique inverse (MDI), ou tout simplement modèle dynamique, est la relation :

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

Le modèle dynamique inverse est utilisé surtout pour la commande.

Pour obtenir le modèle dynamique, il existe plusieurs formalismes ; les plus utilisés :

Le formalisme de Newton-Euler et le formalisme de Lagrange (qu'on va utiliser après dans ce travail).

**I.10.4 Le formalisme d'Euler-Lagrange :**

Le formalisme d'Euler-Lagrange décrit les équations du mouvement en termes de travail et d'énergie du système, ce qui se traduit, lorsque l'effort extérieur sur l'organe terminal est supposé nul, par l'équation :

$$\Gamma = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} \quad (\text{I.16})$$

Avec :

- $L$ : lagrangien du système égale à  $T - V$ .
- $T$  : énergie cinétique totale du système.
- $V$ : énergie potentielle totale du système.

**I.11 Conclusion :**

Dans ce chapitre nous avons présenté quelques notions sur les robots manipulateurs, avec la définition de certains termes.

Puis nous avons passé à la modélisation des robots manipulateurs qui est basée sur la modélisation de la structure mécanique, les actionneurs et les capteurs. Elle permet la



détermination des relations directes et inverses entre les coordonnées généralisées et les coordonnées cartésiennes et leurs dérivées respectives.

La convention de Denavit-Hartenberg fournit une souplesse dans le calcul pour la localisation de la position de l'élément terminal. La position initiale du robot influe sur toute la modélisation. La définition des axes selon Denavit-Hartenberg influe sur le modèle dynamique parce que les moments d'inertie des différents axes changent. Le passage de l'espace des coordonnées cartésiennes de l'élément terminal à celui des coordonnées généralisées ou articulaires présente des singularités physiques (point en dehors de l'espace de travail), ces singularités apparaissent sous forme de contraintes lors du calcul du modèle géométrique inverse.

Les formalismes de Lagrange et Euler permettent d'établir un système d'équations différentielles reliant les coordonnées généralisées aux forces et/ou couples généralisés. Ce formalisme présente plusieurs approches dont le résultat est le même, mais pour des applications différentes il est parfois nécessaire d'avoir des séparations au niveau des termes d'inerties, centrifuges, Coriolis et de gravité.

Les modèles obtenus mettent en évidence la difficulté de commander ce genre de bras manipulateurs, étant donné les forts non linéarités qui régissent le fonctionnement dynamique de ceux-ci.

## **CHAPITRE II**

### **PLANIFICATION DE TRAJECTOIRE ET COMMANDE**

#### **OPTIMALE**

---

## II.1 Introduction [3] [2]:

La génération de mouvement désigne la fonction de calcul des consignes articulaires du robot destinées à réaliser une tâche interprétée sous forme de positions successives de l'outil du robot et de contraintes cinématiques ou dynamiques. Cette fonction de génération de mouvements peut s'exercer dans l'espace articulaire, ou elle consiste à déterminer une loi horaire pour chaque articulation comme dans l'espace opérationnel ou elle vise à déterminer directement une trajectoire cartésienne de l'outil et de son orientation. Le fondement mathématique de la génération de mouvements est le calcul polynomial qui construit l'équation du mouvement à partir des contraintes spatiales et temporelles.

Les problèmes de commande optimale se rencontrent dans la vie de tous les jours : comment arriver à destination le plus rapidement possible, comment minimiser sa consommation... Pour un système dynamique comme *le model dynamique d'un robot manipulateur* et dont les équations sont connues, le problème de commande optimale consiste alors à trouver la commande minimisant un critère donné. C'est sous cette forme que la commande optimale a été étudiée des le XIXe`me siècle avec le calcul des variations. Une des grandes applications de la commande optimale a été l'application au lanceur Apollo dans les années 1960. Notons néanmoins que les difficultés soulevées par ce genre de problème sont loin d'être complètement résolues comme en témoignent les sessions dédiées à la commande optimale dans les conférences d'automatique. La commande optimale reste donc un sujet de recherche d'actualité.

## II.2 Génération de la trajectoire dans l'espace articulaire [2]:

La génération de la trajectoire dans l'espace articulaire donne en résultat un ensemble de données : position, vitesse et accélération articulaires  $(\theta, \dot{\theta}, \ddot{\theta})$  qui sont utilisées comme un signal de référence. Pour une position initiale et finale données dans l'espace de travail on utilise la géométrie inverse pour déterminer les angles articulaires correspondant à cette position et même pour l'orientation, la position initiale du manipulateur devient un ensemble d'angles articulaires de départ et la position finale correspond à un autre ensemble d'angles articulaires d'arrivée, ce qui est demandé pour la planification de la trajectoire est de trouver une fonction lisse pour chaque articulation dont la valeur à l'instant  $t_0$  est la position initiale de l'articulation et dont la valeur à l'instant  $t_f$  est la position désirée de cette même articulation. Il y a beaucoup de fonctions lisses  $\theta(t)$  qui pourraient être utilisées pour interpoler les valeurs d'angles articulaires. On cite :

**II.2.1 La fonction polynomiale cubique [5,6]:**

Il s'agit d'une interpolation simple en effectuant un simple mouvement continu, au moins quatre contraintes sur  $\theta(t)$  sont évident pour avoir une fonction polynomiale cubique de la forme :

$$\theta(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 \tag{II.1}$$

Ainsi on obtient la vitesse articulaire :

$$\dot{\theta}(t) = a_1 + 2 \cdot a_2 \cdot t + 3 \cdot a_3 \cdot t^2 \tag{II.2}$$

Et aussi l'accélération articulaire :

$$\ddot{\theta}(t) = 2 \cdot a_2 + 6 \cdot a_3 \cdot t \tag{II.3}$$

Sachant qu'un polynôme du 3ème degré, admet quatre coefficients, il peut être donc construit à partir de quatre contraintes, deux sont obtenues à partir du choix des valeurs initiales et finales de la position :

$$\theta(0) = \theta_0, \theta(t_f) = \theta_f \tag{II.4}$$

Quant aux deux autres contraintes, elles proviennent du fait que l'articulation démarre et arrive avec une vitesse nulle :

$$\dot{\theta}(0) = 0, \dot{\theta}(t_f) = 0 \tag{II.5}$$

Et en combinant les deux fonctions :  $\theta(t)$  et  $\dot{\theta}(t)$  avec les quatre contraintes, on obtient quatre équations à quatre inconnues :

$$\left\{ \begin{array}{l} \theta_0 = a_0 \\ \theta_f = a_0 + a_1 \cdot t_f + a_2 \cdot t_f^2 + a_3 \cdot t_f^3 \\ a_1 = 0 \\ 0 = a_1 + 2 \cdot a_2 \cdot t_f + 3 \cdot a_3 \cdot t_f^2 \end{array} \right. \tag{II.6}$$

En résolvant le système d'équations on obtient :

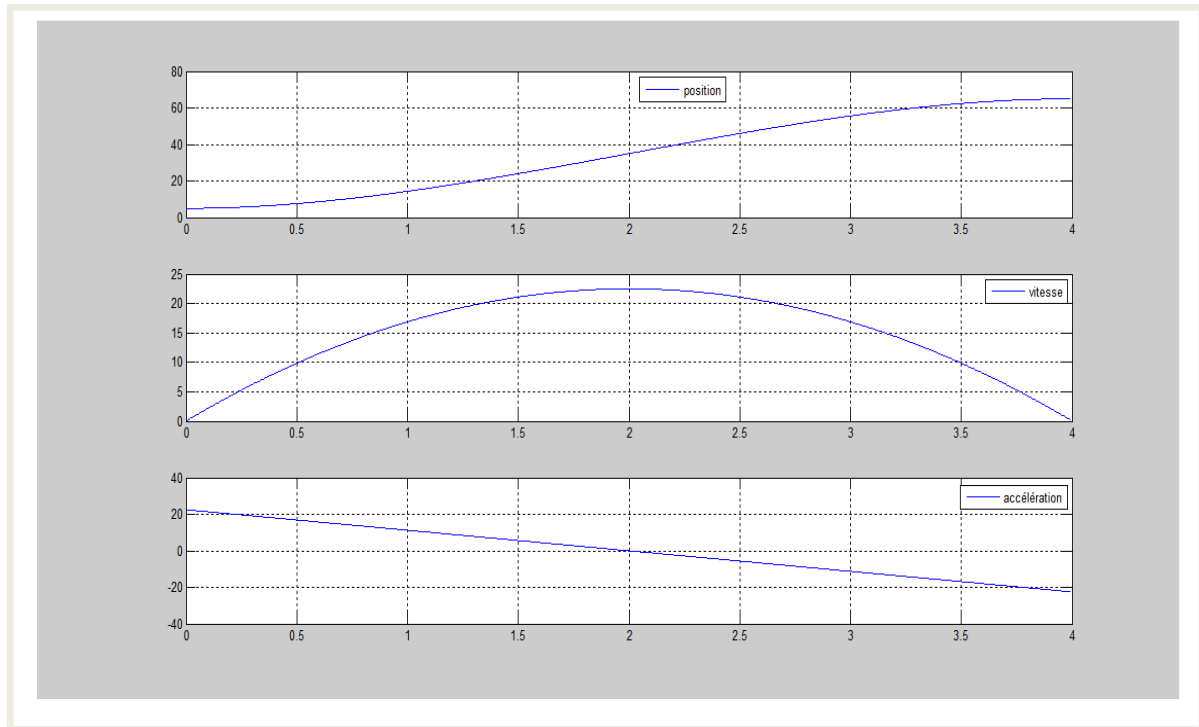
$$\left\{ \begin{array}{l} a_0 = \theta_0 \\ a_1 = 0 \\ a_2 = (3/t_f^2) (\theta_f - \theta_0) \\ a_3 = -(2/t_f^3) (\theta_f - \theta_0) \end{array} \right. \tag{II.7}$$

où,  $t_f$  la durée de mouvement;

Avec ces quatre coefficients, on peut calculer le polynôme cubique qui connecte n'importe quelle position désirée.

Exemple de génération d'une trajectoire articulaire :

On veut générer la trajectoire articulaire pour passer d'un angle initial de 5° à un angle final de 65° dans 4 secondes.



**Fig. II.1** Génération d'une trajectoire polynomiale

### II.2.2 Polynôme de degré cinq [8]:

On peut avoir des polynômes d'ordre supérieur pour la planification de la trajectoire, ils sont quelque fois utilisés dans le cas où on voudrait indiquer la position, la vitesse, et l'accélération au début et à la fin de la trajectoire et aussi sont utilisés quand les robots manipulateurs fonctionnent à une grande vitesse, il est nécessaire d'assurer la continuité des accélérations afin d'éviter l'excitation du mécanisme.

Il faut avoir un polynôme d'ordre cinq de la forme :

$$\theta(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 + a_4 \cdot t^4 + a_5 \cdot t^5 \quad (\text{II.8})$$

Pour déterminer les coefficients de ce polynôme il faut avoir au moins six contraintes, deux sur la position et deux sur la vitesse et deux autres sur l'accélération :

$$\left\{ \begin{array}{l} \theta_0 = a_0 \\ \dot{\theta} = a_1 \\ \ddot{\theta} = 2 \cdot a_2 \\ \theta_f = a_0 + a_1 \cdot t_f + a_2 \cdot t_f^2 + a_3 \cdot t_f^3 + a_4 \cdot t_f^4 + a_5 \cdot t_f^5 \\ \dot{\theta}_f = a_1 + 2 \cdot a_2 \cdot t_f + 3 \cdot a_3 \cdot t_f^2 + 4 \cdot a_4 \cdot t_f^3 + 5 \cdot a_5 \cdot t_f^4 \\ \ddot{\theta}_f = 2 \cdot a_2 + 6 \cdot a_3 \cdot t_f + 12 \cdot a_4 \cdot t_f^2 + 20 \cdot a_5 \cdot t_f^3 \end{array} \right. \quad (\text{II.9})$$

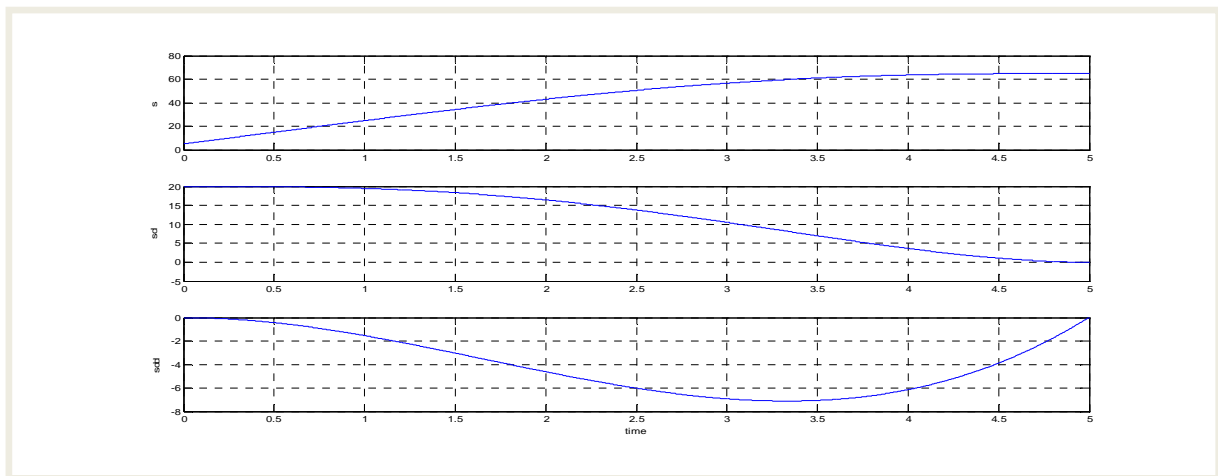
En résolvant le système d'équations (II.8), on trouve :

$$\left\{ \begin{array}{l} a_0 = \theta_0 \\ a_1 = \dot{\theta}_0 \\ a_2 = \ddot{\theta}_0 \\ a_3 = [20 \cdot (\theta_f - \theta_0) - (8 \cdot \dot{\theta}_f + 12 \cdot \dot{\theta}_0) \cdot t_f + (3 \cdot \ddot{\theta}_0 - \ddot{\theta}_f) \cdot t_f^2] / 2 \cdot t_f^2 \\ a_4 = [-30 \cdot (\theta_0 - \theta_f) + (14 \cdot \dot{\theta}_0 + 16 \cdot \dot{\theta}_f) \cdot t_f - (3 \cdot \ddot{\theta}_0 - 2 \cdot \ddot{\theta}_f) \cdot t_f^2] / 2 \cdot t_f^4 \\ a_5 = [-12 \cdot (\theta_f - \theta_0) - 6 \cdot (\dot{\theta}_f + \dot{\theta}_0) \cdot t_f - (\ddot{\theta}_f - \ddot{\theta}_0) \cdot t_f^2] / 2 \cdot t_f^5 \end{array} \right. \quad (\text{II.10})$$

où,  $t_f$  la durée de mouvement;

Exemple de génération d'une trajectoire articulaire :

On veut générer la trajectoire articulaire pour passer d'un angle initial de  $5^\circ$  à un angle final de  $65^\circ$  dans 4 secondes et de vitesse initial de  $20^\circ/\text{s}$ , de accélération de  $0^\circ/\text{s}^2$ .



**Fig. II.2** Génération d'une trajectoire polynomiale de degré cinq.

II.2.3 mouvement continu entre points via :

Dans le cas de mouvement continu aux points via, les fonctions linéaires connectent les points via et les segments paraboliques sont ajoutés autour de ces derniers.

Considérons 3 points correspondant aux angles  $(\theta_j, \theta_k, \theta_l)$  avec les notations suivantes :

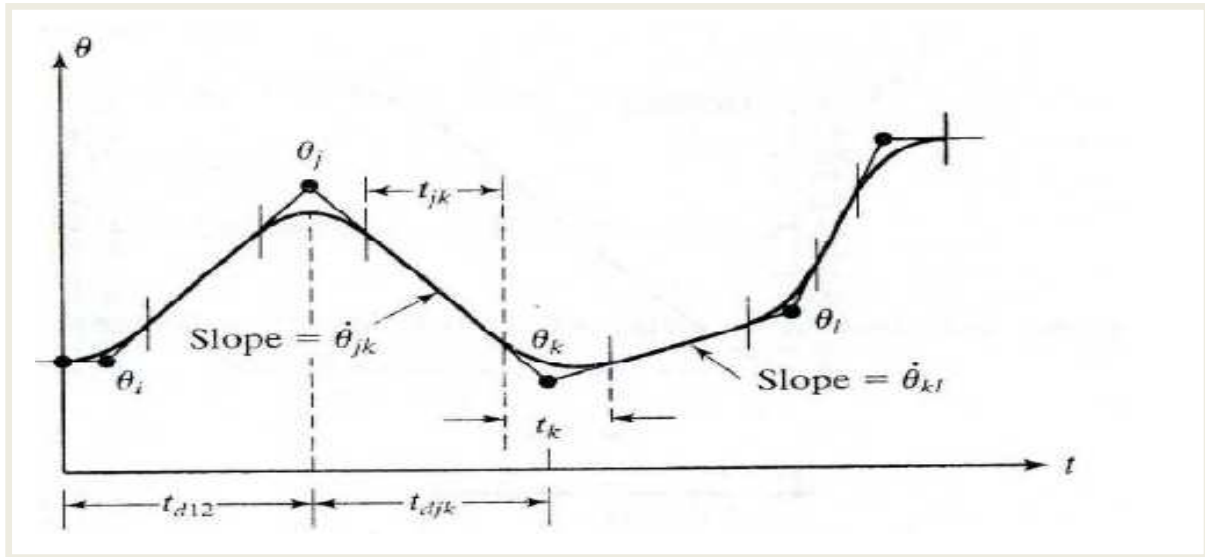


Fig. II.3 Génération d'une trajectoire avec un mouvement continu entre points via.

Etant donnés tous les points  $\theta_k$ , les durées désirées  $t_{djk}$  et l'amplitude de l'accélération  $|\ddot{\theta}_k|$ , on peut calculer alors les durées des extensions paraboliques  $t_k$  et les autres grandeurs. Pour les points via, on a :

$$\left\{ \begin{array}{l} \dot{\theta}_{jk} = \theta_k - \theta_j / t_{djk} \\ \ddot{\theta}_k = \text{sgn}(\dot{\theta}_{kl} - \dot{\theta}_{jk}) * |\ddot{\theta}_k| \\ t_k = \dot{\theta}_{kl} - \dot{\theta}_{jk} / \ddot{\theta}_k \\ t_{jk} = t_{djk} - \frac{1}{2} * t_j - \frac{1}{2} * t_k \end{array} \right. \quad (\text{II.11})$$

Pour les segments initial et final, on considère la durée entière de la région parabolique dans le temps global. Pour le premier segment, on calcule :

$$\left\{ \begin{array}{l} \dot{\theta}(t_1) = \ddot{\theta}_1 * t_1 = (\theta_2 - \theta_1) / (t_{d12} - \frac{1}{2} * t_1) \\ \ddot{\theta}_1 = \text{sgn}(\theta_2 - \theta_1) * |\ddot{\theta}_1| \end{array} \right. \quad (\text{II.12})$$

De (II. 11), on forme une équation de second ordre et on obtient la solution :

$$\begin{cases} t_1 = t_{d12} - \sqrt{t_{d12}^2 - 2 * (\theta_2 - \theta_1) / \ddot{\theta}_1} \\ \dot{\theta}_{12} = (\theta_2 - \theta_1) / (t_{d12} - \frac{1}{2} * t_2) \\ t_{12} = t_{d12} - t_1 - \frac{1}{2} * t_2 \end{cases} \quad (\text{II.13})$$

De la même façon, on obtient pour le segment final [(n-1), n] :

$$\begin{cases} \ddot{\theta}_n * t_n = (\theta(n-1) - \theta_n) / (t_d(n-1)n - \frac{1}{2} * t_n) \\ \ddot{\theta}_n = \text{sgn}(\theta(n-1) - \theta_n) * |\ddot{\theta}_n| \end{cases} \quad (\text{II.14})$$

Ce qui conduit à la solution :

$$\begin{cases} t_n = t_d(n-1)n - \sqrt{t_d(n-1)n^2 - 2 * (\theta_n - \theta(n-1)) / \ddot{\theta}_n} \\ \dot{\theta}(n-1)n = (\theta_n - \theta(n-1)) / (t_d(n-1)n - \frac{1}{2} * t_n) \\ t(n-1)n = t_d(n-1)n - t(n-1) - \frac{1}{2} * t_n \end{cases} \quad (\text{II.15})$$

En utilisant cet ensemble d'équations, on peut calculer les temps et les vitesses associés pour une trajectoire multi-segments.

Exemple de génération d'une trajectoire articulaire :

On veut générer la trajectoire articulaire pour passer d'un angle initial de 10° à un angle final de 5° dans 6 secondes on passe par deux points via (35° et 25°), avec continuité de l'accélération sur la partie parabolique.

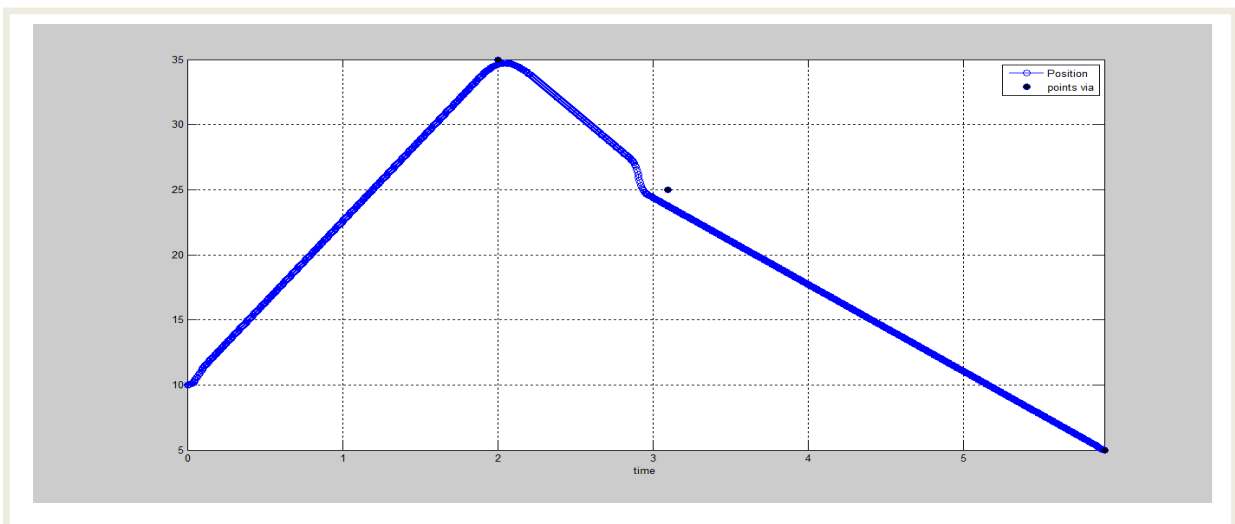


Fig. II.4 Génération d'une trajectoire avec un mouvement continu entre points via.



### II.2.4 Commande opérationnelle [9]:

Les commandes dans l'espace opérationnel contrôlent directement les coordonnées opérationnelles. Contrairement aux asservissements dans l'espace articulaire où les  $n$  entrées et les  $n$  sorties sont respectivement les consignes et les mesures de position articulaire.

Dans la commande opérationnelle, les consignes et les mesures sont exprimées en coordonnées cartésienne. Avec un tel asservissement, il est nécessaire d'effectuer une transformation des commandes du correcteur pour les convertir dans l'espace articulaire afin de pouvoir les appliquer aux actionneurs. De cette transformation (qui utilise le Jacobéen inverse) résulte des non linéarités supplémentaires.

### II.3 Contrôleur de robot [9]:

Le contrôleur de robot doit fonctionner sous un système temps-réel afin de garantir une période d'échantillonnage constante ( $< 5$  ms en général). Ses fonctions sont :

- Echantillonner les mesures
- Gérer les sécurités bas-niveau
- Calculer la commande des asservissements de positions articulaires.

### II.4. Les actionneurs principaux [7]:

Les actionneurs principaux qui sont actuellement utilisés en robotique peuvent être classés en trois catégories selon que la source d'énergie qu'ils utilisent et qu'ils transforment :

- Les actionneurs électriques ;
- Les actionneurs hydrauliques ;
- Les réducteurs

#### II.4.1 Les actionneurs électriques:

Trois types d'actionneurs électriques utilisés en robotique sont :

- Le moteur à courant continu ;
- Le moteur synchrone ;
- Le moteur asynchrone ;

Les moteurs électriques de robotique ont des particularités liées à leur utilisation (performances maximales) :

- Constante de temps électrique ( $L/R$ ) faible ;
- Constante de temps mécanique ( $J/f$ ) faible ;

Avec  $L$ , inductance de l'induit,  $R$ , résistance de l'induit,  $J$  inertie du rotor et  $f$ , frottements visqueux sur le rotor.  $R$  et  $f$  ne peuvent être augmentés car il faut limiter les pertes. D'où  $L$  et

$J$  sont minimisés pour obtenir de bonnes performances dynamiques. Ceci impose un dimensionnement particulier du moteur :

- Moteur long et de diamètre faible ou moteur à rotor plat (type AXEM) pour diminuer  $J$ .
- Conducteurs de forte section pour diminuer  $L$ .

