

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 À TEMPS  
OPTIMAL D'UN ROBOT MANIPULATEUR À  
TROIS DEGRÉS DE LIBERTÉS**

**Réalisé par :**

Mr : BAZEM Bachir

**Encadré par :**

Mr :Haddar hocine

**Promotion:2012/ 2013**

Année: 2012/2013

# Remerciements

*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 mon 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.*

*Je tenons à remercier vivement tous ceux qui nous ont aidés de près ou de loin à accomplir ce travail.*

*Je 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 mais amis .....*

*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 mon dieu*
-  *A ma très chère Mère et à mon cher Père mouloud;*
-  *A mon promoteur Haddar. H ansai les membres de jures*
-  *A mes frères Hakim, Aissa et mes sœurs ;*
-  *A mon cher Amis Nabil et farid ansai mes autres Amis de ma promotion*
-  *A tous mes amis surtout: mis copines de chambre Khaled fannouche et abdesslam ,hay , abdesslam arrour,abdelali , idriss, djemal, Bilal, rafik, mourad, Sami , Bachir, Faycel,...*
-  *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*

*Bachir*



## SOMMAIRE

Liste des notations

Liste des abréviations

Liste des figures

### Chapitre I

#### INTRODUCTION GENERAL

<b>I. NOTION FONDAMENTALE SUR LA ROBOTIQUE.....</b>	<b>1</b>
<b>I.1 GENERALITES .....</b>	<b>1</b>
I.1.1 Introduction .....	1
I.1.2 Historique .....	1
I.1.3 Définitions générales .....	1
I.1.4 Schéma synoptique d'un robot manipulateur .....	2
I.1.5 Classification des robots .....	3
I.1.6 Constituants d'un robot .....	3
I.1.7 Composantes d'une cellule robotisée .....	6
I.1.8 Caractéristiques d'un robot .....	7
<b>I.2. MATRICES DE TRANSFORMATION ENTRE VECTEURS REPERES .....</b>	<b>8</b>
I.2.1 Introduction .....	8
I.2.2 Coordonnées homogènes .....	9
I.2.3 Représentation d'un point dans l'espace .....	9
I.2.4 Représentation d'une direction .....	9
I.2.5 Représentation d'un plan .....	10
I.2.6 Transformation homogène .....	10
I.2.7 Transformation des repères .....	11
I.2.8 Matrice de translation .....	11
<b>I.3 MODULATION D'UN BRAS MANIPULATEUR .....</b>	<b>12</b>
I.3.1 Modélisation des robots manipulateurs .....	12
I.3.2 Modèle géométrique .....	12
I.3.2.1 Modèle géométrique direct .....	13
I.3.2.2 Le modèle géométrique inverse .....	13
I.3.3 Description géométrique (Paramètres de Denavit-Hartenberg) .....	13

I.3.4	Modélisation cinématique.....	15
I.3.4.1	Modèle cinématique directe.....	15
I.3.4.2	Modèle cinématique inverse .....	16
I.3.5	Modélisation dynamique .....	16
I.3.5.1	Modélisation dynamique directe .....	16
I.3.5.2	Modélisation dynamique inverse .....	17
I.3.6	Le formalisme de Lagrange .....	17
I.3.7	La fonction polynomiale cubique .....	17
I.3.8	Polynôme de degré cinq .....	18
I.3.9	Commande opérationnelle .....	19
<b>I.4</b>	<b>GENERATION DE TRAJECTOIRES .....</b>	<b>20</b>
I.4.1	Introduction .....	20
I.4.2	Trajectoire .....	20
I.4.3	Génération de la trajectoire dans l'espace articulaire .....	20
<b>I.5</b>	<b>CONCLUSION .....</b>	<b>21</b>
<b>Chapitre II</b>		
<b>II.</b>	<b>LA COMMANDE OPTIMALE ET L'OPTIMISATION.....</b>	<b>22</b>
II.1	Introduction .....	22
II.2	Problème de la commande optimale .....	22
II.2.1	Rapport de la commande optimale.....	23
II.2.2	Contrôlabilité.....	24
II.2.3	Existence de trajectoires optimales .....	24
II.3	Conditions nécessaires d'optimalité .....	25
II.3.1	Equations d'Euler-Lagrange .....	25
II.3.2	Principe du minimum de pontryagin .....	25
II.4	Méthode de résolutions du problème de commande optimale .....	26
II.4.1	Méthodes indirectes .....	26
II.4.2	Méthodes directes .....	27
II.5	Méthodes générales d'optimisation .....	29
II.5.1	Définition de l'optimisation .....	30
II.5.2	Optimisation locale .....	30
II.5.3	Optimisation globale .....	31
II.5.4	Optimisation stochastique .....	32
II.5.4.1	Méthode adaptative .....	32