#### II.4.1.1 Le moteur à courant continu (MCC):

Le stator crée un champ magnétique fixe. En robotique ce champ est créé par un aimant permanent. Un système de balais permet d'amener le courant au rotor. Ce courant (continu) crée un champ en quadrature par rapport à celui du stator générant ainsi un couple proportionnel à ce courant.

Historiquement, le MCC fut le premier à être utilisé en robotique. Son principal avantage est qu'il est facile à commander (couple proportionnel au courant d'induit). Son principal inconvénient est qu'il utilise des contacts glissants (maintenance régulière).

#### II.4.1.2 Le moteur synchrone:

On le trouve souvent sous l'appellation "*brushless*". Un champ est créé au rotor (en robotique par un aimant permanent). Les bobines du stator sont alimentées par un courant triphasé de fréquence variable dont la phase est telle qu'il crée un champ en quadrature.

Ce courant triphasé est généré par un variateur en fonction de la position du rotor afin de toujours garantir la quadrature entre les 2 champs du rotor et du stator.

Le moteur synchrone a des performances dynamiques meilleures que le MCC et requiert moins d'entretien (pas de balais). Par contre, le variateur est plus complexe.

#### II.4.1.3 Le moteur asynchrone :

Les bobines du stator sont alimentées par un courant triphasé de fréquence variable. Celui-ci crée dans les bobines du rotor un courant induit qui lui-même crée un champ. Toute la difficulté de la commande, aussi appelée *commande vectorielle*, est que le champ du rotor soit à tout moment en quadrature avec le champ du stator.

Le moteur asynchrone à commande vectorielle a les meilleures performances dynamiques. Néanmoins le variateur est plus complexe.

**II.4.1.4 Tableau comparatif des actionneurs électriques:**

	MCC	Synchrone	Asynchrone
Maintenance	***	**	*
Rapport couple/masse	*	**	***
Performances dynamiques	*	**	***
Complexité de la commande	*	**	***
Coût variateur + moteur	**	**	***

*	Peu
**	Moyen
***	Beaucoup

**Fig.II.5** Tableau comparatif des actionneurs électriques

**II.4.2 Les actionneurs hydrauliques :**

Les actionneurs hydrauliques sont rarement employés en robotique. Pour des applications spéciales, il peut être nécessaire d'y recourir. Le principal avantage de ce type d'actionneur est le grand rapport effort/masse ce qui permet de réaliser des robots légers et puissants. Le principal inconvénient est la difficulté pour le commander. Ce type d'actionneur est très imprécis.

**II.4.3 Les actionneurs pneumatiques**

Les actionneurs pneumatiques sont utilisés pour les outils des robots. Ils ne sont jamais utilisés pour déplacer des axes.

**II.4.4 Les réducteurs:**

Les réducteurs mécaniques ont pour but de modifier le rapport de vitesse ou/et le couple entre l'axe d'entrée et l'axe de sortie d'un mécanisme. L'usage d'un réducteur est rendu nécessaire pour réduire la vitesse de rotation des moteurs électriques qui est généralement de 1500 tr/mn. On distingue plusieurs types de réducteurs :

- Réducteurs d'efforts physiques ;

- Réducteurs de vitesses ;
- Orientation des axes.

## II.5 Techniques de commande [11]:

On a le choix entre plusieurs techniques de commande. On distingue :

- **1) Commande dans l'espace libre**
  - Commande par articulation ;
  - Commande jacobéenne ;
  - Commande par découplage non linéaire ;
  - Commande fondée sur une fonction de Lyapunov ;
  - Commande passive ;
  - Commande adaptative ;
  - Commande prédictive ;
  - Commande robuste ;
  - **Commande optimale.**

Celle-ci est l'objet de notre travail.

- **2) Commande dans l'espace contraint.**

## II.6 Problème de la commande optimale [12,13]:

L'objectif de la commande optimale est de déterminer une commande optimale  $u^*(t)$  ou  $u^*(x, t)$  qui force le système à satisfaire des contraintes physiques tout en minimisant ou maximisant un indice de performance  $J$ .

Le problème fondamental de la commande optimale consiste en les trois éléments suivants :

**1.** Un modèle mathématique du système à contrôler qui peut être décrit par un ensemble d'équations d'état de la forme :

$$\dot{x}(t) = f(x(t), u(t), t) \quad (\text{II.16})$$

où  $x(t) \in R^n$  est le vecteur d'état,  $u(t) \in R^m$  est le vecteur de commande,  $f$  est continûment différentiable par rapport à toutes ses arguments.

**2.** Un ensemble de conditions aux limites sur les variables d'état qui donne la valeur des états du système à l'instant initial.

$$x(t_0) = x_0 \quad (\text{II.17})$$

3. Un indice de performance qui doit être minimisé (ou maximisé). L'indice de performance décrit quelques spécifications désirées exprimées mathématiquement sous forme scalaire. L'indice de performance qui nous intéresse peut être formulé comme suit :

$$J = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t) dt \quad (\text{II.18})$$

Où :  $\varphi$  et  $L$  sont des fonctions scalaires, continûment différentiables par rapport à toutes leurs variables. Ces fonctions sont sélectionnées compte tenu des objectives de performances.

Un problème de contrôle optimal se décompose en deux parties : pour déterminer une trajectoire optimale joignant un ensemble initial à une cible, il faut d'abord savoir si cette cible est atteignable. C'est le *problème de contrôlabilité*. Ensuite, une fois ce problème résolu, il faut chercher parmi toutes ces trajectoires possibles celles qui le font en *coût minimal*.

**II.6.1 Rapport de la commande optimale [12]:**

Le problème de la commande optimale sans contraintes peut être énoncé comme suit : Etant donné  $f, x_0, t_0, t_f, \varphi$ , et  $L$ , il s'agit de trouver  $u^*(t)$  ou  $u^*(x, t)$  qui minimise l'indice de performance :

$$J = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t) dt \quad (\text{II.19})$$

sachant que :

$$\dot{x}(t) = f(x(t), u(t), t) \quad x(t_0) = x_0 \quad (\text{II.20})$$

Ce problème peut être résolu par l'une des méthodes suivantes :

- La méthode de la programmation dynamique de Bellman qui est basée sur le principe d'optimalité (L'équation de Hamilton-Jacobi-Bellman (HJB)).
- La méthode variationnelle et le principe du minimum de Pontryagin (équation d'Euler-Lagrange).
- Les méthodes directes utilisant la discrétisation ou la paramétrisation.

En général, il n'est pas possible de résoudre analytiquement le problème (II.19), (II.20). Cependant une solution analytique existe pour un cas particulier de ce problème qui s'énonce comme suit : Trouver la commande optimale qui minimise la fonctionnelle :

$$J = x^T(t_f) S x(t_f) + \int_{t_0}^{t_f} (x^T Q x + u^T R u) dt \quad (\text{II.21})$$

sachant que :

$$x(t) = A(t)x(t) + B(t)u(t) \quad x(t_0) = x_0 \quad (\text{II.22})$$

où  $S$  et  $Q$  sont des matrices semi définies positives et  $R$  est une matrice définie positive. Pour ce problème, la solution peut être exprimée en boucle fermée par :

$$u^*(x, t) = -R^{-1}B^T(t)P(t)x(t) \quad (\text{II.22})$$

où  $p(t)$  est la solution de l'équation de Riccati.

**II.6.2 Contrôlabilité [13]:**

On veut répondre à la question suivante : étant donné le système (II.16), où peut-on aller en temps  $T$  en faisant varier le contrôle  $u$  ? On rappelle tout d'abord la notion d'ensemble accessible. L'ensemble accessible en temps  $T$  pour le système (II.16), noté  $acc(x_0, T)$ , est l'ensemble des extrémités au temps  $T$  des solutions du système partant de  $x_0$  au temps  $t = 0$ .

**II.6.3 Existence de trajectoires optimales [13]:**

Au problème de contrôle s'ajoute un second problème, celui de la minimisation. Parmi toutes les solutions reliant  $x_0$  à  $x_1$ , trouver une trajectoire qui minimise la fonction cout  $J(T, u)$ .

Une telle trajectoire, si elle existe, est dite optimale pour ce coût. L'existence de trajectoires optimales dépend de la régularité du système et du coût. Il se peut aussi qu'un contrôle optimal n'existe pas dans la classe de contrôles considérés, mais existe dans un espace plus gros. En particulier on a intérêt à travailler dans un espace de contrôles complet et qui ait de bonnes propriétés de compacité : voilà pourquoi à nouveau l'espace  $L^2$  est intéressant.

**II.7 Conditions nécessaires d'optimalité [12] :**

**II.7.1 Equations d'Euler-Lagrange :**

Les conditions nécessaires d'optimalité peuvent être déduites à partir des méthodes de calculs variationnels qui sont basées sur le fait qu'à chaque point stationnaire la variation de la fonction coût disparaît pour une variation arbitraire de la commande .

Pour résoudre le problème (II.19) et (II.20), nous utiliserons les multiplicateurs de Lagrange,  $\lambda(t) \in R^n$  . Par conséquent l'indice de performance est donné par :

$$J_A = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} [L(x(t), u(t), t) + \lambda^T(t)(f(x(t), u(t), t) - \dot{x})]dt \quad (\text{II.23})$$

Introduisons le Hamiltonien  $H$  défini par :

$$H(x(t), u(t), \lambda(t), t) = L(x(t), u(t), t) + \lambda^T(t) f(x(t), u(t), t)$$

Nous pouvons, ainsi, écrire (II.23) sous la forme :

$$J_A = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} [H(x(t), u(t), \lambda^T(t), t) - \int_{t_0}^{t_f} \lambda^T \dot{x}(t)] dt \quad (II.24)$$

L'intégration du dernier terme à droite par partie conduit à :

$$\int_{t_0}^{t_f} \lambda^T \dot{x}(t) = \lambda^T(t_f)x(t_f) - \lambda^T(t_0)x(t_0) - \int_{t_0}^{t_f} \dot{\lambda}^T(t)x(t) \quad (II.25)$$

Cependant (II.24) devient :

$$J_A = \varphi(x(t_f), t_f) - \lambda^T(t_f)x(t_f) + \lambda^T(t_0)x(t_0) + \int_{t_0}^{t_f} [H(x(t), u(t), \lambda^T(t), t) + t_0 t_f \lambda^T x] dt \quad (II.26)$$

Le problème original, (II.19) et (II.20), a été converti à un problème de minimisation, (II.26), sans contraintes.

Puisque  $\lambda(t)$  est arbitraire, nous pouvons la choisir telle que :

$$\dot{\lambda}^T(t) = -\frac{\partial H}{\partial x} = -\lambda^T \frac{\partial f}{\partial x} - \frac{\partial L}{\partial x} \quad (II.27)$$

avec la condition finale :

$$\lambda^T(t_f) = -\frac{\partial \varphi}{\partial x} \Big|_{t=t_f} \quad (II.28)$$

### II.7.2 Principe du minimum de Pontryagin :

Dans les problèmes réels, les variables de commande sont, d'habitude, bornées, donc le Hamiltonien ne peut pas être différenciée par rapport à la commande. Dans ce cas les conditions nécessaires sont dérivées par le principe du minimum de Pontryagin qui s'énonce ainsi :

Supposons que  $u^*(t)$  est la commande optimale et  $x^*(t)$  est la trajectoire optimale correspondante, donc il doit exister un vecteur  $\lambda^*(t)$  telles que :

$$\dot{\lambda}^T = -\frac{\partial H}{\partial x} \quad \lambda^T(t_f) = \left(\frac{\partial \varphi}{\partial x}\right)^T$$

$$H(x^*, x^*, u^*, \lambda^*, t) \leq H(x^*, x^*, u, \lambda^*, t) \quad (II.29)$$

pour tout  $t \in [t_0 \quad t_f]$  et pour tout  $u(t) \in U \subset R^m$ . Donc la commande optimale doit minimiser le Hamiltonien défini par (II.25).

L'inégalité (II.29) est très utilisée pour l'obtention de la commande optimale si la commande est bornée. Il est à noter que le principe du minimum est une généralisation de l'approche du calcul variationnel. Il faut aussi signaler que l'approche variationnel et le principe du minimum mènent à un problème connu comme « A non linear two-point boundary problem : NLTPBP » qui est difficile à résoudre de façon analytique. Les méthodes numériques proposées pour résoudre ces problèmes sont appelées méthodes indirectes par opposition aux méthodes directes basées sur la transformation du problème de la commande optimale à un problème de programmation non linéaire.

## IV.8 Méthode de résolutions du problème de commande optimale [12] :

### II.8.1 Méthodes indirectes :

Ces méthodes sont divisées en deux catégories : les méthodes en boucle fermée et les méthodes en boucle ouverte :

#### II.8.1.1 Méthodes en boucle fermée :

Parmi les méthodes proposées pour obtenir la commande optimale en boucle fermée citons les approches suivantes:

- La première approche pour obtenir une approximation de la commande optimale en boucle fermée est basée sur l'utilisation du développement en séries de puissance pour résoudre soit l'équation de HJB soit le NLTPBP.
- La seconde approche pour obtenir la commande optimale en boucle fermée est basée sur la linéarisation des conditions nécessaires d'optimalité autour de la solution optimale ou via le développement de l'indice de performance jusqu'à l'ordre 2 et les contraintes jusqu'à l'ordre 1 autour de la solution optimale .
- La troisième approche est basée sur l'écriture des équations d'état sous forme linéaire.
- La quatrième approche est basée sur la solution du problème inverse de la commande optimale .
- D'autres approches existent également.

#### II.8.1.2 Méthodes de commande en boucle ouverte :

Ces méthodes sont basées sur la résolution du NTPBVP. Parmi ces méthodes, on peut citer entre autres : les méthodes du gradient, les méthodes de quasi linéarisation, les méthodes de la fonction de pénalité etc. Ce sont des méthodes standard pour résoudre les problèmes de la commande optimale.



**II.8.2 Méthodes directes :**

Ces méthodes offrent quelques *avantages* quand elles s'appliquent aux problèmes de la commande optimale. Le premier avantage est que le problème de la commande optimale dynamique peut être converti à un problème d'optimisation statique qui est facile à résoudre par rapport au problème original. Le deuxième avantage est qu'il y a des algorithmes bien développés pour résoudre les problèmes de programmation non linéaire. Le troisième avantage est la possibilité de traiter facilement différents types de contraintes. Par conséquent plusieurs auteurs ont utilisé les méthodes directes pour résoudre le problème de la commande optimale.

**II.8.2.1 Méthodes de discrétisation :**

Toutes les techniques de discrétisation divisent l'intervalle de temps en  $n$  segments :

$$t_0 < t_1 < t_2 < \dots < t_n = t_f \quad (\text{II.30})$$

où les points  $t_i$  sont appelés des nœuds.

Une approche qui applique cette méthode discrétise les variables d'état et les variables de contrôle ; nous avons, donc, la séquence des valeurs inconnues des variables d'état et de contrôle suivante :

$$z = (x_0, x_1, \dots, x_n, u_0, u_1, \dots, u_{n-1}) \quad (\text{II.31})$$

et le système d'équations d'état est remplacé par un ensemble d'équations algébriques qui peuvent être considérées comme des contraintes d'égalité. Ce problème peut, ainsi être résout en utilisant les techniques de la programmation non linéaire. L'inconvénient majeur de cette approche est la dimension élevée du vecteur  $z$ .

Une autre approche possible est de discrétiser la variable de contrôle seule.

$$z = (u_0, u_1, \dots, u_{n-1}) \quad (\text{II.32})$$

et d'intégrer ensuite le système d'équations d'état pour trouver les variables d'état en fonction des variables de contrôle .

**II.8.2.2 Méthodes de paramétrisation :**

La technique de para-métrisation peut être appliquée de trois manières différentes :

**- Paramétrisation de la Commande :**

Cette technique est basée sur l'approximation des variables de la commande en choisissant une structure appropriée de la forme :

$$u_k(t) = \sum_{i=0}^N C_i^{u_k} \varphi_i(t) \quad k = 1, 2, \dots, m \quad (II.33)$$

où les  $C_i^{u_k}$  sont les paramètres inconnus et les  $\varphi_i$  dénotent un ensemble approprié de fonctions formant une base de l'espace de commande à dimension finie. Les variables d'état sont obtenues en fonction des paramètres inconnus en intégrant les équations d'état du système, et en substituant les variables approximatives de la commande et les variables d'état correspondantes dans l'indice de performance, le problème de la commande optimale est ainsi converti à un problème de programmation non linéaire à paramètres statiques qu'il est facile à résoudre que le problème original.

Plusieurs fonctions ont été utilisées pour l'approximation des variables de la commande, parmi lesquelles on peut citer : les fonctions constantes par morceaux, les fonctions linéaires par morceaux, les fonctions splines d'ordre donné. L'application de cette technique nécessite l'intégration des équations d'état qui est un processus coûteux en termes de temps de calcul.

**- Paramétrisation d'Etat-Commande :**

Cette approche est basée sur l'approximation, en même temps, des variables d'état et des variables de commande par une séquence de fonctions connues avec des paramètres inconnus :

$$u_k(t) = \sum_{i=0}^N C_i^{u_k} \varphi_i(t) \quad k = 1, 2, \dots, m \quad (II.34)$$

$$x_j(t) = \sum_{i=0}^N C_i^{x_j} \varphi_i(t) \quad j = 1, 2, \dots, m \quad (II.35)$$

où les  $C_i^{u_k}, C_i^{x_j}$  sont des paramètres inconnus et les  $\varphi_i$  sont un ensemble de fonctions appropriées. En utilisant cette méthode, le problème de la commande optimale est converti à un problème de programmation mathématique non linéaire. L'inconvénient principal de cette approche est le nombre important de paramètres que l'on a à déterminer. Cependant, en utilisant cette approche nous aboutissons à un problème de programmation non linéaire à dimension très élevées en terme du nombre de paramètres inconnus et du nombre de contraintes d'égalité.

**- Paramétrisation d'Etat :**

L'idée de cette technique est de faire l'approximation seulement de la variable d'état du système .

$$x_j(t) = \sum_{i=0}^N C_i^{x_j} \varphi_i(t) \quad j = 1, 2, \dots, m \quad (II.36)$$

Les  $C_i^{x_j}$  sont les paramètres inconnus et les  $\varphi_i$  sont définies comme précédemment. Dans cette technique les variables de commande sont obtenues à partir des équations d'état. Cette technique a quelques avantages :

- On n'a pas besoin d'intégrer les équations d'état.
- Le nombre de paramètres inconnus est faible par rapport aux autres techniques.
- Les paramètres peuvent être manipulés directement.

Cependant, cette technique souffre de quelques inconvénients :

- Il est difficile de traiter les systèmes non linéaires car il n'est pas toujours facile de trouver les variables de commande comme fonction des variables d'état.
- Il n'existe pas une voie systématique pour appliquer cette technique aux problèmes de la commande optimale à aspect général quand le nombre de variable d'état et de commande n'est pas égal.

**II.9 Méthodes générales d'optimisation [14]:**

Maintes fois abordées par plusieurs recherches, il existe bien évidemment un très grand nombre de dérivées de ces techniques, tant le sujet a déjà été étudié et continue de l'être actuellement.

**II.9.1 Définition de l'optimisation :**

Optimiser une fonction continue  $f : S \in R^n \rightarrow R$  consiste à déterminer tous les  $x \in S$  pour lesquels  $f$  atteint son maximum (ou son minimum). On ne parlera ici que de maximisation, mais on remarque au passage que pour minimiser une fonction, il suffit simplement de maximiser son opposé.

Il existe deux types d'optimisation : l'optimisation locale et l'optimisation globale. Le but de l'optimisation locale est de trouver un point qui est un optimum de  $f$  dans un certain voisinage, c'est-à-dire :

$$x \text{ minimum local} \iff \exists \text{ voisinage } V \text{ de } x \text{ tel que } f(x) \leq f(y), \forall y \in V$$

Par contre,  $x$  est un minimum global de  $f$  sur  $S$  si son image par  $f$  est inférieure ou

égale à celle de tous les autres points du domaine  $S$ , et pas seulement dans un certain voisinage. Bien évidemment, tout minimum global est aussi un minimum local.

## II.9.2 Optimisation locale :

### II.9.2.1 Méthode de descente :

La méthode de descente est une méthode ancienne et très efficace lorsqu'il s'agit de résoudre des problèmes difficiles. Elle se base sur la notion de voisinage et fonctionne de la manière suivante :

Etant donné la solution  $s \in S$  que l'on possède, on définit l'ensemble de ses voisins par  $N(s) \subset S$ . On choisit alors un  $s' \in N(s)$  tel que  $f(s') > f(s)$ . On recommence ce processus à partir de  $s'$ , jusqu'à ce que le point où l'on se trouve n'ait plus aucun voisin qui soit meilleur.

Le terme "de descente" vient du fait que dans la littérature, le but est en général de minimiser une fonction : à chaque itération, on choisit une solution qui a une plus petite image, et on "descent" donc progressivement vers le minimum. Quant au choix du voisin, il varie en fonction des méthodes. On peut par exemple prendre le premier que l'on trouve qui est un peu meilleur, ou alors tous les énumérer afin de prendre le meilleur d'entre eux.

### II.9.2.2 Algorithme génétique :

Le principe de l'algorithme génétique est le suivant : étant donné une population de solution  $P$ , on choisit deux éléments  $x, y \in P$  que l'on utilise pour générer une nouvelle solution  $z = h(x, y)$  qui est ajoutée à  $P$ . On répète cette opération  $N$  fois afin d'obtenir un nouvel ensemble de solutions  $P \subset P'$ . Ensuite, on élimine aléatoirement  $N$  éléments de  $P'$ , après quoi on réitère à nouveau tout le processus.

Lorsque l'on procède à l'élimination de certains éléments de  $P'$ , on tient compte de leur qualité, c'est-à-dire que l'on va privilégier, d'un point de vue probabiliste, l'élimination de solutions dont l'image par  $f$  est petite relativement aux autres. La fusion  $h$  contient quant à elle un effet aléatoire, ce qui entraîne que si  $x$  et  $y$  sont deux "bonnes" solutions, il se peut tout de même que  $h(x, y)$  soit "mauvaise".

## II.9.3 Optimisation globale :

### II.9.3.1 Méthode des variations locales :

La méthode des variations locales est relativement simple à implémenter et fonctionne comme suit :

Etant donné un pas  $\rho > 0$  et un point  $(x^0, \dots, x_n^0) \in R^n$ , on choisit, parmi les trois points  $\{(x_1^0 - \rho, \dots, x_n^0), (x_1^0, \dots, x_n^0), (x_1^0 + \rho, \dots, x_n^0)\}$ , celui dont l'image est la plus grande. Puis, à partir du point choisi, on exécute la même opération sur la deuxième composante, et ainsi de suite. Après avoir traité la dernière composante, on recommence avec la première, jusqu'à ce que l'on soit satisfait du point obtenu.

### II.9.3.2 Branch and bound:

La méthode "branch and bound" divise le problème considéré en sous-problèmes plus faciles à étudier. En fait, il s'agit de scinder l'espace de recherche en sous-espaces, et de borner les solutions dans chaque sous-espace. Voici comment cela se passe en pratique : le problème de départ constitue le problème racine de l'arbre de recherche. On calcule une évaluation de la solution recherchée en essayant de borner inférieurement et supérieurement les solutions du domaine entier. Si les bornes inférieure et supérieure coïncident, la solution optimale est déterminée. Sinon, on divise le domaine de recherche en deux sous-problèmes, que l'on va étudier séparément.

Si à un quelconque moment, on trouve une borne supérieure qui est plus petite que la borne inférieure d'un domaine situé "plus haut" dans l'arborescence, on peut arrêter d'explorer cette branche de l'arbre de recherche puisque toute solution que l'on y trouvera sera forcément moins bonne que celle que l'on a déjà. On continue ainsi jusqu'à ce que toutes les branches de l'arbre aient été étudiées.

### II.9.3.3 Branch and prune :

Cette méthode, introduite en 1997, est similaire à la précédente dans le sens où il y a création d'un arbre de recherche, par division du problème initial en sous-problèmes. Il s'agit en fait là de tester si un domaine est consistant, c'est-à-dire susceptible d'être intéressant, en l'éliminant s'il ne l'est pas et en le divisant s'il l'est mais qu'il n'est pas suffisamment précis pour être analysé. Par exemple, un ensemble sera inconsistant s'il ne contient aucun point dont la dérivée s'annule. Il est en effet inutile dans ce cas de s'y attarder, puisqu'il est impossible que cet ensemble contienne un quelconque maximum (la fonction à maximiser est supposée lisse, donc en particulier différentiable). Si par contre, on sait qu'il en contient un mais qu'il est encore impossible de l'identifier précisément, alors on sépare l'intervalle en deux parties, comme c'était le cas dans la méthode branch and bound.

**II.9.4 Optimisation stochastique :**

On va considérer, par ailleurs, toujours le problème de minimum suivant :

$$\min_{x \in S} f(x) \quad (\text{II.37})$$

où  $f$  est une fonction continue sur  $S \in \mathbb{R}^n \rightarrow \mathbb{R}$ .

**II.9.4.1 Méthode adaptative :**

Le grand inconvénient des méthodes proposées jusqu'à présent est qu'elles ne tirent quasiment aucun profit des itérations antérieures, ce qui constitue un gaspillage d'information important. Le principe de la méthode adaptative est, pour les itérations futures, de se concentrer uniquement sur des points dont l'image est supérieure à un minimum local que l'on a atteint. Plus clairement :

A partir d'un point initial généré aléatoirement sur l'espace de définition de la fonction, on effectue une recherche du minimum local le plus proche, via la méthode de Newton disponible sous Matlab (fonction `fmincon`). Depuis ce minimum, on génère des points sur une demi-droite de direction aléatoire, et on regarde si l'un d'entre eux possède une image supérieure à notre minimum. Si ce n'est pas le cas, on observe des points dans une autre direction, et ainsi de suite jusqu'à ce que l'on trouve un meilleur point.

Une nouveauté de cette méthode, hormis le fait que l'on effectue une recherche locale, est que l'on n'approxime plus du tout le gradient pour tirer la demi-droite. La raison est que puisque pour le point du part duquel la demi-droite est un minimum local, le gradient est nul. C'est pour cela que la direction de la demi-droite est choisie aléatoirement.

**II.9.5 Considération du gradient [14]:**

Le gradient joue un rôle essentiel dans l'optimisation. Dans le cadre des méthodes d'optimisation, il sera également important d'analyser le comportement de la fonction objective dans certaines directions.

Quand on connaît les premières dérivées, il serait naturel de suivre la direction inverse du gradient pour chercher un minimum, puisque c'est dans cette direction que la fonction décroît le plus (cette technique à été utilisée par Cauchy au XIX<sup>e</sup> siècle).

La méthode de plus grande pente consiste en une série de minimisation unidimensionnelle, chacune suivant la direction de la plus grande pente ou la minimisation commence. Bien sur, la direction du gradient n'est pas constante, on s'attend à de multiples

itérations pour trouver le minimum, mais on montre que la méthode est convergente pour une fonction quadratique.

Note : la direction de plus grande pente est évidemment une combinaison de variables. On y remarque une propriété des directions successives de recherche : si chaque minimisation par recherche linéaire est exacte, les directions successives sont orthogonales, ce qui n'est pas optimale pour une recherche efficace. Dans cette méthode, on part de  $x_0$  (vecteur initial) et on calcule le vecteur gradient :

$$g_i = \frac{\partial f(x_i)}{\partial x} \text{ au point } x_i \quad (\text{II.38})$$

On effectue la recherche linéaire du scalaire  $t$  qui rend  $f$  minimum dans la direction  $-g_i$  ('Linear Search'), soit :

$$x_{i+1} = x_i - t_i g_i \quad (\text{II.39})$$

Puis on réitère en recalculant le gradient en ce point. Il est à remarquer que la nouvelle direction de descente est perpendiculaire à la précédente.

On peut arrêter les itérations arbitrairement dès que l'on atteint une valeur du gradient suffisamment petite ('vallée' de la fonction), que le nombre de cycle est jugé assez grand, que les paramètres varient trop faiblement ou que le temps alloué pour le calcul est dépassé.

En fait, il est facile de trouver des fonctions non pathologiques où la direction vers le minimum est perpendiculaire au gradient.

**Pour les formes quadratiques définies positives :**

Une forme quadratique générale est :

$$F(x) = a + gx + \frac{1}{2} Hx^2 \quad (\text{II.40})$$

où

$$g = \frac{\partial F}{\partial x} \text{ à } x = 0 \text{ (} g \text{ est le gradient) ;} \quad (\text{II.41})$$

et

$$H = \frac{\partial^2 F}{\partial x^2} \text{ à } x = 0 \text{ (} H \text{ est la deuxième dérivée ou le Hessien)} \quad (\text{II.42})$$

Cette fonction a un minimum si et seulement si  $H \geq 0$  (si  $H = 0$ , le minimum est à l'infini, car la dérivée seconde étant nulle, la fonction est plate. Si  $H$  est négatif, alors on se trouve sur une parabole de courbure négative, donc sans minimum). Le minimum, s'il existe, est à  $x = -H^{-1}g$ . En effet, au minimum, la dérivée de  $F$  est nulle, donc :

$$F'(x) = g + Hx = 0 \Rightarrow x = -g/h \quad (\text{II.43})$$

Quand on utilise une forme quadratique pour approximer une fonction générale non linéaire, il serait logique de prendre un pas de  $x = -g/h$  pour trouver une approximation du minimum, mais cette approximation n'est valable que si  $H > 0$ , sinon on saute à un maximum ou à l'infini.

Si  $H < 0$ , le remède serait de prendre un pas de  $x = -g$ , c'est-à-dire mettre arbitrairement  $H$  égale à 1, de manière à ce que le pas soit au moins dans la bonne direction (celui de l'inverse du gradient ou croit la fonction), même si ce pas a maintenant une longueur arbitraire. L'examen de la figure ci-dessous montre que c'est la seule chose l'on puisse faire sans information supplémentaire.

## II.10 Description de la fonction *Fmincon* de MATLAB [15]:

*Fmincon* est l'abréviation anglaise de « *FindMinimum of Constrained non Linear Multivariable Function* » ou trouver le minimum d'une fonction multi-variable avec contrainte non linéaire.

Le but de cette fonction est de minimiser une fonction  $f$  sur un ensemble  $S$ .

*Fmincon* est la fonction de MATLAB (optimization toolbox version 2.2) pour les problèmes d'optimisation sous contraintes. Pour les problèmes généraux d'optimisation non linéaire, *Fmincon* utilise la méthode SQP (programmation quadratique séquentielle). L'estimation initiale de l'hessien du lagrangien est la matrice identité. Sa correction est fait par la formule de Broyden, Fletcher, Goldfarb et Shanno (BFGS). Si le gradient de la fonction coût n'est pas fourni, MATLAB l'estime par différence finie.

*Fmincon* recherche le minimum d'une fonction soumise à des contraintes linéaires et/ou non linéaires dans un intervalle donné. Le but de la fonction est la minimisation d'une fonction  $f$  sur un ensemble  $S$ . Si l'on se trouve en un point  $x_k$ , on cherche à se diriger vers un point  $y$  dont l'image par  $f$  est plus petit. Pour cela, on approxime  $f$  par une fonction  $q$  relativement simple, qui reproduit assez bien  $f$  dans un voisinage  $N$  de  $x$ . Ce voisinage est appelé « région de confiance ».

On minimise alors  $q$  sur  $N$ , ce qui nous donne le point  $y$ . Ensuite, on pose (puisque l'on cherche à minimiser  $f$ ) :

$$x_{k+1} = y \text{ si } f(y) < f(x_k) \quad (\text{II.44})$$

Et sinon on rétrécit la région de confiance et on approxime à nouveau  $f$  par une fonction  $q$  sur le nouveau voisinage.



Du point de vue de l'implémentation, le gradient  $g$  est approximé partiellement dans toutes les directions, et la matrice hessienne  $H$  est calculée récursivement par une méthode de type quasi-newtonienne, on l'occurrence l'algorithme BFGS. Ainsi chaque itération de la routine *Fmincon* coûte de l'ordre de  $n$  évaluations de la fonction  $f$  si  $n$  est la dimension du problème.

En effet, l'approximation du gradient se fait par la méthode des différence finie, c'est-à-dire exactement comme dans la méthode de la descente.

Alors on a un problème de la forme suivante :

$$\min_x (F(x))$$

$$\text{Tel que : } \begin{cases} C(x) \leq 0 \\ C_{eq} = 0 \\ Ax \leq B \\ A_{eq}x = B_{eq} \\ LB \leq x < UB \end{cases} \quad (\text{II.45})$$

$x, B, B_{eq}, LB$  et  $UB$  Sont des vecteurs,  $A$  et  $A_{eq}$  sont des matrices,  $C(x)$  et  $C_{eq}(x)$  sont des fonctions à valeurs en vecteurs,  $F(x)$  est une fonction scalaire.  $f(x), C(x)$  et  $C_{eq}(x)$  Peuvent être des fonctions non linéaires.

### **II.11 Conclusion :**

Dans ce chapitre nous avons présenté une méthode pour la génération de trajectoire qui est la méthode de l'interpolation polynomiale (d'ordre cubique, cinq et le mouvement continu entre points via), Et on a présenté quelques notions sur les actionneurs qui sont actuellement utilisés en robotique (électriques, hydrauliques, ou pneumatiques) en fonction de nos besoins; dans le cas générale on utilise les actionneurs électriques quand il s'agit d'une articulation rotoïde et pour soulever des poids légers. Par contre on utilise les actionneurs pneumatiques ou hydrauliques dans le cas d'une articulation prismatique et aussi quand il est question de soulever un poids plus important.

Ensuite, les différentes méthodes de commande optimale ont été rappelées brièvement. Suivies d'un rappel sur la commande optimale avec les diverses méthodes de résolution du problème de la commande optimale : méthodes directes, méthodes indirectes. A la fin du chapitre, le problème d'optimisation et les méthodes de sa résolution y sont expliqués. En particulier, la fonction *Fmincon* du logiciel MATLAB a été introduite.

## **CHAPITRE III**

### **APPLICATION SUR UN ROBOT**

---

### III.1 Introduction :

Notre travail consiste à faire la génération d'une trajectoire optimale étant donnée la pose initiale et la pose finale du bras manipulateur .La trajectoire générée doit minimiser l'effort fourni par les actionneurs du robot et doit aussi éviter la surcharge de ces derniers c'est-à-dire éviter leurs saturations. Elle doit aussi maximiser la charge du robot c'est-à-dire le poids soulevé. La méthode utilisée ici permet aussi de prendre en considération l'évitement d'obstacle et ceci en ajoutant un nouveau terme à la fonction coût. La trajectoire est décrite en utilisant le B-spline qui sera introduit en premier lieu. Les points de contrôle du B-spline constituent l'ensemble de paramètres de cette description et le B-spline constitue la base qui permet sa génération. L'expression analytique du gradient de la fonction de coût par rapport à ces points de contrôle permet à la procédure d'optimisation de converger facilement vers le minimum et la rend plus robuste.

On aura ainsi transformé le problème de recherche d'une commande optimale qui minimise le critère  $J_c$  en un problème d'optimisation.

### III.2 Génération de la trajectoire par B-splines :

Les trajectoires articulaire sont approximées par le B-spline d'ordre  $K$  à travers un espace de nœuds d'une séquence ordonnée de temps, soient :

La séquence de nœuds :

$$0 = t_0 = \dots = t_{K-1} \leq t_K \leq \dots \leq t_m \leq t_{m+1} = \dots = t_{m+K} = t_f \quad (\text{III.1})$$

Avec :  $m \geq K - 1$

Les points de contrôle  $P_j$  :

$$P_j = \{p_0, \dots, p_m\} \quad (\text{III.2})$$

Les trajectoires articulaires  $\theta(t, p) \in \mathbb{R}^n$  sont alors calculées comme suit :

$$\theta(t, p) = \sum_{j=0}^m p^j B_{j,K}(t) \quad (\text{III.3})$$

Où :  $B_{j,K}(t)$  est la fonction de base du B-spline

$$B_{j,K}(t) = \begin{cases} g_{jl}(t), & \text{si } t_{j+l} \leq t < t_{j+l+1} \\ 0, & \text{si } t < t_j \text{ ou } t \geq t_{j+K} \end{cases} \quad (\text{III.4})$$

Avec :  $l=0 \sim K - 1$

$g_{jl}(t)$ : est un polynôme d'ordre  $K - 1$ .

La fonction  $B_{j,K}(t)$  peut être obtenue en utilisant la formule récursive suivante :

$$B_{j,K}(t) = \frac{t-t_j}{t_{j+K-1}-t_j} B_{j,K-1}(t) + \frac{t_{j+K}-t}{t_{j+K}-t_{j+1}} B_{j+1,K-1}(t) \quad (\text{III.5})$$

Avec la convention, si un dénominateur est nul du fait d'un nœud multiple, on considère que le quotient correspondant est nul

$$\left[ \frac{t-t_j}{0} = \frac{t_{j+K}-t}{0} = 0 \text{ (Quand } t_{j+K-1} - t_j = 0 \text{ et/ou } t_{j+K} - t_{j+1} = 0) \right]$$

En commençant par :

$$B_{j,1}(t) = \begin{cases} 1, & \text{si } t_j \leq t < t_{j+1} \\ 0, & \text{sinon} \end{cases} \quad (\text{III.6})$$

➤ Détermination des vitesses et des accélérations :

On a la dérivée de  $B_{j,K}(t)$  donnée par :

$$\frac{\partial B_{j,K}(t)}{\partial t} = (K-1) \left[ \frac{B_{j,K-1}(t)}{t_{j+K-1}-t_j} - \frac{B_{j+1,K-1}(t)}{t_{j+K}-t_{j+1}} \right] \quad (\text{III.7})$$

La dérivée par rapport au temps de la trajectoire articulaire  $\theta(t, p)$  est donnée comme suit :

$$\frac{d}{dt} \theta(t, P) = \sum_{j=0}^m p^j \frac{\partial}{\partial t} B_{j,K}(t) = (K-1) \sum_{j=0}^{m+1} \frac{p^j - p^{j-1}}{t_{j+K-1} - t_j} B_{j,K-1}(t) \quad (\text{III.8})$$

avec  $p^{-1}, p^{m+1}$  sont définis comme zéro.

La dérivée d'ordre supérieur de  $\theta(t, p)$  est déterminée de manière similaire,

$$\frac{d^2}{dt^2} \theta(t) = \sum_{j=0}^m p^j \frac{\partial^2}{\partial t^2} B_{j,K}(t) = (K-1)(K-2) \sum_{j=0}^{m+2} \left[ \frac{p^j - p^{j-1}}{t_{j+K-1} - t_j} - \frac{p^{j-1} - p^{j-2}}{t_{j+K-2} - t_{j-1}} \right] * \quad (\text{III.9})$$

➤ Principales propriétés des fonctions B-spline :

1. la fonction  $B_{j,K}$  est sur chaque intervalle  $[t_j, t_{j+1}[$  un polynôme de degré  $\leq K$  ;

2. la fonction  $B_{j,K}$  s'annule en dehors de l'intervalle  $[t_j, t_{j+1}[$  ;

3. pour  $t \in [t_j, t_{j+1}[$ ,  $0 < B_{j,K} \leq 1$ ;

4. sur l'intervalle  $[t_K, t_{m-K}[$ ,  $\sum_{j=0}^{m-K-1} B_{j,K} = 1$ ;
5. la fonction  $B_{j,K}$  est de classe  $c^\infty$  à droite de chaque point ;
6. au voisinage d'un nœud de multiplicité  $r$ , la fonction  $B_{j,K}$  est seulement de classe  $c^{K-r}$  ;

### III.3 Planification de trajectoire optimale avec chemin contraint et effort minimal :

Considérons un manipulateur sous forme de chaîne ouverte de degré de liberté  $n$  le problème d'optimisation pour ce système est comme suit :

$$\left\{ \begin{array}{l} \text{minimiser } j_c = j_w + j_e \\ \text{tel que:} \\ M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) = \Gamma \\ \underline{\Gamma} \leq \Gamma(t) \leq \bar{\Gamma} \\ \underline{\theta} \leq \theta(t) \leq \bar{\theta} \text{ et } p_w > 0 \\ \underline{\dot{\theta}} \leq \dot{\theta}(t) \leq \bar{\dot{\theta}} \\ \theta(0) = \theta_0, \dot{\theta}(0) = 0 \\ \theta(t_f) = \theta_f, \dot{\theta}(t_f) = 0 \end{array} \right. \quad \begin{array}{l} \text{(III. 10)} \\ \text{(III. 11)} \\ \text{(III. 12)} \\ \text{(III. 13)} \\ \text{(III. 14)} \\ \text{(III. 15)} \\ \text{(III. 16)} \end{array}$$

avec :

$$j_w = -w p_w \quad \text{(III.17)}$$

$$j_e = \frac{1}{2} \int_0^{p t_f} \Gamma^T w_e \Gamma dt \quad \text{(III.18)}$$

L'équation(III.11) représente la dynamique inverse du manipulateur où  $\theta$  est le vecteur de dimension  $n$  de coordonnées articulaires.  $M(\theta)$  est la matrice d'inertie de dimension  $(n \times n)$  et  $h(\theta, \dot{\theta})$  contient les effets de coriolis, ceux de gravité et enfin ceux de frottement.

Les contraintes sur les couples articulaires, les coordonnées  $\theta$  et les vitesses articulaires  $\dot{\theta}$  sont données respectivement par les équations (III.12),(III.13) et (III.14), avec les bornes supérieures et inférieures sur  $\Gamma$ ,  $\theta$  et  $\dot{\theta}$  supposées données au départ. Le point soulevé par le manipulateur est  $p_w$ .

Le coût  $j_w$  est introduit pour maximiser la charge soulevée par le manipulateur, et le coût  $j_e$  est introduit pour minimiser l'effort appliqué par les actionneurs.

#### Le cas d'une surcharge des actionneurs :

Dans le cas où le couple est en dehors de  $[\underline{\tau}, \bar{\tau}]$ , on introduit le coût  $j_p$  pour minimiser la surcharge des actionneurs, autrement dit pour limiter le couple  $\Gamma$  dans l'intervalle  $[\underline{\tau}, \bar{\tau}]$ .

$$j_p = \frac{1}{2} \int_0^{p_t t_f} \varphi_+^T W_p \varphi_+ dt \quad (\text{III.19})$$

Avec

$$\varphi_+ = \begin{cases} \varphi & \text{si } \varphi > 0 \\ 0 & \text{si non} \end{cases} \quad (\text{III.20})$$

$$\varphi = \begin{bmatrix} \tau - \bar{\tau} \\ \tau - \bar{\tau} \end{bmatrix} \quad (\text{III.21})$$

Donc le coût  $j_c$  devient :

$$j_c = j_w + j_e + j_p \quad (\text{III.22})$$

### Utilisation du facteur d'échelle :

Le paramètre  $p_t$  est un coefficient d'échelle pour le temps. Si un coefficient égale à 1 et que le temps  $t_f$  est choisi, on aura un problème à temps fixe. par contre si  $t_f$  est choisi et que le coefficient  $p_t$  est considéré comme paramètre additionnel inconnu, on aura un problème à temps finale libre.

Si l'on utilise un facteur d'échelle  $p_t$  pour le temps, qui normalise l'espace de temps (l'espace des nœuds dans notre cas), l'équation (III.3) devient :

$$\theta(t, p_t, P) = \theta(\eta, P) = \sum_{j=0}^m p^j B_{j,K}(\eta) \quad (\text{III.23})$$

Avec  $\eta = p_t t$ .

$$\frac{\partial}{\partial t} B_{j,K}(\eta) = \frac{1}{p_t} \frac{\partial}{\partial t} B_{j,K}(t) \quad (\text{III.24})$$

On remplace  $\partial \eta$  par  $p_t \partial t$ , et l'on obtient :

$$\frac{\partial}{\partial p_t} B_{j,K}(\eta) = -\frac{\eta}{p_t} \frac{\partial}{\partial t} B_{j,K}(\eta) \quad (\text{III.25})$$

Ceci implique que :

$$\frac{\partial}{\partial p_t} \theta(\eta, P) = -\frac{\eta}{p_t} \frac{\partial}{\partial \eta} \theta(\eta, P) \quad (\text{III.26})$$

Avec ce paramétrage de la trajectoire articulaire, le problème d'optimisation sera transformé en la forme suivante :

$$\left\{ \begin{array}{l} \text{minimiser } j_c = j_w + j_e + j_p \\ \text{tel que :} \\ M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) = \Gamma \\ \underline{\Gamma} \leq \Gamma(t) \leq \overline{\Gamma} \\ \underline{\theta} \leq p^j \leq \overline{\theta} \text{ et } (p_t, p_w) > 0 \\ \dot{\theta} \leq \frac{(K-1)(p^j - p^{j-1})}{t_{j+K-1} - t_j} \leq \overline{\dot{\theta}} \\ p^0 = p^1 = \theta_0, p^{m-1} = p^m = \theta_f \\ \theta(p_t t_f) = \theta_f, \dot{\theta}(p_t t_f) = 0 \end{array} \right. \quad \begin{array}{l} \text{(III. 27)} \\ \\ \text{(III. 28)} \\ \text{(III. 29)} \\ \text{(III. 30)} \\ \text{(III. 31)} \\ \text{(III. 32)} \\ \text{(III. 33)} \end{array}$$

La résolution numérique de ce problème est possible, toutefois dans le cas où le gradient approximatif obtenu par la méthode de différence finie conduit à une matrice hessienne mal conditionnée, l'algorithme peut échouer. Ainsi, une solution analytique pour ce gradient constitue un grand avantage.

Dans notre travail, pour résoudre ce problème, nous avons utilisé le paramétrage de la trajectoire articulaire avec B-spline du cinquième ordre  $\theta = \theta(p_t t, P)$ .

### III.3.1 Modèle dynamique d'un bras manipulateur rigide à deux degrés de liberté :

Pour le calcul du modèle dynamique d'une chaîne cinématique (**Fig III.1**), on utilise la formulation de lagrange décrit précédemment au chapitre I (I.16), et l'on définit les paramètres suivants :

$m_i$  : masse du corps  $i$  de la chaîne cinématique.

$w_1 = [0 \ 0 \ \dot{\theta}_1]^T$  : la vitesse angulaire du segment 1.

$w_2 = [0 \ 0 \ \dot{\theta}_1 + \dot{\theta}_2]^T$  : la vitesse angulaire du segment 2.

$\|w_1\| = \dot{\theta}_1, \|w_2\| = (\dot{\theta}_1 + \dot{\theta}_2)$ : les normes de  $w_1$  et  $w_2$ .

$\dot{r}_{c1} = w_1 \times r_{c1}$  : vitesse du centre de masse du segment 1.

$$\dot{r}_{c1}^2 = \dot{\theta}_1^2 \frac{l_1^2}{4}$$

$$r_{c2} = \begin{bmatrix} l_1 \cos \theta_1 + \frac{l_2}{2} \cos (\theta_1 + \theta_2) \\ l_1 \sin \theta_1 + \frac{l_2}{2} \sin (\theta_1 + \theta_2) \\ 0 \end{bmatrix} : \text{position du centre de masse du segment 2 ;}$$

$$\dot{r}_{c2}^2 = l_1^2 \dot{\theta}_1^2 + \frac{l_2^2}{4} (\dot{\theta}_1 + \dot{\theta}_2)^2 + l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2 : \text{vitesse du centre d'inertie du segment 2 ;}$$

$$(i)_{I_{ci}} = \begin{bmatrix} \frac{m_i}{12}(b_i^2 + c_i^2) & 0 & 0 \\ 0 & \frac{m_i}{12}(c_i^2 + l_i^2) & 0 \\ 0 & 0 & \frac{m_i}{12}(l_i^2 + b_i^2) \end{bmatrix} : \text{matrice d'inertie.}$$

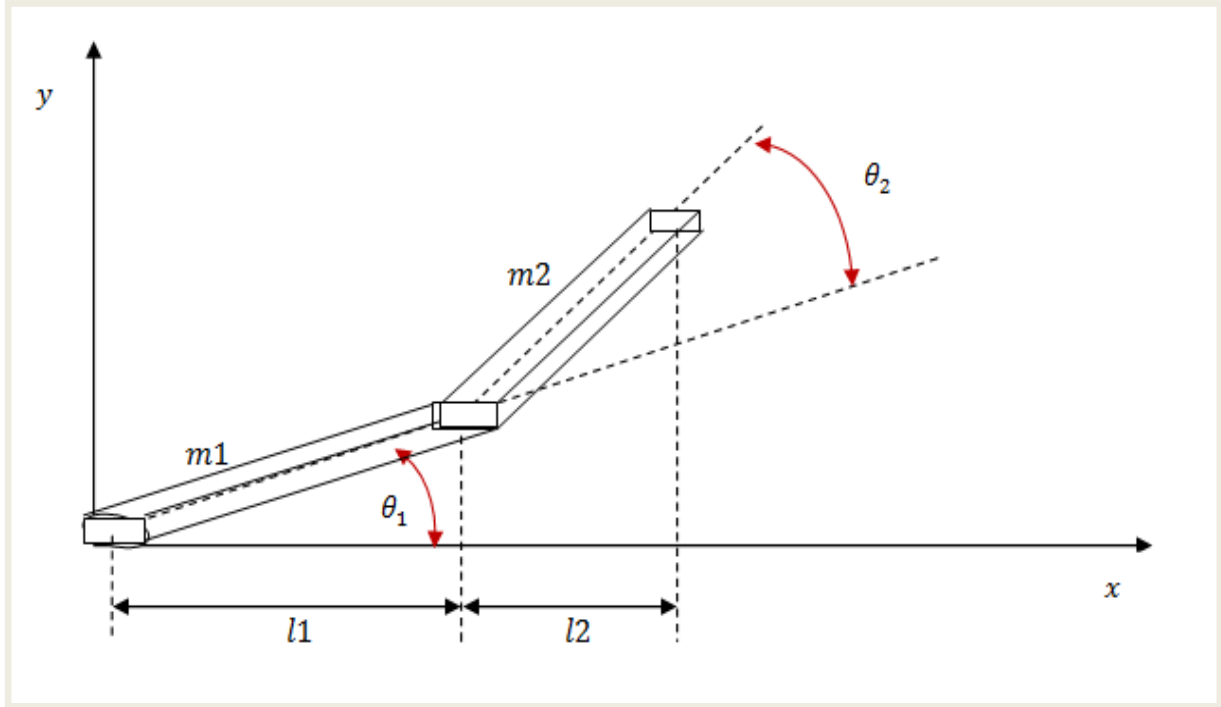


Fig.III.1 représentation d'un bras manipulateur à 2 ddl

**III.3.1.1 Modèle dynamique d'un robot à 2 ddl sans charge sur l'effecteur :**

➤ Calcul de l'énergie cinétique :

L'énergie cinétique du segment 1 :

$$E_{c1} = \frac{1}{2} \dot{r}_{c1}^2 m_1 + \frac{1}{2} {}^{(1)}w_1^T \cdot (I_{c1} {}^{(1)}w_1)$$

$$E_{c1} = \frac{l_1^2}{8} m_1 \dot{\theta}_1^2 + \frac{(l_1^2 + b_1^2)}{24} m_1 \dot{\theta}_1^2 \tag{III.34}$$

L'énergie cinétique du segment 2 :

$$E_{c2} = \frac{1}{2} \dot{r}_{c2}^2 m_2 + \frac{1}{2} {}^{(2)}w_2^T ({}^{(2)}I_{c1} {}^{(1)}w_1)$$

$$E_{c2} = \frac{m_2}{2} l_1^2 \dot{\theta}_1^2 + \frac{m_2}{8} l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{m_2}{2} l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos \theta_2 + \frac{m_2}{24} (l_2^2 + b_2^2) (\dot{\theta}_1 + \dot{\theta}_2)^2 \tag{III.35}$$

L'énergie cinétique totale :



$$T = E_{c1} + E_{c2} = \frac{l_1^2}{8} m_1 \dot{\theta}_1^2 + \frac{(l_1^2 + b_1^2)}{24} m_1 \dot{\theta}_1^2 + \frac{m_2}{2} l_1^2 \dot{\theta}_1^2 + \frac{m_2}{8} l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{m_2}{2} l_1 l_2 (\dot{\theta}_1^2 + \theta_1 \theta_2 \cos \theta_2 + m_2 24 l_2^2 + b_2^2 (\theta_1 + \theta_2)^2)$$

➤ Calcul de l'énergie potentielle :

L'énergie potentielle du segment 1 :

$$E_{P1} = \frac{m_1}{2} l_1 g \sin \theta_1 \tag{III.36}$$

L'énergie potentielle du segment 2 :

$$E_{P2} = m_2 g l_1 \sin \theta_1 + m_2 g \frac{l_2}{2} \sin(\theta_1 + \theta_2) \tag{III.37}$$

L'énergie potentielle totale :

$$V = E_{P1} + E_{P2} = \frac{m_1}{2} l_1 g \sin \theta_1 + m_2 g l_1 \sin \theta_1 + m_2 g \frac{l_2}{2} \sin(\theta_1 + \theta_2)$$

➤ Calcul du lagrangien :

$$L = \frac{l_1^2}{8} m_1 \dot{\theta}_1^2 + \frac{(l_1^2 + b_1^2)}{24} m_1 \dot{\theta}_1^2 + \frac{m_2}{2} l_1^2 \dot{\theta}_1^2 + \frac{m_2}{8} l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{m_2}{2} l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos \theta_2 + m_2 24 l_2^2 + b_2^2 (\theta_1 + \theta_2)^2 - m_2 l_1 g \sin \theta_1 + m_2 g l_1 \sin \theta_1 + m_2 g l_2 \sin(\theta_1 + \theta_2)$$

➤ Les équations du mouvement

❖ Couple 1 :  $\tau_1 = \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1}$

La dérivée du lagrangien par rapport à  $\dot{\theta}_1$  :

$$\frac{\partial L}{\partial \dot{\theta}_1} = \frac{m_1}{4} l_1^2 \dot{\theta}_1 + \frac{m_1}{12} (l_1^2 + b_1^2) \dot{\theta}_1 + m_2 l_1^2 \dot{\theta}_1 + \frac{m_2}{4} l_2^2 \dot{\theta}_1 + \frac{m_2}{4} l_2^2 \dot{\theta}_2 + m_2 l_1 l_2 \dot{\theta}_1 \cos \theta_2 + m_2 24 l_2^2 + b_2^2 (\theta_1 + \theta_2)^2 + m_2 21 l_2^2 + b_2^2 \theta_1 + m_2 21 l_2 (l_2^2 + b_2^2) \theta_2$$

La dérivée du  $\frac{\partial L}{\partial \dot{\theta}_1}$  par rapport au temps est donnée comme suit :

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) &= \frac{m_1}{4} l_1^2 \ddot{\theta}_1 + \frac{m_1}{12} (l_1^2 + b_1^2) \ddot{\theta}_1 + m_2 l_1^2 \ddot{\theta}_1 + \frac{m_2}{4} l_2^2 \ddot{\theta}_1 + \frac{m_2}{4} l_2^2 \ddot{\theta}_2 \\ &+ m_2 l_1 l_2 \ddot{\theta}_1 \cos \theta_2 - m_2 l_1 l_2 \dot{\theta}_2 \dot{\theta}_1 \sin \theta_2 + \frac{m_2}{2} l_1 l_2 \ddot{\theta}_2 \cos \theta_2 - \frac{m_2}{2} l_1 l_2 \dot{\theta}_2^2 \sin \theta_2 \\ &+ \frac{m_2}{12} (l_2^2 + b_2^2) \ddot{\theta}_1 + \frac{m_2}{12} (l_2^2 + b_2^2) \ddot{\theta}_2 \end{aligned}$$

La dérivée du lagrangien par rapport à  $\theta_1$  est :

$$\frac{\partial L}{\partial \theta_1} = -\frac{m_1}{2} l_1 g \cos \theta_1 - m_2 g l_1 \cos \theta_1 - m_2 g \frac{l_2}{2} \cos(\theta_1 + \theta_2)$$

Le couple du segment 1 est donné par la relation suivante:

$$\begin{aligned} \tau_1 = & \left[ \frac{m_1}{4} l_1^2 + \frac{m_1}{12} (l_1^2 + b_1^2) + m_2 l_1^2 + \frac{m_2}{4} l_2^2 + m_2 l_1 l_2 \cos \theta_2 + \frac{m_2}{12} (l_2^2 + b_2^2) \right] \ddot{\theta}_1 \\ & + \left[ \frac{m_2}{4} l_2^2 + \frac{m_2}{2} l_1 l_2 \cos \theta_2 + \frac{m_2}{12} (l_2^2 + b_2^2) \right] \ddot{\theta}_2 \\ & + \left[ -m_2 l_1 l_2 \dot{\theta}_2 \dot{\theta}_1 \sin \theta_2 - \frac{m_2}{2} l_1 l_2 \dot{\theta}_2^2 \sin \theta_2 + \frac{m_1}{2} l_1 g \cos \theta_1 + m_2 g l_1 \cos \theta_1 \right. \\ & \left. + m_2 g \frac{l_2}{2} \cos(\theta_1 + \theta_2) \right] \end{aligned}$$

❖ Couple 2 :  $\tau_2 = \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2}$

$$\frac{\partial L}{\partial \dot{\theta}_2} = \frac{m_2}{4} l_2^2 \dot{\theta}_2 + \frac{m_2}{4} l_2^2 \dot{\theta}_1 + \frac{m_2}{2} l_1 l_2 \dot{\theta}_1 \cos \theta_2 + \frac{m_2}{12} (l_2^2 + b_2^2) \dot{\theta}_2 + \frac{m_2}{12} (l_2^2 + b_2^2) \dot{\theta}_1$$

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) = & \frac{m_2}{4} l_2^2 \ddot{\theta}_2 + \frac{m_2}{4} l_2^2 \ddot{\theta}_1 + \frac{m_2}{2} l_1 l_2 \ddot{\theta}_1 \cos \theta_2 - \frac{m_2}{2} l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2 + \frac{m_2}{12} (l_2^2 + b_2^2) \ddot{\theta}_2 + \\ & \frac{m_2}{12} (l_2^2 + b_2^2) \ddot{\theta}_1 \end{aligned}$$

$$\frac{\partial L}{\partial \theta_2} = -\frac{m_2}{2} l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \sin \theta_2 - m_2 \frac{l_2}{2} g \cos(\theta_1 + \theta_2)$$

$$\begin{aligned} \tau_2 = & \left[ \frac{m_2}{4} l_2^2 + \frac{m_2}{2} l_1 l_2 \cos \theta_2 + \frac{m_2}{12} (l_2^2 + b_2^2) \right] \ddot{\theta}_1 + \left[ \frac{m_2}{4} l_2^2 + \frac{m_2}{12} (l_2^2 + b_2^2) \right] \ddot{\theta}_2 \\ & + \left[ -\frac{m_2}{2} l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2 + \frac{m_2}{2} l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \sin \theta_2 + m_2 \frac{l_2}{2} g \cos(\theta_1 + \theta_2) \right] \end{aligned}$$

Donc le couple est :

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

On peut écrire les équations du mouvement sous forme matricielle comme suit :

$$\tau = M(\theta) \ddot{\theta} + h(\theta, \dot{\theta}) \quad (\text{III.38})$$

avec :

$$M(\theta) =$$

$$\begin{bmatrix} \frac{m_1}{4} l_1^2 + \frac{m_1}{12} (l_1^2 + b_1^2) + m_2 l_1^2 + \frac{m_2}{4} l_2^2 + m_2 l_1 l_2 \cos \theta_2 + \frac{m_2}{12} (l_2^2 + b_2^2) & \frac{m_2}{4} l_2^2 + \frac{m_2}{2} l_1 l_2 \cos \theta_2 + \frac{m_2}{12} (l_2^2 + b_2^2) \\ \frac{m_2}{4} l_2^2 + \frac{m_2}{2} l_1 l_2 \cos \theta_2 + \frac{m_2}{12} (l_2^2 + b_2^2) & \frac{m_2}{4} l_2^2 + \frac{m_2}{12} (l_2^2 + b_2^2) \end{bmatrix}$$

$$h(\theta, \dot{\theta}) =$$

$$\begin{bmatrix} -m_2 l_1 l_2 \dot{\theta}_2 \dot{\theta}_1 \sin \theta_2 - \frac{m_2}{2} l_1 l_2 \dot{\theta}_2^2 \sin \theta_2 + \frac{m_1}{2} l_1 g \cos \theta_1 + m_2 g l_1 \cos \theta_1 + m_2 g \frac{l_2}{2} \cos(\theta_1 + \theta_2) \\ -\frac{m_2}{2} l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2 + \frac{m_2}{2} l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \sin \theta_2 + m_2 \frac{l_2}{2} g \cos(\theta_1 + \theta_2) \end{bmatrix}$$

Où :

$M(\theta)$  est la matrice d'inertie de dimension (2\*2)

$h(\theta, \dot{\theta})$  contient divers effets, ceux de coriolis, centrifuges, de gravité et enfin ceux de frottement.

### III.3.1.2 Modèle dynamique d'un robot à 2 ddl avec un poids ( $p_w$ ) sur l'organe terminal :

➤ Calcul de l'énergie cinétique :

$$T = E_{c1} + E_{c2} + E_{cpw} \quad (\text{III.39})$$

$E_{cpw}$  : est l'énergie cinétique de poids,

$$E_{cpw} = \frac{p_w}{2g} l_1^2 \dot{\theta}_1 + \frac{p_w}{2g} l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{p_w}{g} l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2 \quad (\text{III.40})$$

➤ L'énergie potentielle

$$V_{pw} = p_w [l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)] \quad (\text{III.41})$$

$$\dot{V} = V + V_{pw}$$

➤ Lagrangien :

$$\dot{L} = \dot{T} - \dot{V} \quad (\text{III.42})$$

➤ Les équations du mouvement :

En suivant le même raisonnement que dans le premier cas, on obtient :

$$\tau = \dot{M}(\theta)\ddot{\theta} + \dot{h}(\theta, \dot{\theta}) \quad (\text{III.43})$$

Avec :

$\dot{M} =$

$$\begin{bmatrix} \left( \frac{m_1}{4} l_1^2 + \frac{(l_1^2 + b_1^2)}{12} m_1 + m_2 l_1^2 + \frac{m_2 l_2^2}{4} + m_2 l_1 l_2 \cos \theta_2 + \right) & \left( \frac{m_2}{2} l_2^2 + \frac{m_2}{2} l_1 l_2 \cos \theta_2 + \frac{(l_2^2 + b_2^2)}{12} m_2 + \right) \\ \left( \frac{m_2}{12} (l_2^2 + b_2^2) + \frac{p_w}{g} l_1^2 + \frac{p_w}{g} l_2^2 + \frac{2p_w}{g} l_1 l_2 \cos \theta_2 \right) & \left( \frac{p_w}{g} l_2^2 + \frac{p_w}{g} l_1 l_2 \cos \theta_2 \right) \\ \left( \frac{m_2}{4} l_2^2 + \frac{m_2}{2} l_1 l_2 \cos \theta_2 + \frac{(l_2^2 + b_2^2)}{12} m_2 \right) & \left( \frac{m_2}{4} l_2^2 + \frac{(l_2^2 + b_2^2)}{12} m_2 + \frac{p_w}{g} l_2^2 \right) \\ \left( \frac{p_w}{g} l_2^2 + \frac{p_w}{g} l_1 l_2 \cos \theta_2 \right) & \end{bmatrix}$$

$h =$

$$\begin{bmatrix} \left( -m_2 l_1 l_2 \dot{\theta}_2 \dot{\theta}_1 \sin \theta_2 - \frac{m_2}{2} l_1 l_2 \dot{\theta}_2^2 \sin \theta_2 - \frac{m_1}{2} l_1 g \cos \theta_1 + m_2 g l_1 \cos \theta_1 + m_2 g \frac{l_2}{2} \cos(\theta_1 + \theta_2) - \right) \\ \left( \frac{p_w}{g} l_1 l_2 \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \sin \theta_2 + p_w l_1 \cos \theta_1 + p_w l_2 \cos(\theta_1 + \theta_2) \right) \\ \left( \frac{m_2}{2} l_1 l_2 \dot{\theta}_1^2 \sin \theta_2 + m_2 \frac{l_2}{2} g \cos(\theta_1 + \theta_2) + \right) \\ \left( \frac{p_w}{g} l_1 l_2 \dot{\theta}_1^2 \sin \theta_2 + p_w l_2 \cos(\theta_1 + \theta_2) \right) \end{bmatrix}$$

### III.3.2 La trajectoire initiale :

Soit un robot à deux degrés de liberté, qui passe de la position initiale  $\theta(t_0)$  à la position finale  $\theta(t_f)$ , avec un temps d'exécution  $T = 20s$ .

On prend :  $t_0 = 0, t_f = 20s$

Pour notre choix initial, nous avons supposé que la trajectoire forme un arc de cercle.

$$(x - x_c)^2 + (y - y_c)^2 = R^2 \tag{III.44}$$

Avec :

$x_c$  et  $y_c$  : sont des coordonnées du centre du cercle,  $(x_c, y_c) = (0,0)$

$R$  : c'est le rayent de centre  $R = 0.75$

$x$  : variée de 0 jusqu'à 0.75 avec un pas de 0.075,

La résolution de l'équation (V.44) permet d'obtenir les valeurs de  $y$ .

La figure (**Fig.V.2**) représente la trajectoire initiale du robot dans l'espace opérationnel.

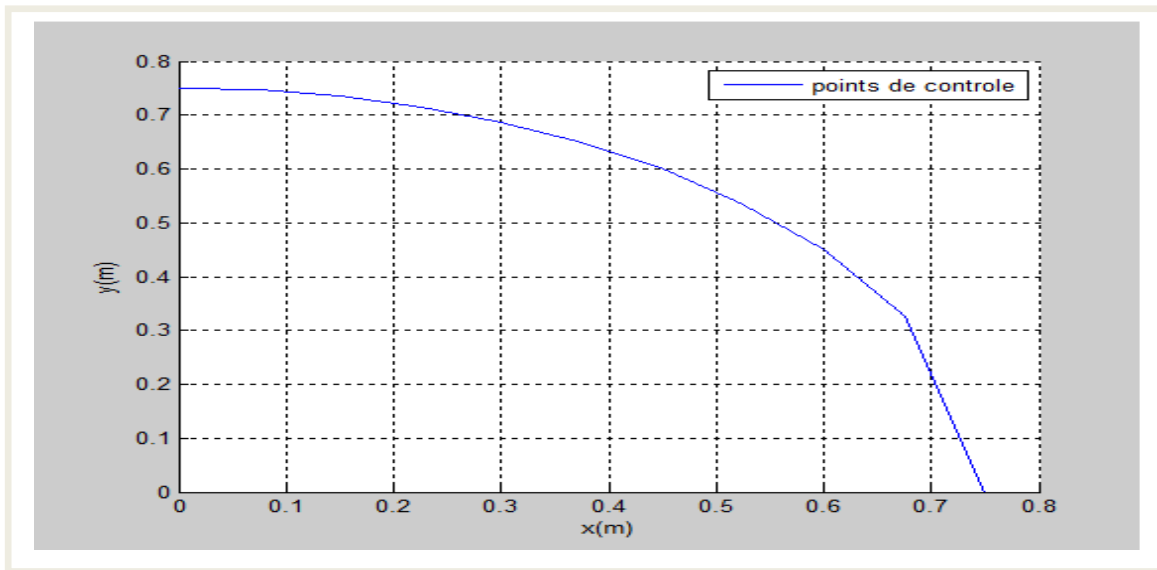


Fig.III.2 trajectoire initiale du robot

**III.3.3 Détermination des points de contrôle :**

Le modèle géométrique direct du robot à 2ddl est donné par la relation suivante :

$$\begin{cases} x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{cases} \quad (III.45)$$

Pour déterminer le modèle géométrique inverse, on suit une démarche analytique, et l'on obtient les relations suivantes :

$$\begin{cases} \theta_1 = \text{Arctg} \left( \frac{y(l_1 + l_2 \cos(\theta_2)) - x l_2 \sin(\theta_2)}{x(l_1 + l_2 \cos(\theta_2)) + y l_2 \sin(\theta_2)} \right) \\ \theta_2 = \pm \text{Arcos} \left( \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2} \right) \end{cases} \quad (III.46)$$

Résolvant l'équation (III.46) , on obtient les résultats rapportés dans le tableau ci-dessous.

Nombre de points	x	y	$\theta_1$	$\theta_2$
1	0.0000	0.7500	90.00	0.00
2	0.0750	0.7462	84.30	0.00
3	0.1500	0.7348	78.50	0.00
4	0.2250	0.7155	72.57	0.00
5	0.3000	0.6874	66.45	0.00

6	0.3750	0.6495	60.03	0.00
7	0.4500	0.6000	53.15	0.00
8	0.5250	0.5356	45.59	0.00
9	0.6000	0.4500	36.88	0.00
10	0.6750	0.3269	25.85	0.00
11	0.7500	0.0000	0.00	0.00

**Tableau III.1** : détermination des points de contrôle.

### III.3.4 Optimisation de la trajectoire :

Pour l'optimisation de la trajectoire nous avons procédé comme suit :

#### III.3.4.1 Approximation de la trajectoire par B-splines :

##### ➤ Calcul de $B_{j,5}$

En utilisant la formule récursive (III.5) donnée au chapitre précédent et on commençant par la formule (III.6) on prend :

$$t_0 = 0; t_f = 20;$$

D'après l'équation(III.1) , on obtient :

$$t = [0 \ 0 \ 0 \ 0 \ 0 \ :2 \ :20 \ 20 \ 20 \ 20 \ 20],$$

le nombre de point est 14 (m=14),

et on a les point de contrôle  $p^j$  qui sont déterminés dans le tableau (**Tableau III.1**) ( $p_i^j = \theta_i^j$ ).

On remarque que le nombre de points (m=14) est supérieur au nombre de points du contrôle ( $m \geq p_i^j$ ) ; donc il faut ajouter trois points de contrôle, une au début ( $p_i^0 = p_i^1$ ) et les deux autres à la fin ( $p_i^{11} = p_i^{12} = p_i^{13}$ ).

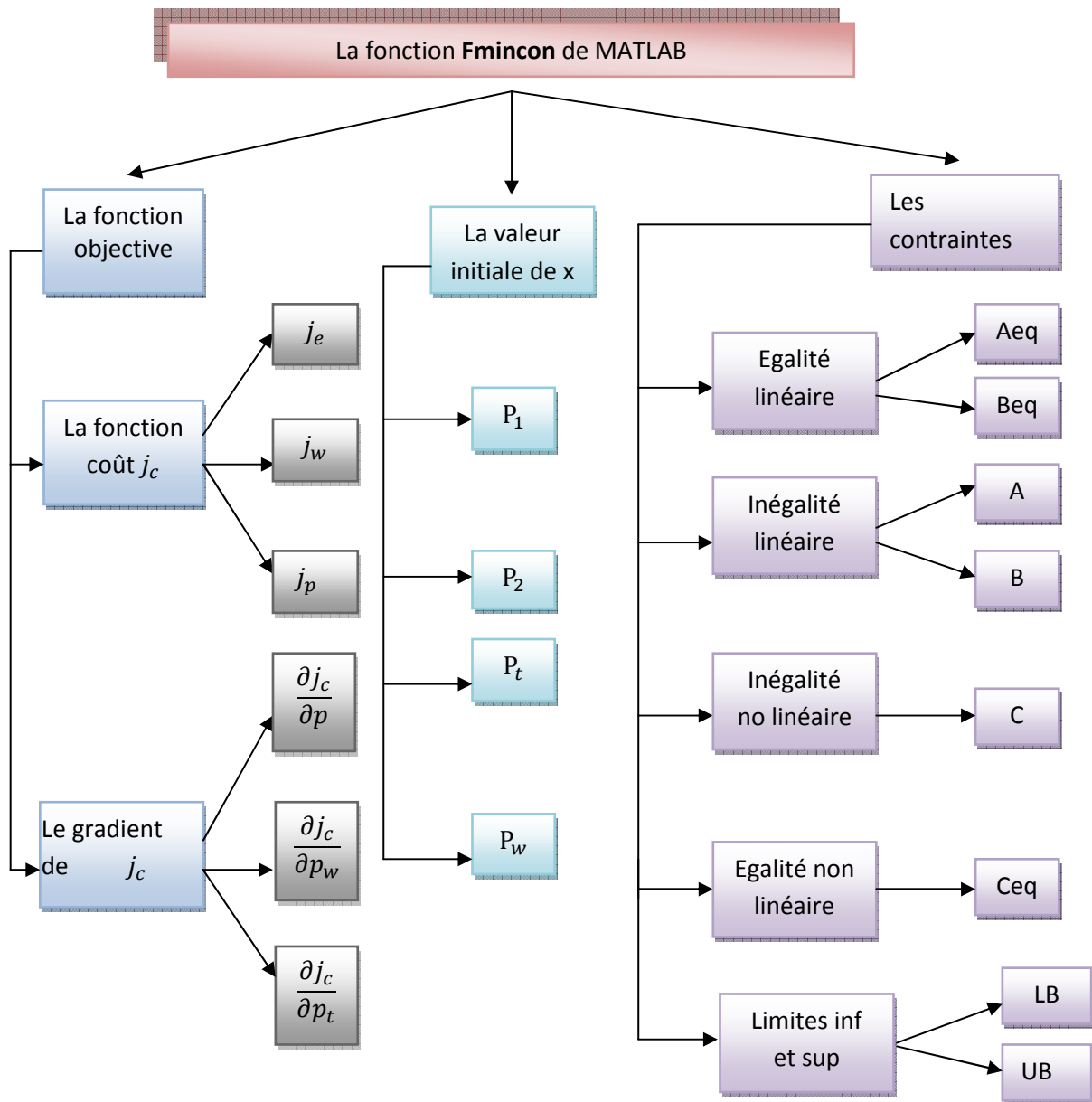
#### III.3.4.2 Optimisation :

Pour l'optimisation on utilise la fonction **Fmincon** de MATLAB qui est présentée au chapitre II.

Avec :

La fonction objective (II.27) , les contraintes linéaires d'égalités (III.32) et inégalité (III.33) et les limites sur la position (III.30).

La figure (Fig III.3) représente un diagramme qui explique comment procéder pour appliquer la fonction **Fmincon** de MATLAB.



**Fig.III.3** explication de la fonction Fmincon.

➤ Pour **Aeq** et **Beq** on les définit à partir des égalités linéaires  $p^0 = p^1 = \theta_0, p^{m-1} = p^m = \theta_f$ .

➤ Pour **A** et **B** on peut les déduire à partir de l'équation suivante :  $\underline{\theta} \leq \frac{(K-1)(p^j - p^{j-1})}{t_{j+K-1} - t_j} \leq \bar{\theta}$ ;

➤ Les limites sur la position **LB** et **UB** sont déterminée comme

suit :  $\underline{\theta} \leq p^j \leq \bar{\theta}$  et  $(p_t, p_w) > 0$

➤ **C** on le définit à partir de  $\underline{\tau} \leq \tau(p_t t) \leq \bar{\tau}$  ;

### Remarque :

Dans notre cas on va pas utiliser les contraintes « égalité non linéaire ( **Ceq** ) »

On introduire le gradient par l'instruction « gradobj », dans la fonction objective.

### Calcul des gradients :

➤ Le gradient de couple

$$\Gamma = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta})$$

$$\nabla \Gamma = \left[ \frac{\partial \Gamma}{\partial P} \quad \frac{\partial \Gamma}{\partial p_w} \right]^T \quad (\text{III.47})$$

❖ Calcul de la dérivée partielle de couple par rapport aux points de contrôle  $\left(\frac{\partial \Gamma}{\partial P}\right)$  :

$$\frac{\partial \Gamma_i}{\partial P} = \frac{\partial \Gamma_i}{\partial \theta_i} \frac{\partial \theta_i}{\partial P} + \frac{\partial \Gamma_i}{\partial \dot{\theta}_i} \frac{\partial \dot{\theta}_i}{\partial P} + \frac{\partial \Gamma_i}{\partial \ddot{\theta}_i} \frac{\partial \ddot{\theta}_i}{\partial P} \quad (\text{III.48})$$

La dérivée du couple 1 par rapport à  $p^j$

$$\bullet \quad \frac{\partial \Gamma_1}{\partial P} = \frac{\partial \Gamma_1}{\partial \theta_1} \frac{\partial \theta_1}{\partial P} + \frac{\partial \Gamma_1}{\partial \theta_2} \frac{\partial \theta_2}{\partial P} + \frac{\partial \Gamma_1}{\partial \dot{\theta}_1} \frac{\partial \dot{\theta}_1}{\partial P} + \frac{\partial \Gamma_1}{\partial \dot{\theta}_2} \frac{\partial \dot{\theta}_2}{\partial P} + \frac{\partial \Gamma_1}{\partial \ddot{\theta}_1} \frac{\partial \ddot{\theta}_1}{\partial P} + \frac{\partial \Gamma_1}{\partial \ddot{\theta}_2} \frac{\partial \ddot{\theta}_2}{\partial P} \quad (\text{III.49})$$

$$\begin{aligned} \frac{\partial \Gamma_1}{\partial \theta_1} = & -\frac{l_1}{2} m_1 g \sin(\theta_1) - m_2 g l_1 \sin(\theta_1) - \frac{m_2 g l_2}{2} \sin(\theta_1 + \theta_2) - p_w l_1 \sin(\theta_1) \\ & + p_w l_2 \sin(\theta_1 + \theta_2) \end{aligned}$$

$$\frac{\partial \Gamma_1}{\partial \theta_2} = -m_2 l_1 l_2 \sin(\theta_2) \ddot{\theta}_1 - \frac{2p_w}{g} l_1 l_2 \sin(\theta_2) \ddot{\theta}_1 - \frac{m_2}{2} l_1 l_2 \sin(\theta_2) \ddot{\theta}_2 - \frac{p_w}{g} l_1 l_2 \sin(\theta_2) \ddot{\theta}_2 -$$

$$\frac{m_2}{2} l_1 l_2 \dot{\theta}_2 \cos \theta_2 (2\dot{\theta}_1 + \dot{\theta}_2) - m_2 g \frac{l_2}{2} \sin(\theta_1 + \theta_2) - \frac{p_w}{g} l_1 l_2 \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2 -$$

$$p_w l_2 \sin \theta_1 + \theta_2$$

$$\frac{\partial \Gamma_1}{\partial \dot{\theta}_1} = -m_2 l_1 l_2 \dot{\theta}_2 \sin \theta_2 - \frac{2p_w}{g} l_1 l_2 \dot{\theta}_2 \sin \theta_2$$

$$\frac{\partial \Gamma_1}{\partial \ddot{\theta}_1} = -m_2 l_1 l_2 \dot{\theta}_1 \sin \theta_2 - m_2 l_1 l_2 \dot{\theta}_2 \sin \theta_2 - \frac{2p_w}{g} l_1 l_2 \dot{\theta}_2 \sin \theta_2 - \frac{2p_w}{g} l_1 l_2 \dot{\theta}_1 \sin \theta_2$$



$$\frac{\partial \Gamma_1}{\partial \dot{\theta}_1} = \frac{m_1 l_1^2}{4} + \frac{(l_1^2 + b_1^2)}{12} m_1 + m_2 l_1^2 + \frac{m_2 l_2^2}{4} + m_2 l_1 l_2 \cos \theta_2 + \frac{m_2}{12} (l_2^2 + b_2^2) + \frac{p_w}{g} l_1^2 + \frac{p_w}{g} l_2^2 + 2p_w g l_1 l_2 \cos \theta_2$$

$$\frac{\partial \Gamma_1}{\partial \dot{\theta}_2} = \frac{m_2 l_2^2}{4} + \frac{m_2}{2} l_1 l_2 \cos \theta_2 + \frac{m_2}{12} (l_2^2 + b_2^2) + \frac{p_w}{g} l_2^2 + \frac{p_w}{g} l_1 l_2 \cos \theta_2$$

La dérivée du couple 2 par rapport à  $p^j$  :

$$\bullet \frac{\partial \Gamma_2}{\partial p^j} = \frac{\partial \Gamma_2}{\partial \theta_1} \frac{\partial \theta_1}{\partial p^j} + \frac{\partial \Gamma_2}{\partial \theta_2} \frac{\partial \theta_2}{\partial p^j} + \frac{\partial \Gamma_2}{\partial \dot{\theta}_1} \frac{\partial \dot{\theta}_1}{\partial p^j} + \frac{\partial \Gamma_2}{\partial \dot{\theta}_2} \frac{\partial \dot{\theta}_2}{\partial p^j} + \frac{\partial \Gamma_2}{\partial \ddot{\theta}_1} \frac{\partial \ddot{\theta}_1}{\partial p^j} + \frac{\partial \Gamma_2}{\partial \ddot{\theta}_2} \frac{\partial \ddot{\theta}_2}{\partial p^j} \quad (\text{III.50})$$

Avec :

$$\frac{\partial \Gamma_2}{\partial \theta_1} = -m_2 \frac{l_2}{2} g \sin(\theta_1 + \theta_2) - p_w l_2 \sin(\theta_1 + \theta_2)$$

$$\frac{\partial \Gamma_2}{\partial \theta_2} = \frac{m_2}{2} l_1 l_2 \sin(\theta_2) \ddot{\theta}_1 - \frac{p_w}{g} l_1 l_2 \sin(\theta_2) \ddot{\theta}_1 + \frac{m_2}{2} l_1 l_2 \dot{\theta}_1^2 \cos \theta_2 - m_2 \frac{l_2}{2} g \sin(\theta_1 + \theta_2) + \frac{p_w}{g} l_1 l_2 \cos(\theta_2) \dot{\theta}_1^2 - p_w l_2 \sin(\theta_1 + \theta_2)$$

$$\frac{\partial \Gamma_2}{\partial \dot{\theta}_1} = m_2 l_1 l_2 \dot{\theta}_1 \sin \theta_2 + 2 \frac{p_w}{g} l_1 l_2 \dot{\theta}_1 \sin \theta_2$$

$$\frac{\partial \Gamma_2}{\partial \dot{\theta}_2} = 0$$

$$\frac{\partial \Gamma_2}{\partial \ddot{\theta}_1} = \frac{l_2^2}{4} m_2 + \frac{l_1 l_2}{2} m_2 \cos \theta_2 + \frac{(l_2^2 + b_2^2)}{12} m_2 + \frac{p_w}{g} l_2^2 + \frac{p_w}{g} l_1 l_2 \cos \theta_2$$

$$\frac{\partial \Gamma_2}{\partial \ddot{\theta}_2} = m_2 \frac{l_2^2}{4} + \frac{m_2}{12} (l_2^2 + b_2^2) + \frac{p_w}{g} l_2^2$$

Avec :

$$\frac{\partial \theta_1}{\partial p_1^j} = \frac{\partial \theta_2}{\partial p_2^j} = B_{j,K}(t)$$

$$\frac{\partial \dot{\theta}_1}{\partial p_1^j} = \frac{\partial \dot{\theta}_2}{\partial p_2^j} = (K-1) \left[ \frac{1}{t_{j+K-1}-t_j} B_{j,K-1}(t) - \frac{1}{t_{j+K}-t_{j+1}} B_{j+1,K-1}(t) \right]$$

$$\frac{\partial \ddot{\theta}_1}{\partial p_1^j} = \frac{\partial \ddot{\theta}_2}{\partial p_2^j} = (K-1)(K-2) \left[ \frac{1}{t_{j+K-1}-t_j} \frac{B_{j,K-2}(t)}{t_{j+K-2}-t_j} + \left( -\frac{1}{t_{j+K}-t_{j+1}} - \frac{1}{t_{j+K-1}-t_j} \right) \frac{B_{j+1,K-2}(t)}{t_{j+K-1}-t_{j+1}} + \frac{1}{t_{j+K}-t_{j+1}} \frac{B_{j+2,K-2}(t)}{t_{j+K}-t_{j+2}} \right]$$

Où :  $j = [1 \dots m]$  et  $K = 5$

D'après ces calculs, on obtient :

$$\frac{\partial \Gamma}{\partial P} = \begin{pmatrix} \frac{\partial \Gamma_1}{\partial P} \\ \frac{\partial \Gamma_2}{\partial P} \end{pmatrix} \quad (\text{III.51})$$

❖ Calcul de la dérivée partielle du couple par rapport au poids ( $\frac{\partial \Gamma}{\partial p_w}$ )

$$\frac{\partial M}{\partial p_w} = \begin{bmatrix} \frac{l_1^2}{g} + \frac{l_2^2}{g} + \frac{2l_1l_2}{g} \cos(\theta_2) & \frac{l_2^2}{g} + \frac{l_1l_2}{g} \cos(\theta_2) \\ \frac{l_1l_2}{g} \cos(\theta_2) + \frac{l_2^2}{g} & \frac{l_2^2}{g} \end{bmatrix} \times \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$\frac{\partial h}{\partial p_w} = \begin{bmatrix} -\frac{l_1l_2}{g} \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_2) + l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ l_2 \cos(\theta_1 + \theta_2) + \frac{l_1l_2}{g} \dot{\theta}_1^2 \sin(\theta_2) \end{bmatrix}$$

Donc :

$$\frac{\partial \Gamma}{\partial p_w} =$$

$$\begin{bmatrix} \frac{l_1^2}{g} + \frac{l_2^2}{g} + \frac{2l_1l_2}{g} \cos(\theta_2) & \frac{l_2^2}{g} + \frac{l_1l_2}{g} \cos(\theta_2) \\ \frac{l_1l_2}{g} \cos(\theta_2) + \frac{l_2^2}{g} & \frac{l_2^2}{g} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$+ \begin{bmatrix} -\frac{l_1l_2}{g} \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_2) + l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ l_2 \cos(\theta_1 + \theta_2) + \frac{l_1l_2}{g} \dot{\theta}_1^2 \sin(\theta_2) \end{bmatrix}$$

(III.52)

➤ le gradient de  $J_c$  :

On a:  $J_c = J_w + J_e + J_p$

$$\nabla J_c = \left[ \frac{\partial J_c}{\partial P} \quad \frac{\partial J_c}{\partial p_w} \quad \frac{\partial J_c}{\partial P_t} \right]^T \quad (\text{III.53})$$

Avec :  $J_w = -wp_w$

$$J_e = \frac{1}{2} \int_0^{p_t t_f} \Gamma^T w_e \Gamma dt$$

$$J_p = \frac{1}{2} \int_0^{p_t t_f} \varphi_+^T W_P \varphi_+ d\eta$$

- ❖ calcul de la dérivée partielle du coût  $J_c$  par rapport au poids  $p_w$ :

$$\frac{\partial J_c}{\partial p_w} = \frac{\partial J_w}{\partial p_w} + \frac{\partial J_e}{\partial p_w} \quad (\text{III.54})$$

La dérivée du  $J_w$  par rapport au poids directement est donnée comme suit :

$$\frac{\partial J_w}{\partial p_w} = -W$$

Avec :

$$\frac{\partial J_e}{\partial p_w} = 0$$

- ❖ calcul de la dérivée partielle du coût  $J_c$  par rapport aux points de contrôle  $p^j$  :

$$\frac{\partial J_c}{\partial P} = \frac{\partial J_c}{\partial \Gamma} \frac{\partial \Gamma}{\partial P} \quad (\text{III.55})$$

La dérivée du  $J_c$  par rapport au couple directement est donnée comme suit :

$$\frac{\partial J_c}{\partial \Gamma} = \frac{\partial J_w}{\partial \Gamma} + \frac{\partial J_e}{\partial \Gamma} \quad (\text{III.56})$$

Ainsi, La dérivée du  $J_w$  par rapport au couple est donnée comme suit :

$$\frac{\partial J_w}{\partial \Gamma} = \frac{\partial}{\partial \Gamma} (-w p_w) = 0$$

Ainsi, La dérivée du  $J_e$  par rapport au couple est donnée comme suit :

$$\frac{\partial J_e}{\partial \Gamma} = \frac{\partial}{\partial \Gamma} \left( \frac{1}{2} \int_0^{p_t t_f} \Gamma^T w_e \Gamma dt \right) = \left( \int_0^{p_t t_f} w_e \Gamma dt \right)^T$$

Avec :

$$\frac{\partial \Gamma}{\partial P} : \text{on l'a calculée dans la première partie.}$$

- ❖ calcul de la dérivée du coût  $J_c$  par rapport au facteur d'échelle  $p_t$  :

$$\frac{\partial J_c}{\partial p_t} = \frac{\partial J_w}{\partial p_t} + \frac{\partial J_e}{\partial p_t} = \frac{\partial}{\partial p_t} \left[ \left( \frac{1}{2} \int_0^{p_t t_f} \Gamma^T w_e \Gamma dt \right) + (-w p_w) \right] = \frac{t_f}{2} \Gamma^T w_e \Gamma + 0 \quad (\text{III.57})$$

### III.4 Résultats de simulation :

Pour les paramètres du robot, nous avons pris :

$$m_1=14\text{kg}; m_2=7\text{kg}; l_1=0.5\text{m}; l_2=0.25\text{m}; b_1=0.01\text{m}; b_2=0.01\text{m}; g=9.98\text{m/s}^2.$$

Au départ, nous nous sommes appuyés sur les paramètres initiaux suivants :

$$p_1 = [90, 90, 84.30, 78.50, 72.57, 66.45, 60.03, 53.15, 45.59, 36.88, 25.85, 0, 0, 0];$$

$$p_2 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];$$

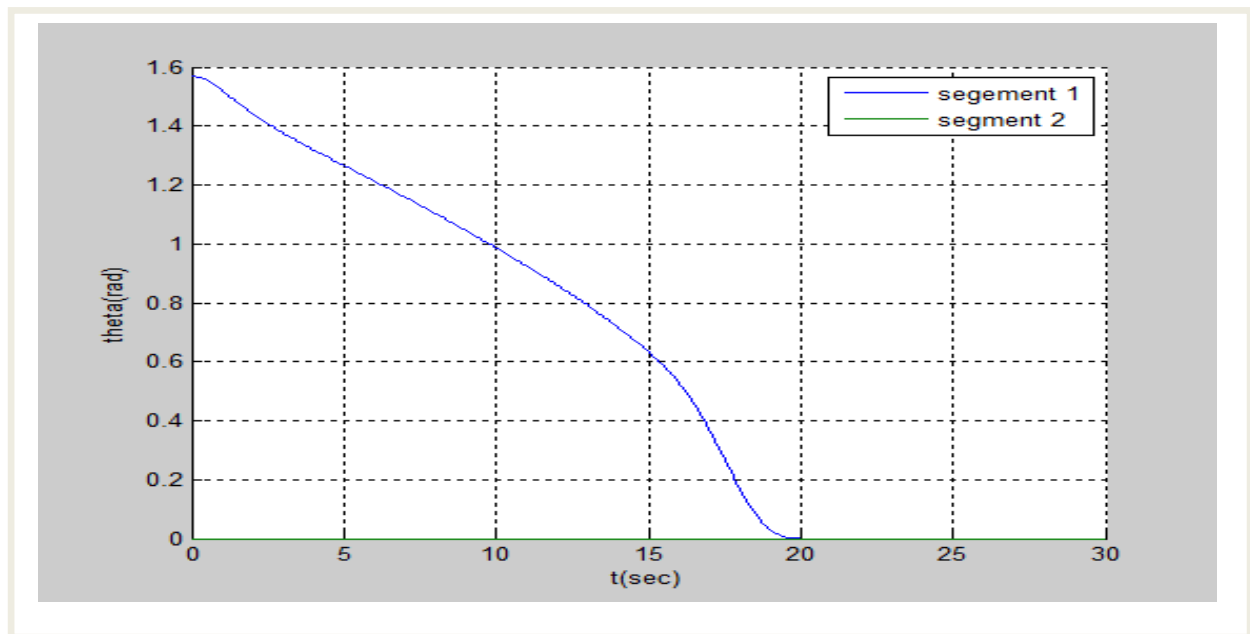
$$p_w = 30 \text{ kg};$$

$$p_t = 10;$$

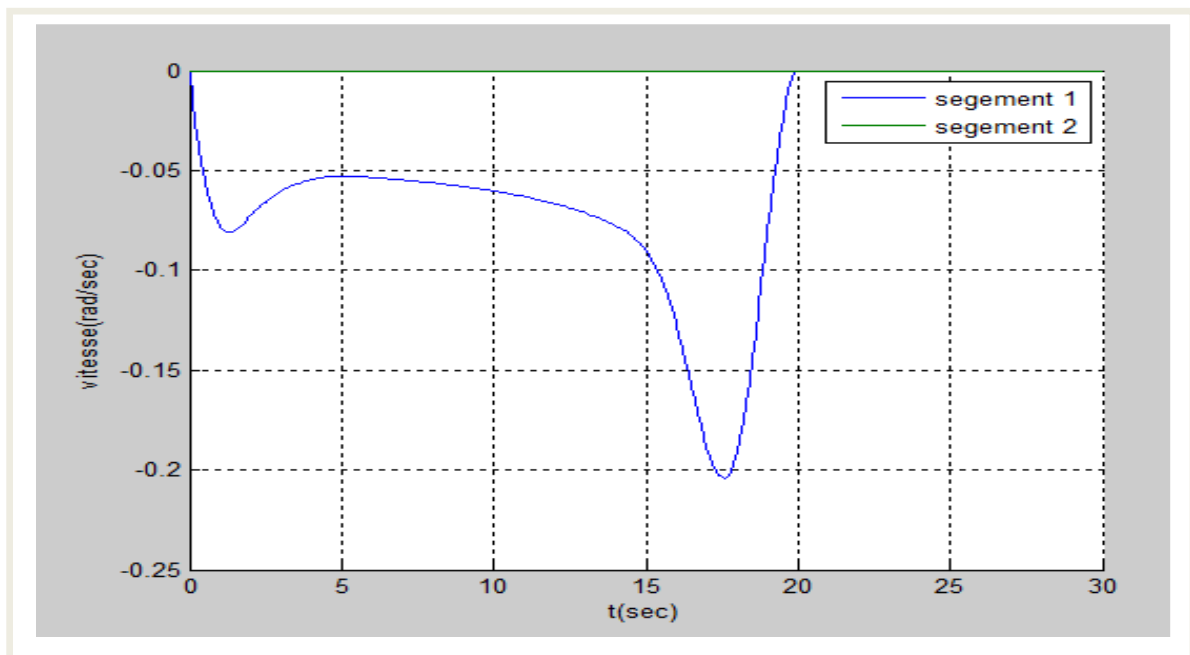
$$\Gamma_1 \text{ max} = 100 \text{ N.m}; \Gamma_2 \text{ max} = 200 \text{ N.m}; \Gamma_1 \text{ min} = -100 \text{ N.m}; \Gamma_2 \text{ min} = -200 \text{ N.m}$$

Le cout initial est :

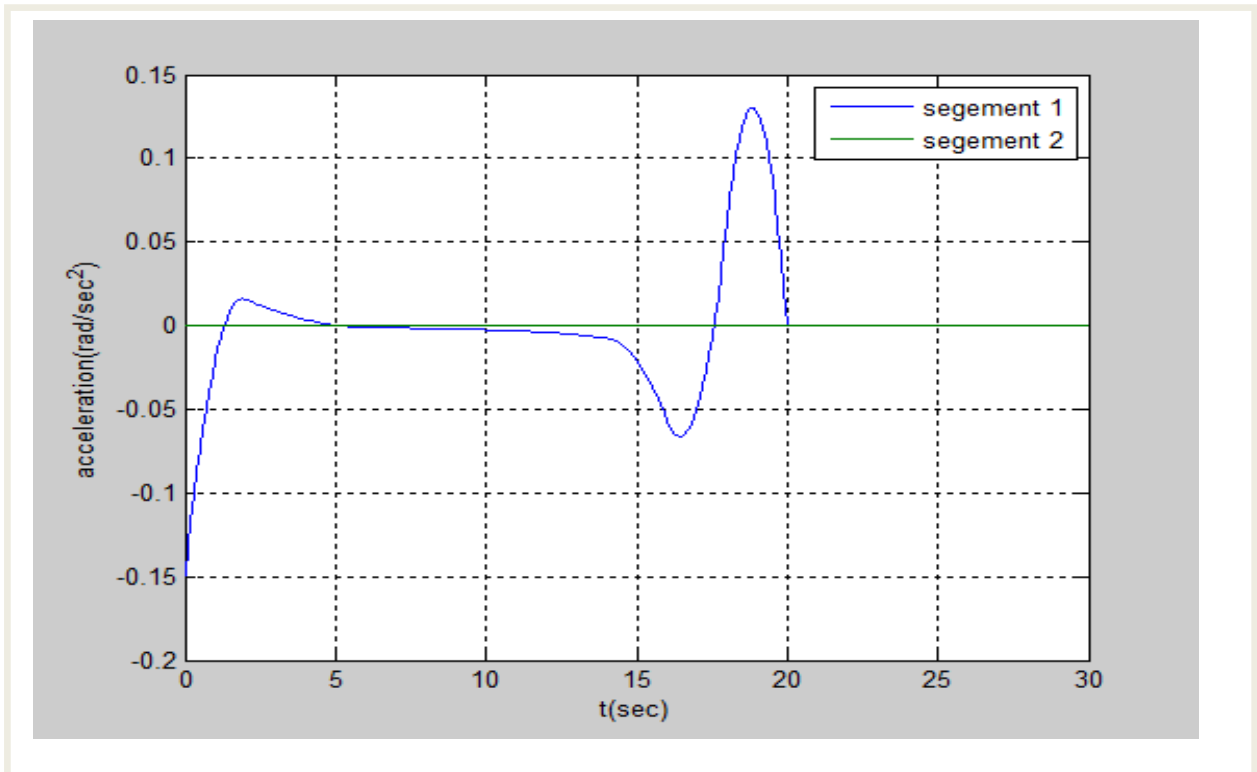
$$J_c \text{ initial} = 4.10.10^4.$$



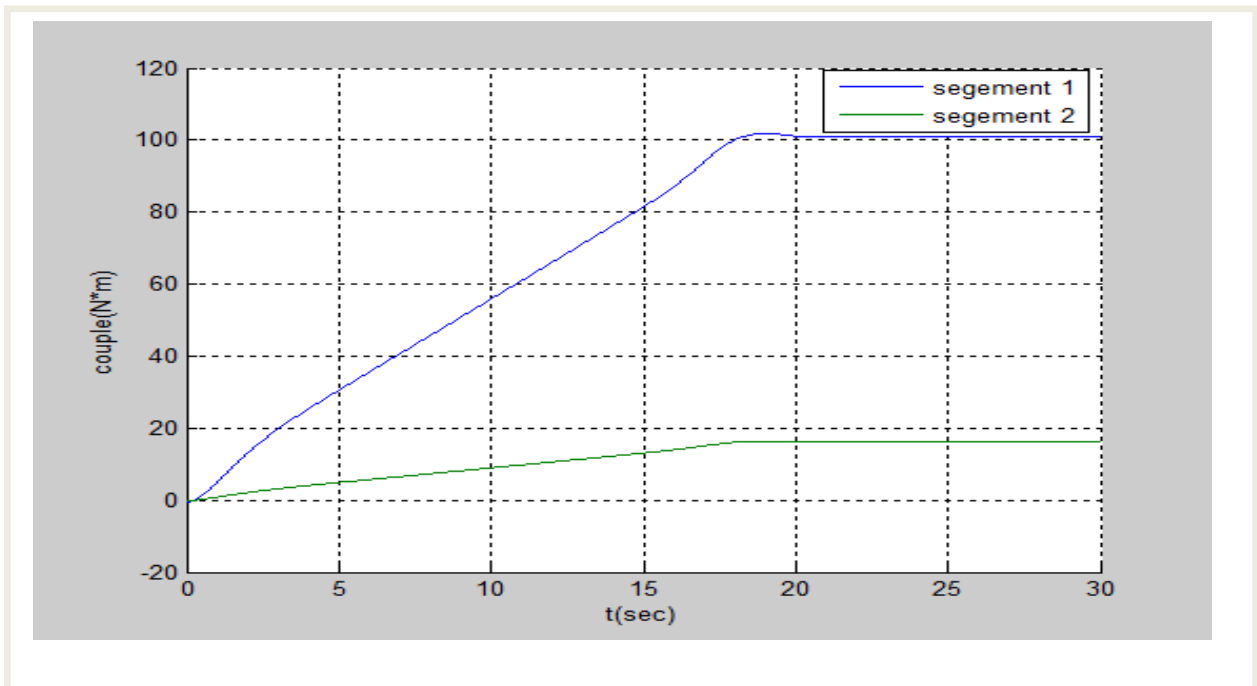
**Fig.III.4** les positions articulaires  $\theta_1$  et  $\theta_2$  pour la trajectoire initiale



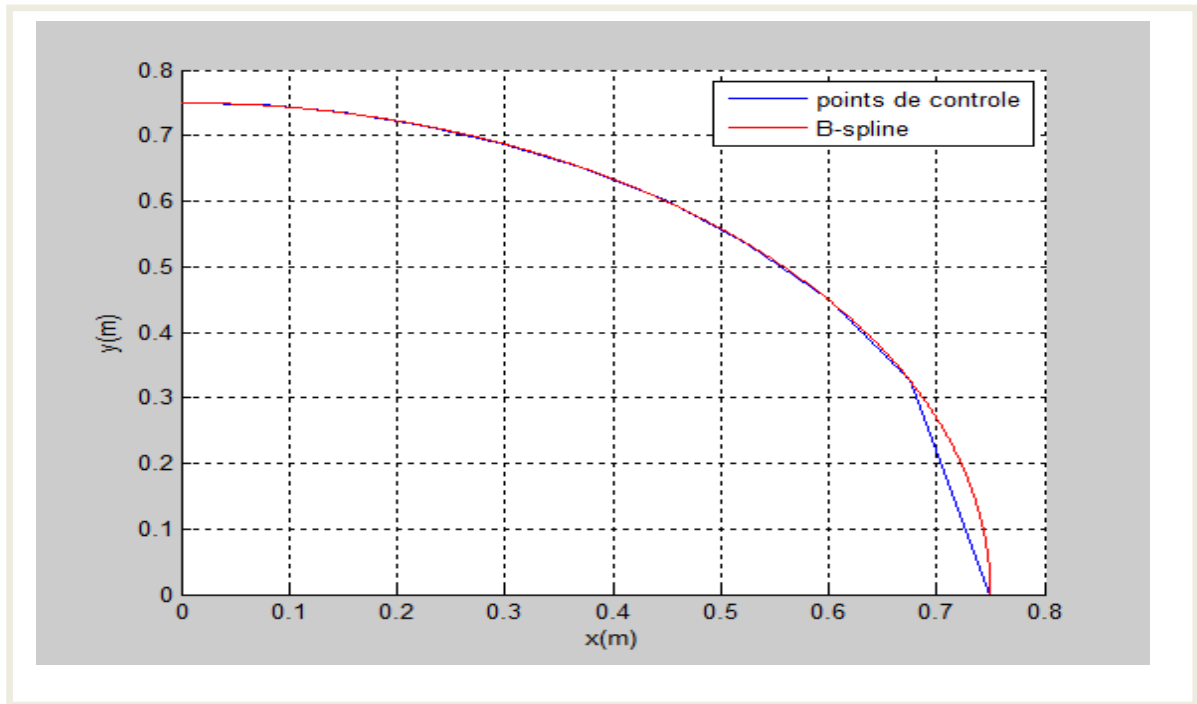
**Fig.III.5** les vitesses articulaires  $\dot{\theta}_1$  et  $\dot{\theta}_2$  pour la trajectoire initiale



**Fig.III.6** les accélérations articulaires  $\ddot{\theta}_1$  et  $\ddot{\theta}_2$  pour la trajectoire initiale



**Fig.III.7** les couples initiaux



**Fig.III.8** génération de la trajectoire initiale avec deux méthodes

**Premier essai :**

- On prend les contraintes sur les vitesses articulaires:

$$-130 * \frac{\pi}{180} \leq \dot{\theta}_1 \leq 130 * \frac{\pi}{180} \text{ et } -150 * \frac{\pi}{180} \leq \dot{\theta}_2 \leq 150 * \frac{\pi}{180},$$

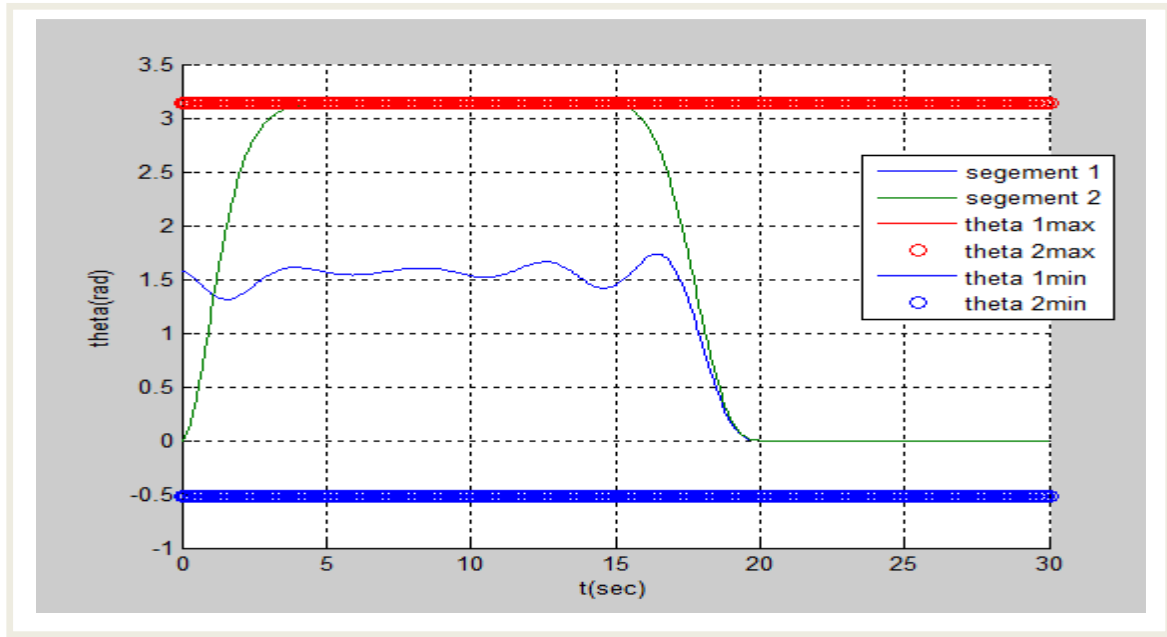
- Les contraintes sur les positions articulaires :

$$-180 * \frac{\pi}{180} \leq \theta_1 \leq 180 * \frac{\pi}{180} \text{ et } -180 * \frac{\pi}{180} \leq \theta_2 \leq 180 * \frac{\pi}{180},$$

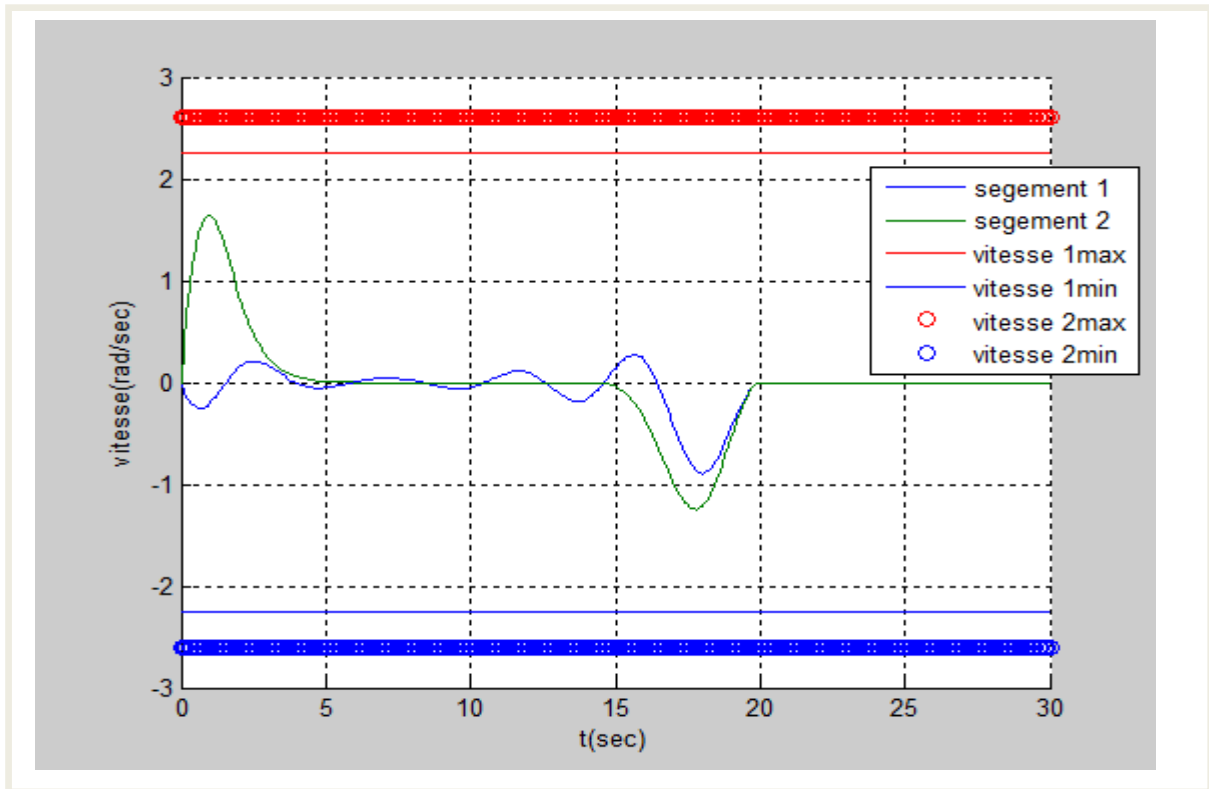
- Les contraintes sur les couples :

$$-100 \leq \Gamma_1 \leq 100 \text{ et } -200 \leq \Gamma_2 \leq 200,$$

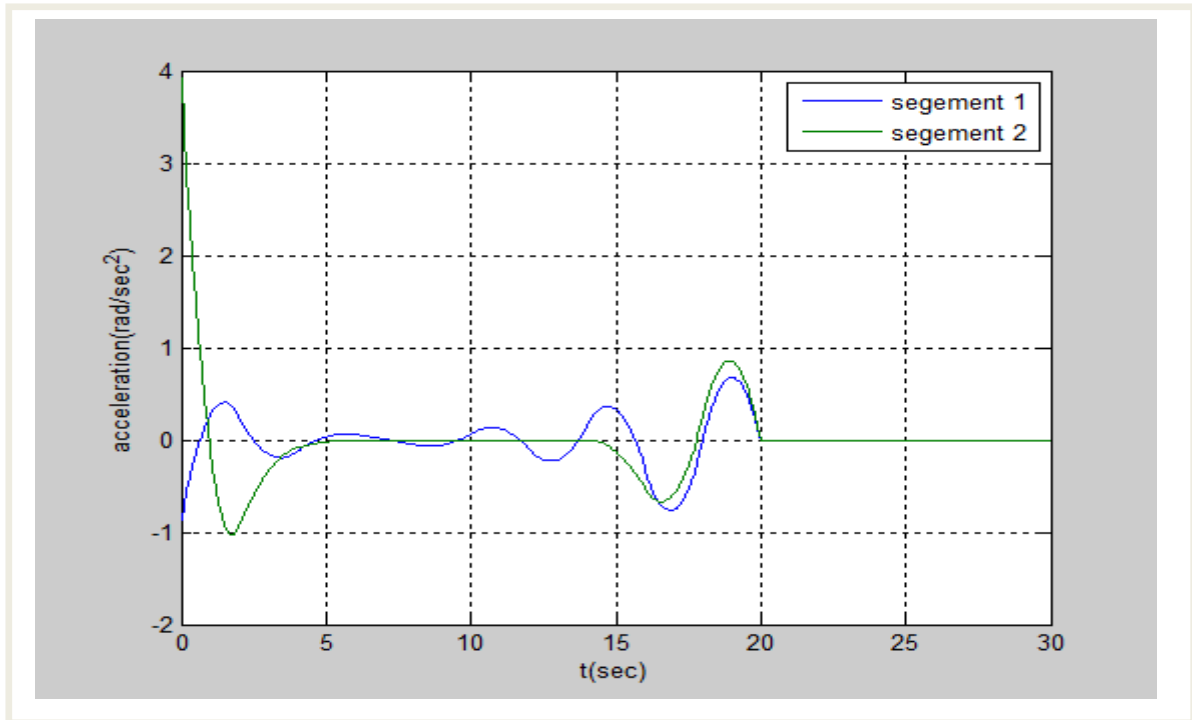
$$\theta_1(0) = 90, \theta_1(t_f) = 0, \theta_2(0) = 0, \theta_2(t_f) = 0,$$



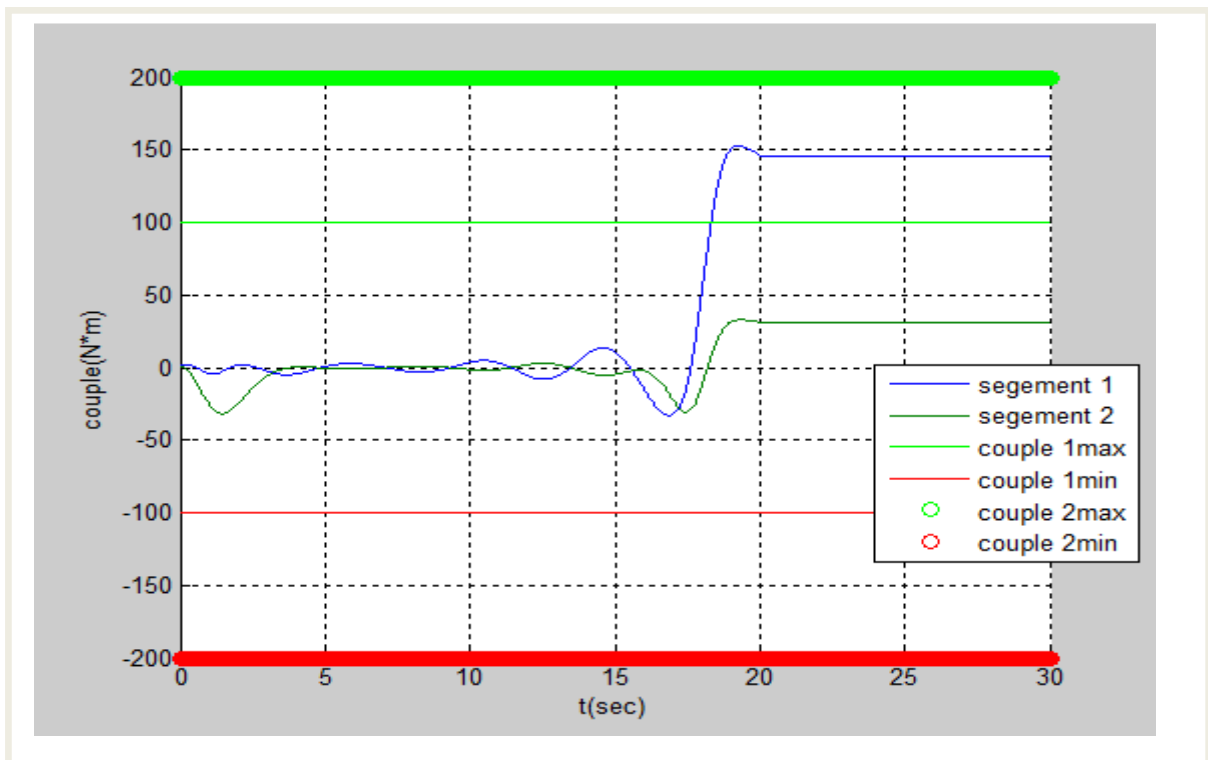
**Fig.III.9** Variations des positions articulaires durant la trajectoire optimale



**Fig.III.10** les vitesses articulaires le long de la trajectoire optimale

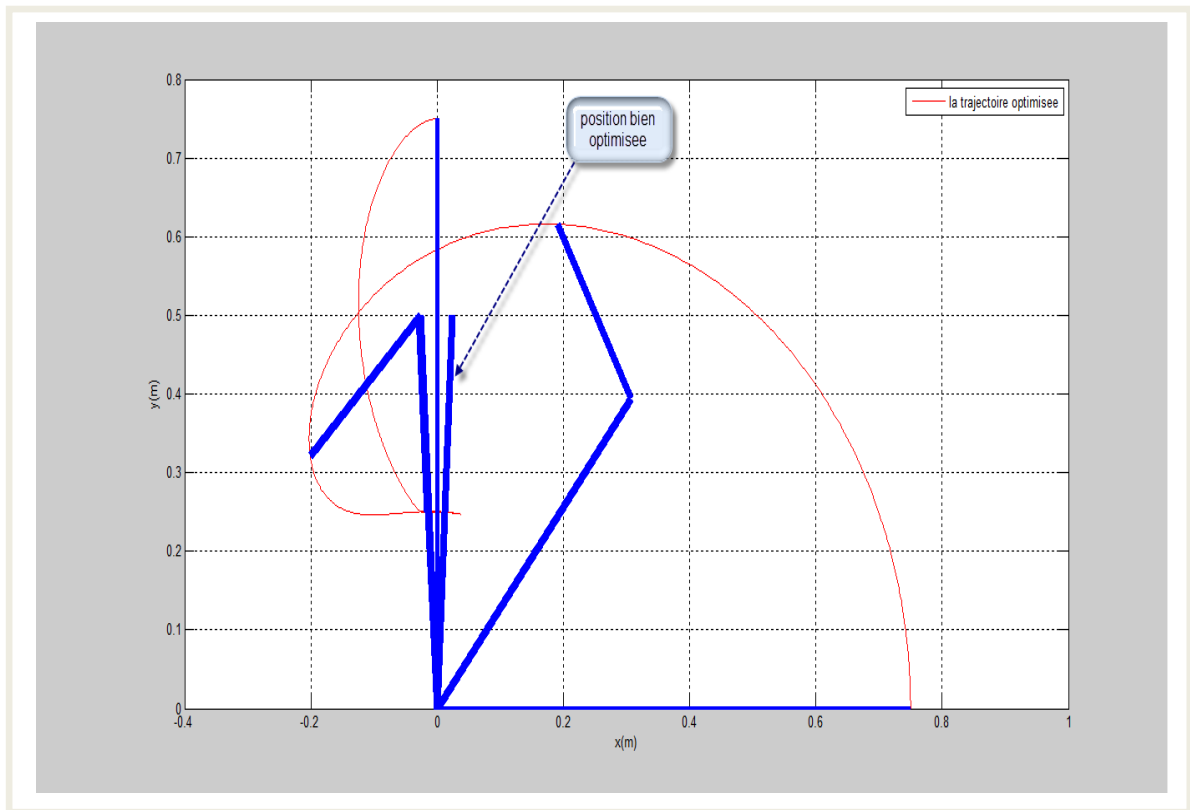


**Fig.III.11** Les accélérations articulaires optimales



**Fig.III.12** Les couples articulaires durant la trajectoire optimale





**Fig.III.13** Le chemin optimisé de l'outil dans l'espace cartésien

### Les résultats de premier essai :

Les points de contrôle (Coefficients du B-Spline):

$$p1 = [90,90,89,91,87,93,84,98,76,105,45,0,0,0];$$

$$p2 = [0,0,20,180,180,180,180,180,180,180,45,0,0,0];$$

Le coût optimisé est :

$$J_c = 1.87 \cdot 10^4 ;$$

Le poids maximum que peut prendre le robot dans le cas de la trajectoire optimale :

$$p_w = 90 ;$$

Le facteur d'échelle est :

$$p_t = 10 ;$$

Après la simulation on remarque que :

- La trajectoire optimale améliore le coût.
- Le poids solvable par le robot est maximisé.

- Les contraintes sur les positions articulaires et les vitesses sont vérifiées.
- Le couple est limité entre le couple maximum et le couple minimum (pas de surcharge sur les actionneurs).

### Deuxième essai :

- Pour ces essais nous allons changer les contraintes comme suit :

- vitesses articulaires:

$$-30 * \frac{\pi}{180} \leq \dot{\theta}_1 \leq 30 * \frac{\pi}{180} \text{ et } -50 * \frac{\pi}{180} \leq \dot{\theta}_2 \leq 50 * \frac{\pi}{180},$$

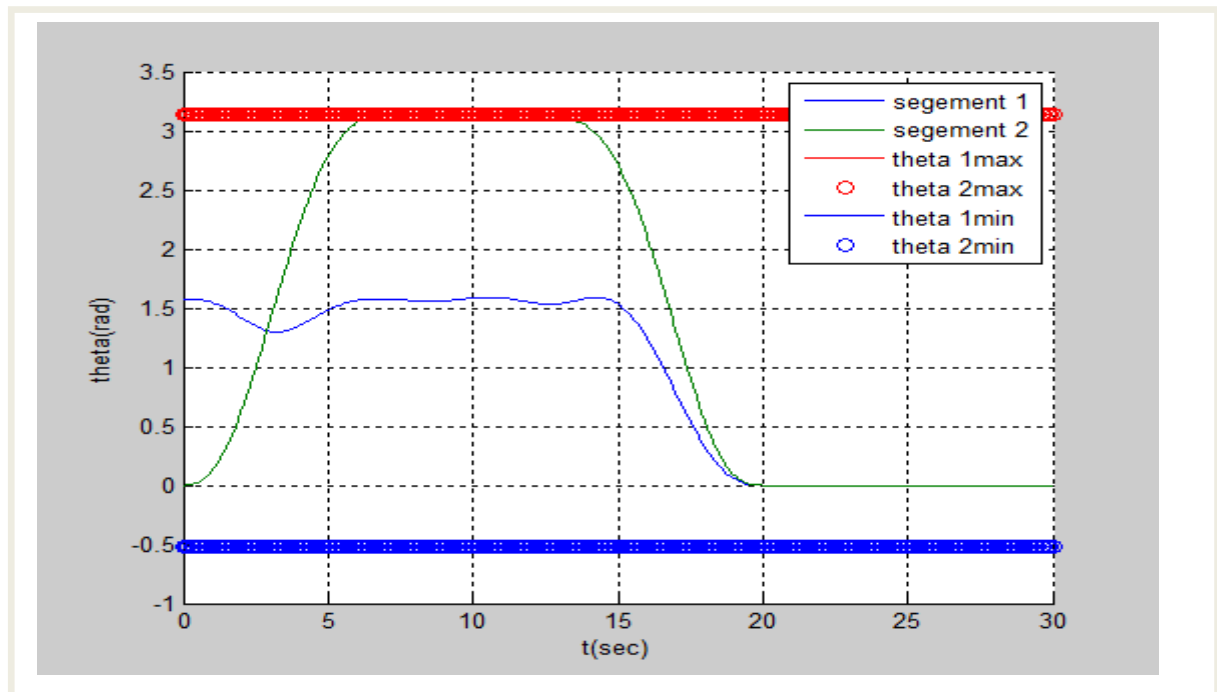
- positions articulaires :

$$-180 * \frac{\pi}{180} \leq \theta_1 \leq 180 * \frac{\pi}{180} \text{ et } -180 * \frac{\pi}{180} \leq \theta_2 \leq 180 * \frac{\pi}{180},$$

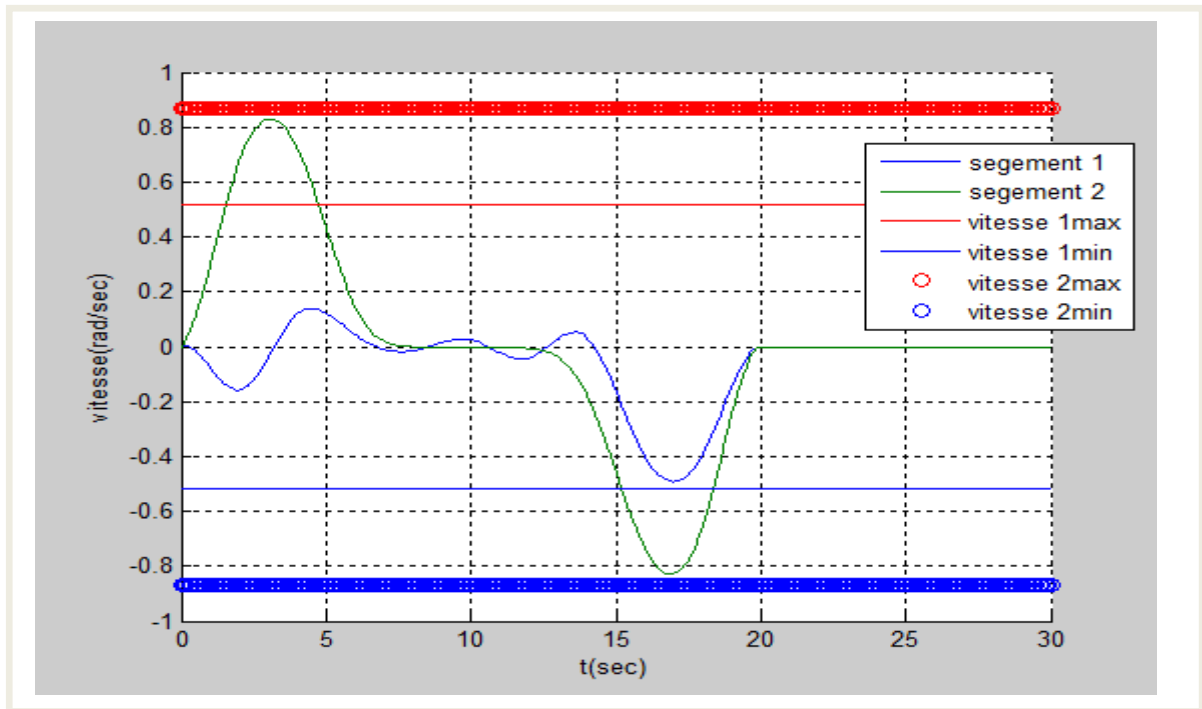
- couples articulaires :

$$-100 \leq \Gamma_1 \leq 100 \text{ et } -200 \leq \Gamma_2 \leq 200,$$

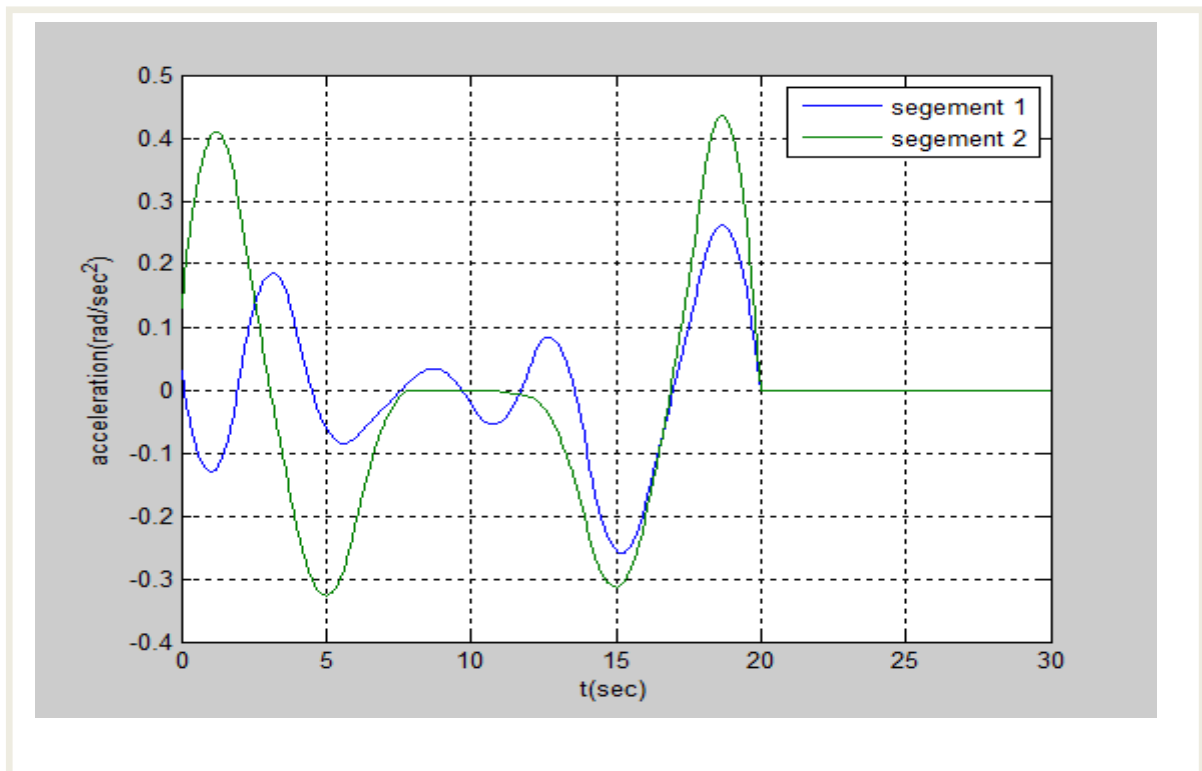
$$\theta_1(0) = 90, \theta_1(t_f) = 0, \theta_2(0) = 0, \theta_2(t_f) = 0,$$



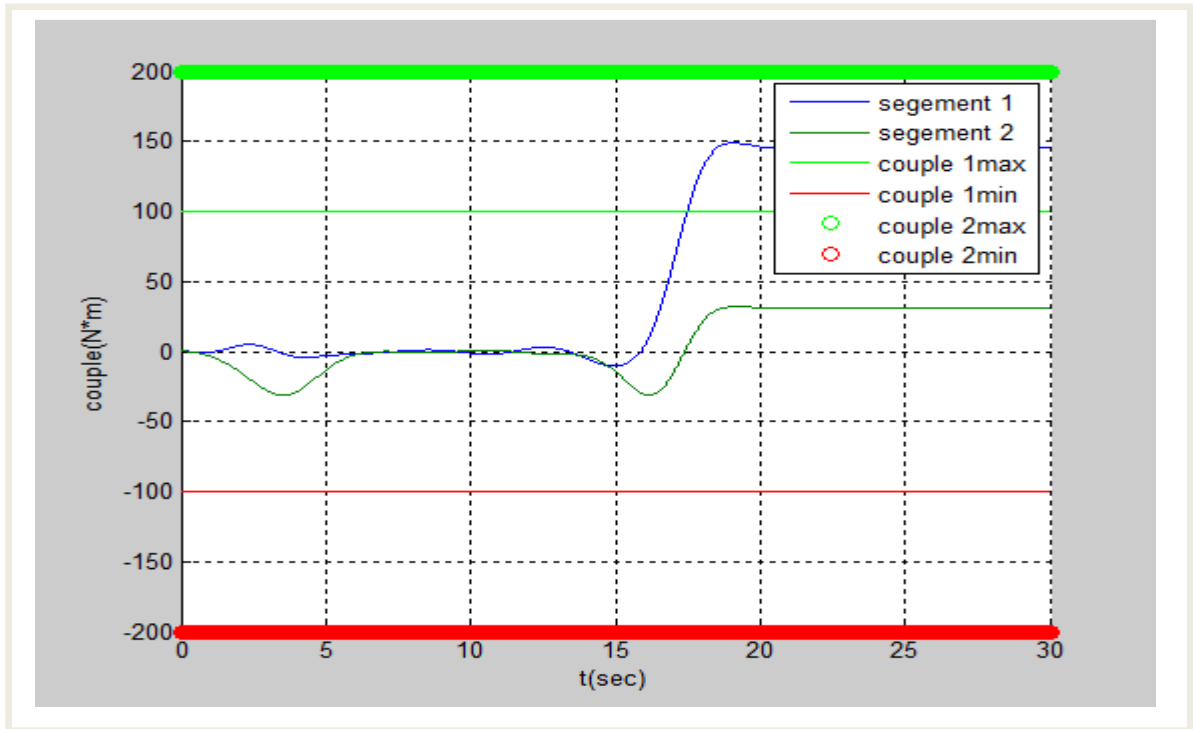
**Fig.III.14** Variations des positions articulaires durant la trajectoire optimale



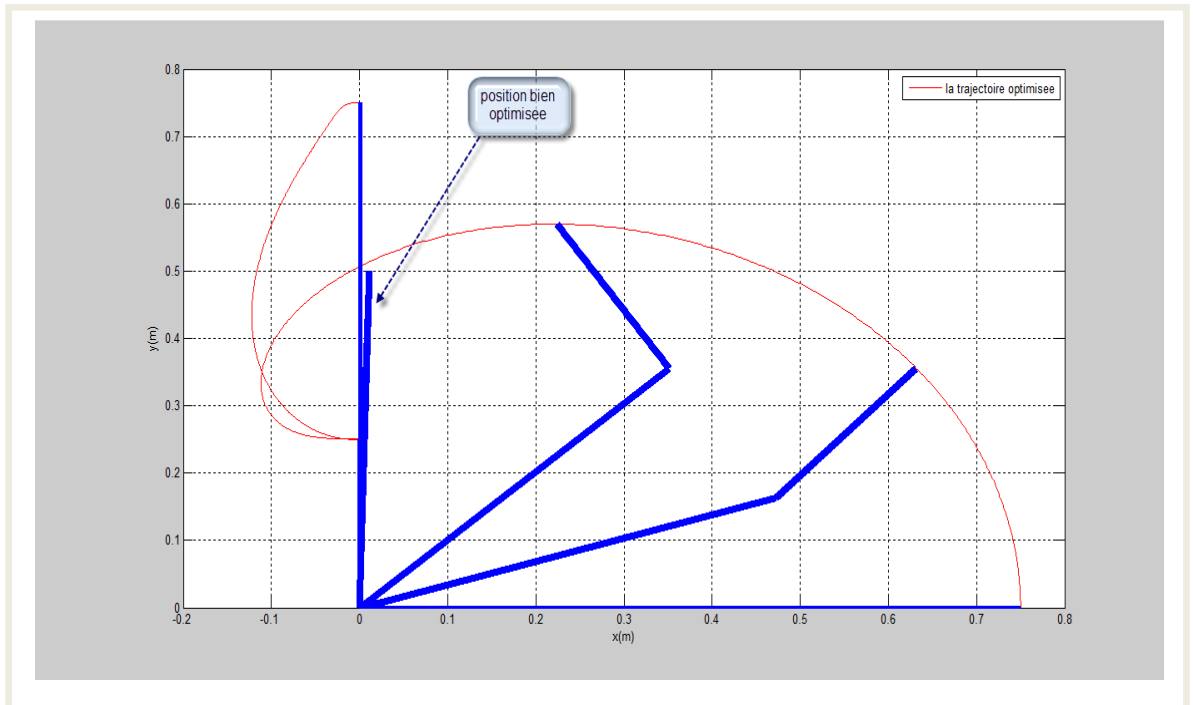
**Fig.III.15** les vitesses articulaires le long de la trajectoire optimale



**Fig.III.16** Les accélérations articulaires optimales



**Fig.III.17** Les couples articulaires durant la trajectoire optimale



**Fig.III.18** Le chemin optimisé de l'outil dans l'espace cartésien

**Les résultats de deuxième essai :**

Les points de contrôle :

$$p_1 = [90,90,91,60,90,91.71,86.61,95.9,80.21,105,45,0,0,0] ;$$

$$p_2 = [0,0,5,80,180,180,180,180,180,175,75,0,0,0] ;$$

Le coût optimisé est:

$$J_c = 2.10 \cdot 10^4 ;$$

Le poids optimisé est:

$$p_w = 90 ;$$

Le facteur d'échelle est :

$$p_t = 10 ;$$

Après la simulation on remarque que :

- Dans le cas de la trajectoire optimale le coût est amélioré;
- le poids est maximisé.
- Les contraintes sur les positions et les vitesses articulaires sont vérifiées ;
- le couple est limité entre le couple maximum et le couple minimum (pas de surcharge sur les actionneurs) ;

**Troisième essai :**

- On prend les limites sur les vitesses articulaires :

$$-130 * \frac{\pi}{180} \leq \dot{\theta}_1 \leq 130 * \frac{\pi}{180} \text{ et } -130 * \frac{\pi}{180} \leq \dot{\theta}_2 \leq 130 * \frac{\pi}{180},$$

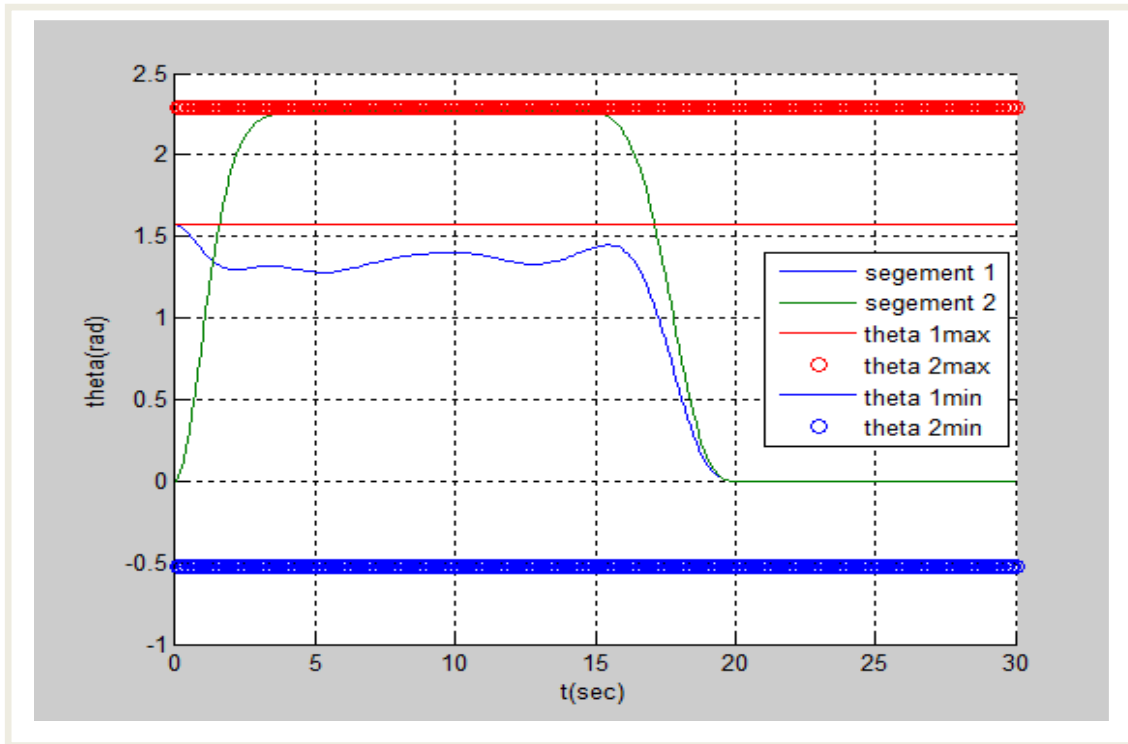
- Et les limites sur les positions articulaires :

$$-30 * \frac{\pi}{180} \leq \theta_1 \leq 90 * \frac{\pi}{180} \text{ et } -30 * \frac{\pi}{180} \leq \theta_2 \leq 130 * \frac{\pi}{180},$$

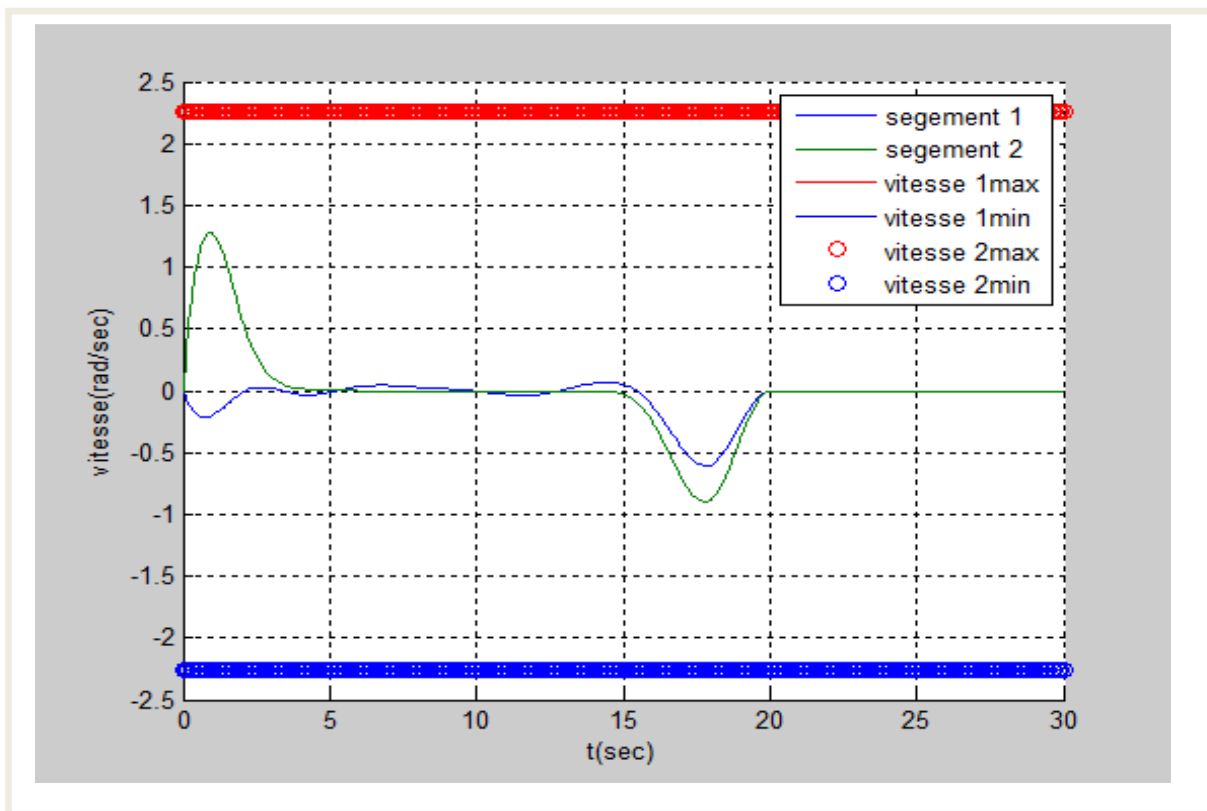
- Et les limites sur les couples :

$$-100 \leq \Gamma_1 \leq 100 \text{ et } -200 \leq \Gamma_2 \leq 200,$$

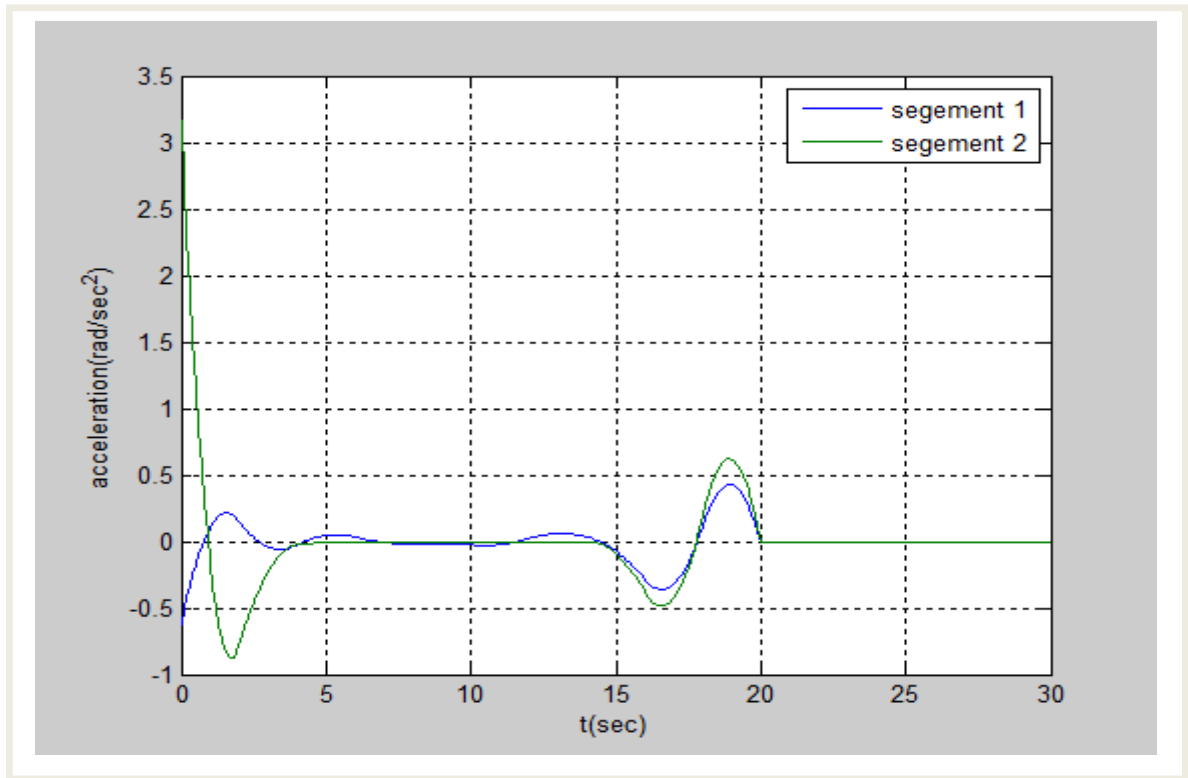
$$\theta_1(0) = 90, \theta_1(t_f) = 0, \theta_2(0) = 0, \theta_2(t_f) = 0,$$



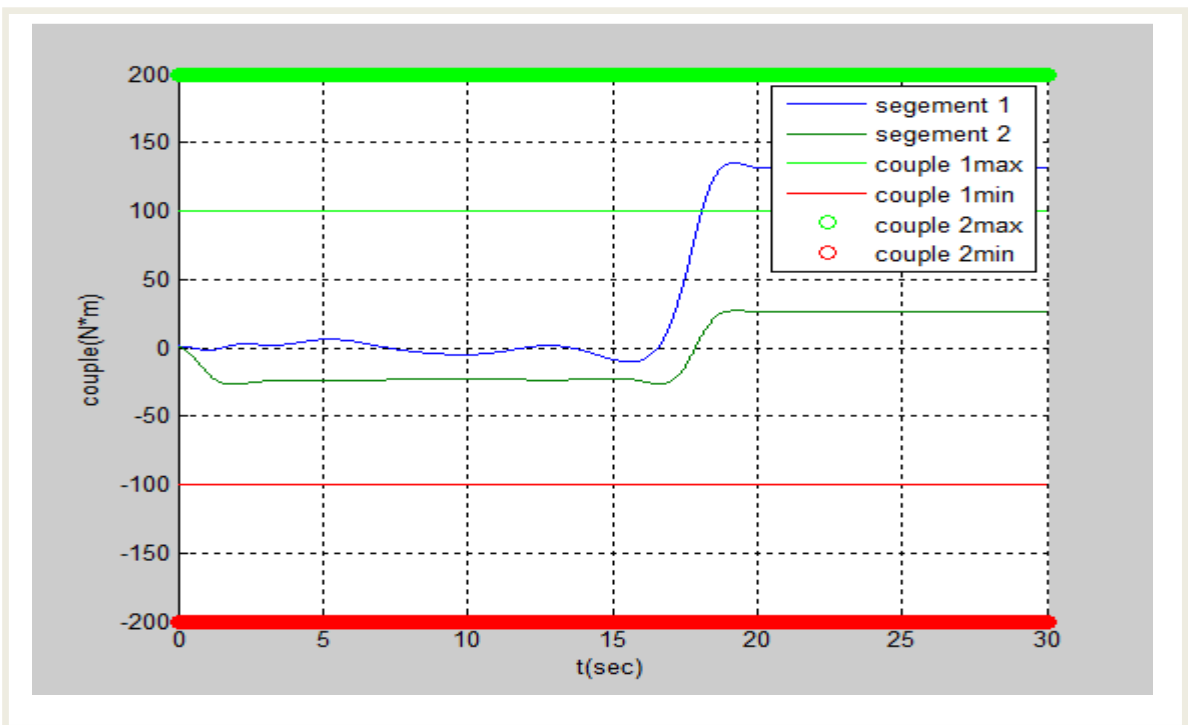
**Fig.III.19** Variations des positions articulaires durant la trajectoire optimale



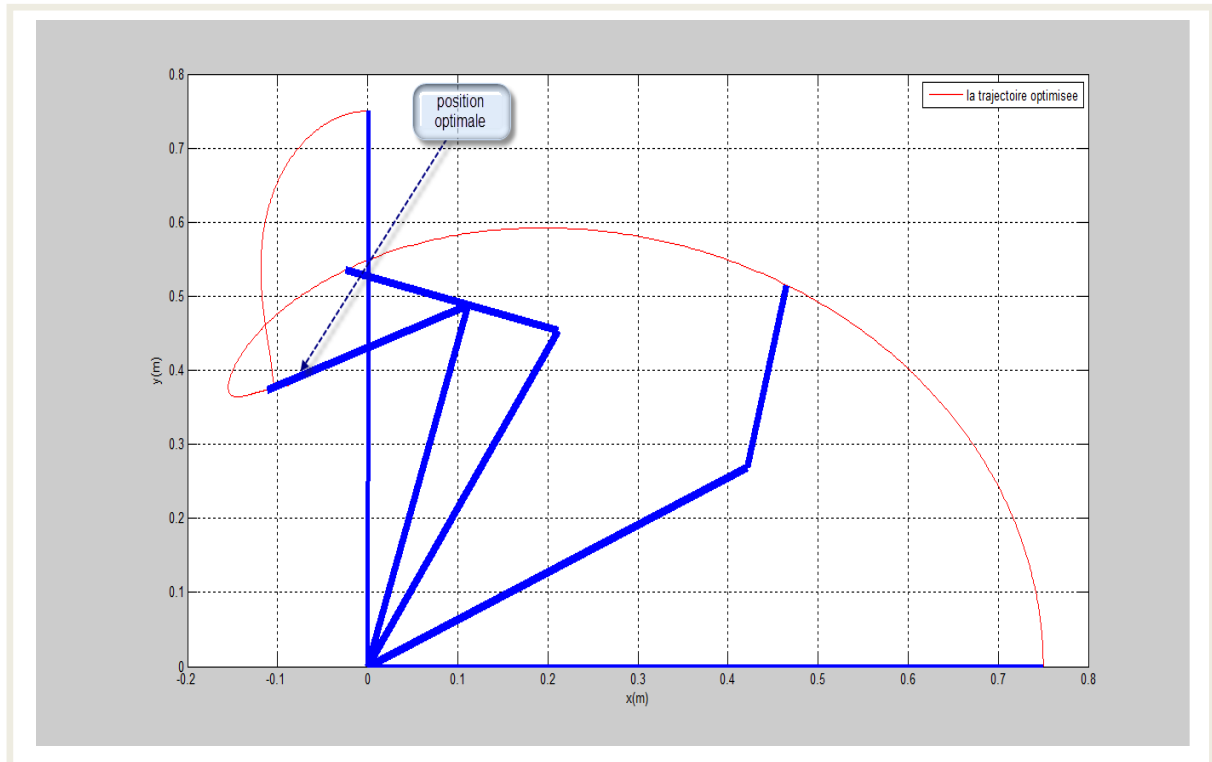
**Fig.III.20** les vitesses articulaires le long de la trajectoire optimale



**Fig.III.21** Les accélérations articulaires optimales



**Fig.III.22** Les couples articulaires durant la trajectoire optimale



**Fig.III.23** Le chemin optimisé de l'outil dans l'espace cartésien

**Les résultats de la troisième simulation :**

Les coefficients du B-spline pour la trajectoire optimale sont :

$$p1 = [90,90,66.29,81.77,69.34,77.58,80.30,81.26,72.25,83.43,90,0,0,0] ;$$

$$p2 = [0,0,120,130,130,130,130,130,130,130,0,0,0] ;$$

Le coût optimisé est :

$$J_c = 2.07 \cdot 10^4 ;$$

Le poids maximal solvable par le robot :

$$p_w = 70 ;$$

Le facteur d'échelle et :

$$p_t = 10 ;$$

Après la simulation on remarque que :

- La trajectoire optimale améliore le coût ;
- le poids solvable est maximisé ; Néanmoins, la diminution de l'intervalle de variation des variables articulaires a pour effet la diminution de ce poids ceci est dû



au faite que l'on moins de possibilités pour éviter les configurations qui nécessitent de plus grand efforts.

- les contraintes sur les positions et les vitesses articulaires sont vérifiées ;
- La contrainte sur la position articulaire du segment 2 est active durant la trajectoire optimale, mais elle reste toujours vérifiée malgré qu'elle soit plus serrée.
- le couple est limité entre le couple maximum et le couple minimum (pas de surcharge sur les actionneurs) ;

#### Quatrième essai :

- On prend les limites sur les vitesses articulaires :

$$-50 * \frac{\pi}{180} \leq \dot{\theta}_1 \leq 50 * \frac{\pi}{180} \text{ et } -50 * \frac{\pi}{180} \leq \dot{\theta}_2 \leq * \frac{\pi}{180} 50 ,$$

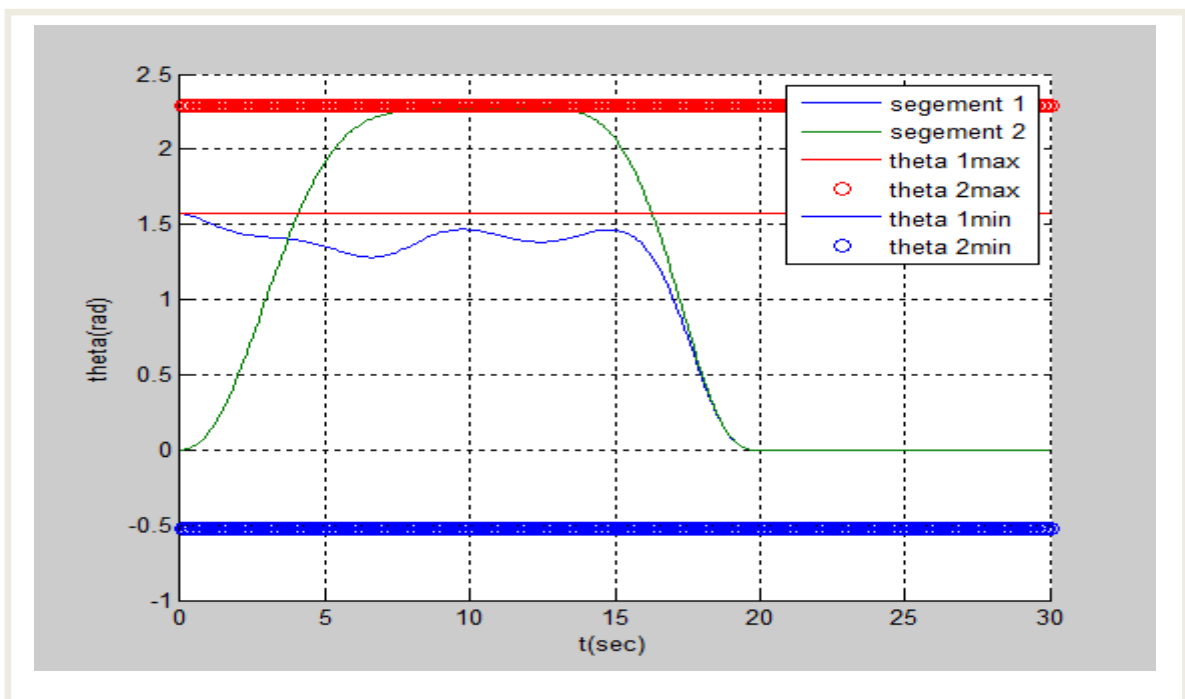
- Et les limites sur les positions articulaires :

$$-30 * \frac{\pi}{180} \leq \theta_1 \leq 90 * \frac{\pi}{180} \text{ et } -30 * \frac{\pi}{180} \leq \theta_2 \leq * \frac{\pi}{180} 130 ,$$

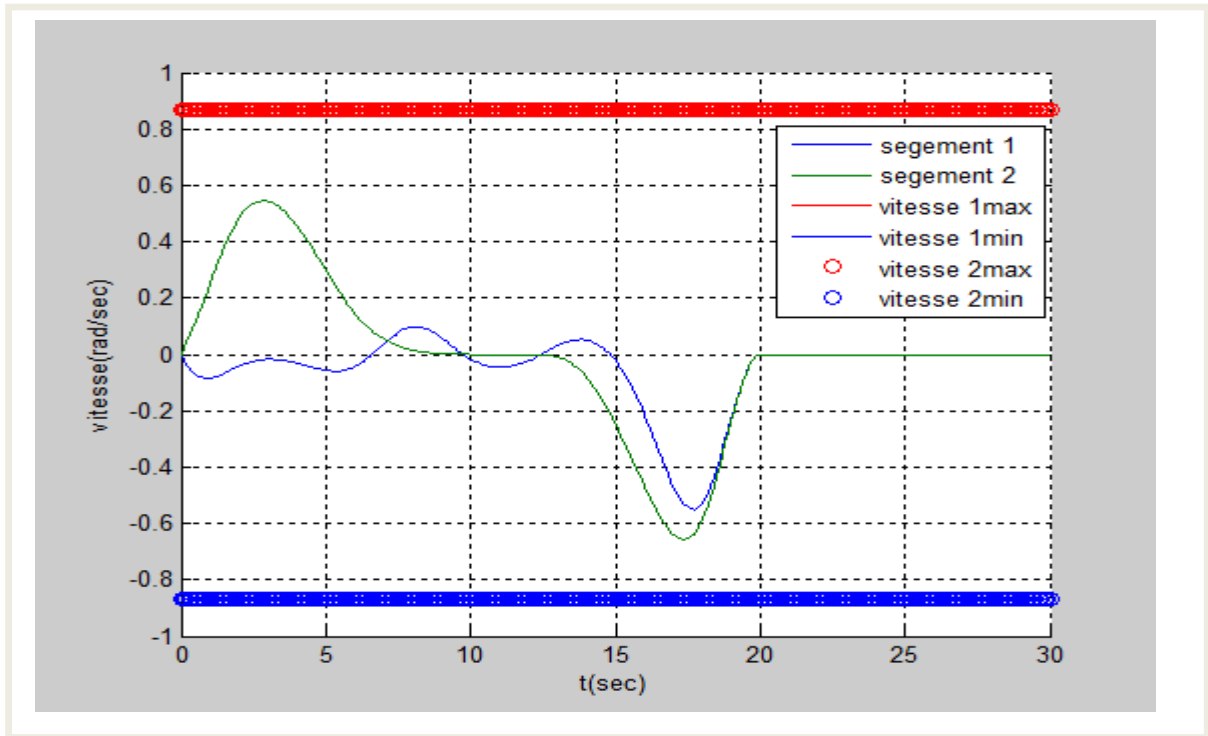
- Et les limites sur les couples :

$$-100 \leq \Gamma_1 \leq 100 \text{ et } -200 \leq \Gamma_2 \leq 200 ,$$

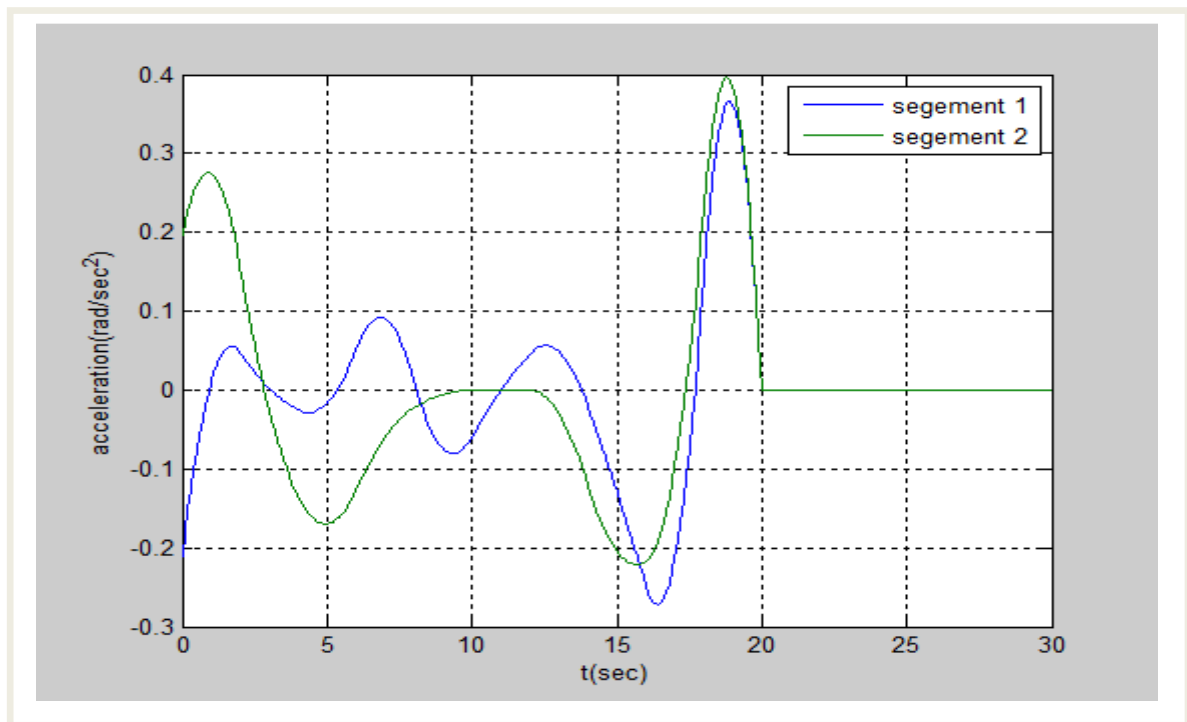
- $\theta_1(0) = 90, \theta_1(t_f) = 0, \theta_2(0) = 0, \theta_2(t_f) = 0,$



**Fig.III.24** Variations des positions articulaires durant la trajectoire optimale



**Fig.III.25** les vitesses articulaires le long de la trajectoire optimale



**Fig.III.26** Les accélérations articulaires optimales

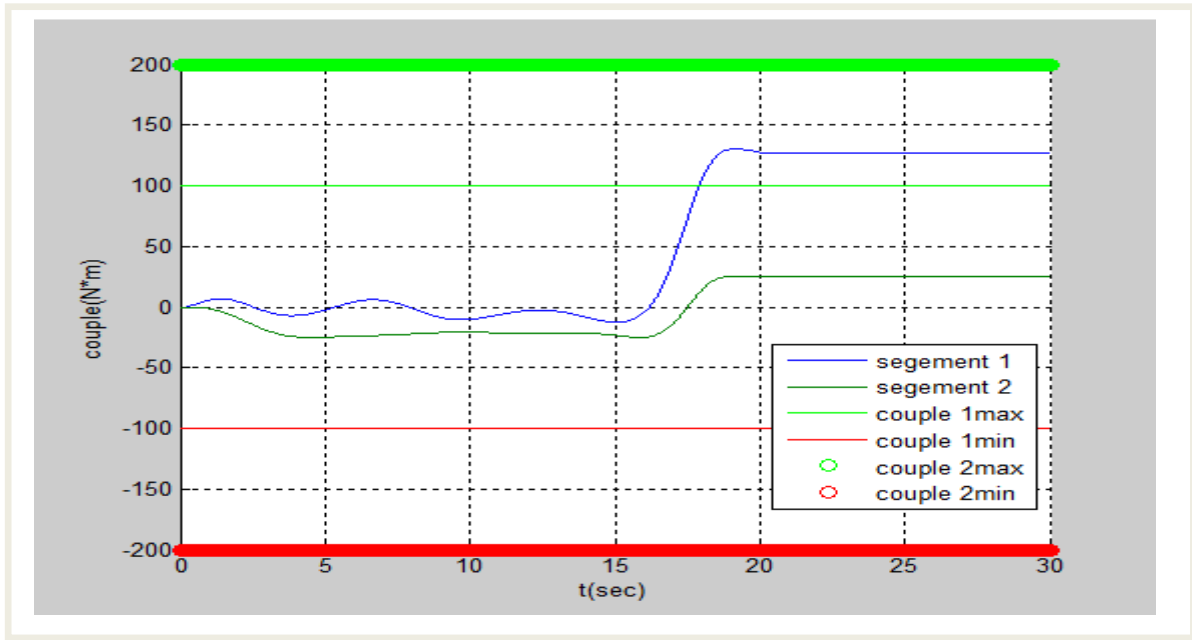


Fig.III.27 Les couples articulaires durant la trajectoire optimale

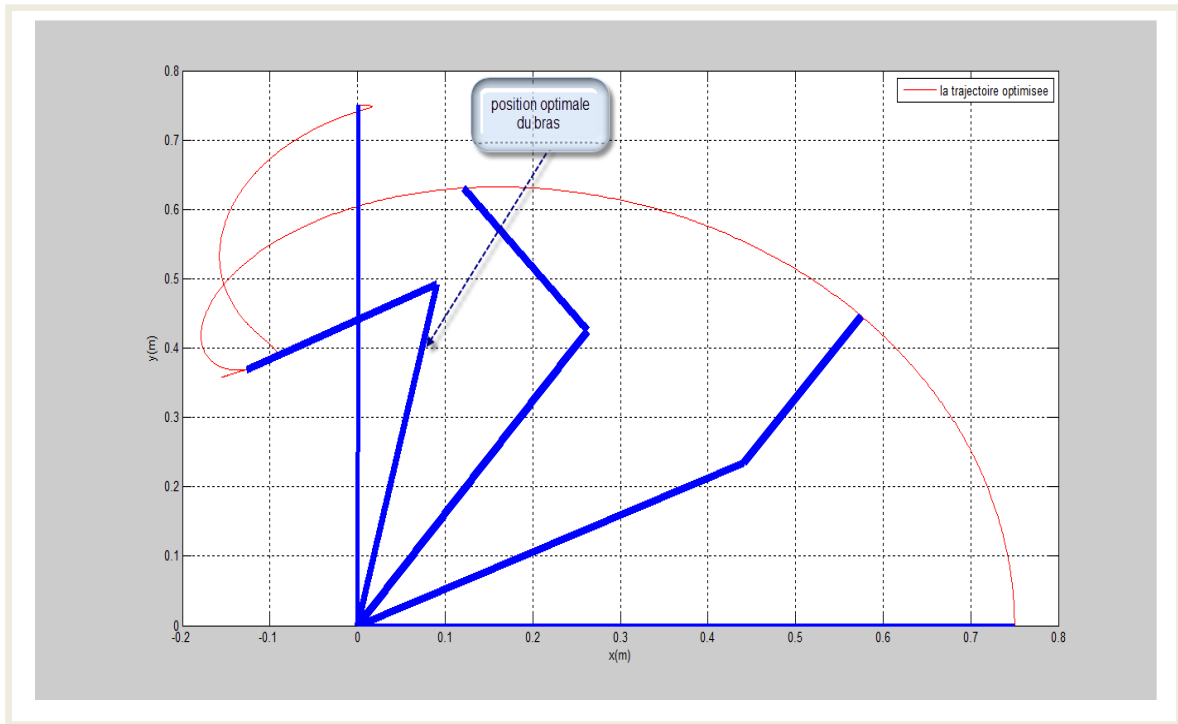


Fig.III.28 Le chemin optimisé de l'outil dans l'espace cartésien

**Les résultats de la dernière simulation :**

Les points de contrôle optimal :

$$P1 = [90,90,81.93,81.45,79.5,67,88.63,81.52,75.94,89.8,75,0,0,0] ;$$

$$P2 = [0,0,7.48,61.42,119.61,130,130,130,130,130,75,0,0,0] ;$$

Le coût optimisé est :

$$J_c = 2.35 \cdot 10^4 ;$$

Le poids optimisé est :

$$p_w = 65 ;$$

Le facteur d'échelle est :

$$p_t = 10 ;$$

Après la simulation on remarque que :

- la trajectoire optimise le coût ;
- le cout est diminué ;
- le poids est maximisé ;
- les contrainte sur la position et la vitesse sont vérifier ;
- le couple est limité entre le couple maximum et le couple minimum (pas de surcharge sur les actionneurs) ;

**Commentaire :**

La figure (**Fig.III.13**) représente la meilleure trajectoire optimale obtenue, ainsi que quelques configurations par lesquelles passe le robot le long de cette trajectoire. On remarque que le robot ne prend pas un chemin directe vers le point de destination, mais il passe par une singularité dans laquelle les deux segments sont en position vertical ( $\theta_1 = \frac{\pi}{2}, \theta_2 = \pi$ ). Dans cette configuration les couples qu'applique le poids de la charge sur les deux articulations sont nulles, car ce dernier (poids) coupe les axes de rotation des deux articulations. Ici c'est la structure mécanique du robot qui porte le poids mais les moteurs des deux articulations ne contribuent pas. Au voisinage de cette configuration, les deux couples restent faibles. Ainsi, cette configuration est idéale pour la minimisation des couples que doit générer les deux actionneurs et par suite le critère de performance. D'un autre côté, cette configuration permet d'augmenter le poids solvable par le robot ( $p_w$ ) car ce dernier n'est pas complètement appliqué aux actionneurs qui ne seront pas poussés à leurs limites.

### III.5 Conclusion :

L'objectif de ce dernier chapitre est la mise en application de ce que nous avons identifié comme méthodes efficaces dans les chapitres antérieurs.

Dans la première étape, nous avons utilisé la formulation de Lagrange pour le calcul du modèle dynamique du robot pour pouvoir déterminer les couples des actionneurs en fonction des accélérations articulaires.

La deuxième étape était dédiée à la génération de la trajectoire en se servant du B-spline d'ordre 5 et d'une paramétrisation de cette trajectoire.

Dans la troisième étape, Nous avons établi les relations théoriques nécessaires pour le calcul du gradient de la fonction du coût. Ce gradient permet de trouver de façon efficace le minimum de ce dernier.

Dans la dernière étape, nous avons procédé à la détermination de la trajectoire optimale en utilisant la fonction MATLAB 'FMINCON' guidée par le gradient de la fonction objectif.

## **CONCLUSION GÉNÉRALE**

## **Conclusion générale :**

Dans ce travail, nous avons traité le problème de planification de trajectoire optimale pour un robot à deux degrés de liberté. Nous avons utilisé le B-spline pour paramétrer la trajectoire. Cette dernière est écrite sous forme d'une combinaison linéaire des fonctions de base du B-spline. Le B-spline garantit l'obtention d'une trajectoire lisse d'ordre désiré. Nous avons considéré un critère de coût physique qui compte pour l'effort des actionneurs, la maximisation du poids solvable par le robot et la minimisation de surcharge. La méthode prend aussi en considération d'autres contraintes à savoir les limites sur les variables articulaires et sur les vitesses articulaires ainsi que les dynamiques du robot. Nous avons calculé analytiquement l'expression du gradient de la fonction du coût par rapport aux paramètres. Ce gradient est précieux pour guider l'optimisation et la rendre fiable.

Les résultats de simulation, montrent l'efficacité de la méthode. La trajectoire optimale obtenue satisfait toutes les contraintes et minimise le coût. Nous avons constaté que le coût que nous avons choisi, favorise les configurations singulières pour lesquelles la charge est directement portée par la structure du robot et pas entièrement à travers les actionneurs. Ceci minimise évidemment l'effort des actionneurs et maximise le poids solvable.

Le critère que nous avons considéré peut être étendu pour prendre en considération l'évitement d'obstacle en rajoutant un terme qui compte pour cette performance pourvu qu'on puisse calculer le gradient de ce dernier de façon efficace.

### Référence bibliographique :

- ❖ [1] **Benmisra. A**, 'Programmation des robots industriel et application sur le robot manipulateur', Thèse de Magistère, Université de Saad Dahleb de Blida (Algérie), 2007
- ❖ [2] **Wisama. K, Dombre. E**, 'Modélisation Identification et Commande des robots', 2<sup>ème</sup> édition, Hermes Science Publications, Paris, 1999.
- ❖ [3] **Isma Akli**, 'Elaboration d'une stratégie de coordination de mouvements pour un manipulateur mobile redondant', Thèse de Magister, USTHB, Algérie, 2007.
- ❖ [4] **Wisama. K**, 'Commande des robots manipulateurs', Hermes Science Publications, Paris ,2002.
- ❖ [5] **John J. Craig**, 'Introduction to Robotics, Mechanics and Control', 2nd ed, Addison-Wesley PUBLISHING Company, Canada, 1989.
- ❖ [6] **Mark W. Spong and M. Vidyasagar**, ' Robot Dynamics and Control', QuinnWoodbine, USA, 1981.
- ❖ [7] **Vibet. C**, 'Robots principes et contrôle'. Edition Marketing, juillet 1987
- ❖ [8] **Azoui. N**, 'commande non linéaire d'un bras manipulateur', mémoire de magister en électrotechnique option robotique, Université de Batna, 2008/2009 .
- ❖ [9] **Benaissa. I, Hamache. A**, 'Commande en position-orientation d'un Robot Manipulateur dans l'espace opérationnel', Mémoire d'ingénieur, Ecole nationale polytechnique, Algérie, 2004.
- ❖ [10] **Dombre. E**, 'Analyse et Modélisation des robots manipulateurs', Hermes Science Publications, Paris, 2001.
- ❖ [11] **Ignacio Herrera Aguilar**, 'Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service', Thèse l'université de Toulouse III, septembre 2007.
- ❖ [12] **Boussiala. N**, « méthodes numériques pour la résolutions des problèmes de commande optimale via les fonctions à blocs d'impulsions BPFS », Thèse de doctorat en électronique, université de constantine, décembre 2008.
- ❖ [13] **Emmanuel Trélat**, 'commande optimale', Note du cours, 2007/2008.
- ❖ [14] **Roux Patrick**, ' Nombre aléatoires et optimisation non linéaire', Mémoire d'ingénieur, école polytechnique fédérale de Lausanne, 2003/2004.
- ❖ [15] **La documentation du logiciel « MATLAB »**.