II.5.6 Considération du gradient .....	33
II.6 Conclusion .....	34
<b>Chapitre III</b>	
<b>III. PLANIFICATION DE TRAJECTOIRES A TEMPS OPTIMAL .....</b>	<b>35</b>
III.1.Introduction .....	35
III.2. Formulation du problème de commande à temps optimal .....	35
III.3. Solution du problème de la commande à temps optimal .....	39
III.4. Résultats de simulations .....	42
III.4.1 Cinématique du robot .....	43
III.4.2 Dynamiques du robot .....	45
III.4.3. Résultat pour la première trajectoire .....	48
III.4.4. Résultat pour la deuxième trajectoire .....	52
III.4.5. Résultats pour le troisième chemin considéré .....	55
III.5. Conclusion .....	59
<b>CONCLUSION GENERAL</b>	

## Liste des notations

$m(\text{chap I})$	Nombre de degrés de liberté
	Articulation rotoïde
	Articulation prismatique
$(P, P, P)$	Coordonnées cartésiennes
$p$	Facteur d'échelle
$T$	Matrice de transformation
$(, , )$	Matrice de translation pure
$(, )$	Matrice de rotation pure
$(, )$	Variation opérationnelles
	Matrice jacobienne
	Force
	Masse
	Vitesse angulaire
$c$	Tenseur d'inertie
	Lagrange
$T$	Energie cinétique totale
	Energie potentielle totale
	Couple
$(q \text{ ou } \theta)$	Position articulaire
$(\dot{q} \text{ ou } \dot{\theta})$	Vitesse articulaire
$(\ddot{q} \text{ ou } \ddot{\theta})$	Accélération articulaire
$p$	Position initial
$p$	Position final
	Temps
$P$	Vecteur de points de contrôle
	Le couple
	Position articulaire
$\cdot$	Vitesse articulaire
$\ddot{\cdot}$	Accélération articulaire
	vecteur position
$\cdot$	Vecteur de vitesse
$t$	Temps final
$r$	Position du centre de masse
$\dot{r}$	Vitesse du centre de masse
$\cdot$	Le couple minimal
$\cdot$	Le couple maximal

# 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>Figure I-1-</b>	Schéma synoptique d'un robot manipulateur .....	2
<b>Figure I-2-</b>	Un robot manipulateur .....	4
<b>Figure I-3-</b>	Représentation articulation rotoïde .....	5
<b>Figure I-4-</b>	Représentation articulation prismatique .....	5
<b>Figure I-5-</b>	Représentation d'un point dans l'espace.....	9
<b>Figure I-6-</b>	Matrice de transformation ${}^i T_j$ de repère $R_i$ , au repère $R_j$ .....	10
<b>Figure I-7-</b>	Translation pure.....	11
<b>Figure I-8-</b>	Rotation autour de $\theta$ l'axe x.....	11
<b>Figure I-9-</b>	Paramètres géométriques dans le cas d'une structure ouverte.....	15
<b>Figure I-10-</b>	Orientation des référentiels associés à un système articulé.....	15
<b>Figure III-1-</b>	Le mouvement du robot à 3 ddl.....	36
<b>Figure III-2-</b>	Courbe de commutation de vitesse dans le plan de phase $(\theta, \dot{\theta})$ .....	41
<b>Figure III-3-</b>	Structure du robot à 3ddl.....	44
<b>Figure III-4-</b>	La première trajectoire considérée.....	49
<b>Figure III-5-</b>	Courbe de vitesse maximale autorisée accompagnée de la courbe obtenue par intégration vers l'avant de $\ddot{\theta} = (\ddot{\theta}_1, \ddot{\theta}_2)$ et de la courbe obtenue par intégration de $\ddot{\theta} = (\ddot{\theta}_1, \ddot{\theta}_2)$ .....	50
<b>Figure III-6-</b>	Les itérations effectuées pour chercher les tangentes à la courbe de vitesse maximale autorisée.....	51
<b>Figure III-7-</b>	Profile optimal de vitesse avec la courbe de vitesse maximale autorisée pour garder l'outil sur le chemin.....	52
<b>Figure III-8-</b>	Les couples articulaires correspondants à la solution à temps optimal .....	53
<b>Figure III-9-</b>	Chemin de l'outil pour le deuxième exemple.....	54
<b>Figure III-10-</b>	Profile optimal de vitesse et la courbe de vitesse maximale autorisée correspondant au chemin de la figure -9-.....	55
<b>Figure III-11-</b>	Couples articulaires correspondants au chemin du deuxième exemple.....	55
<b>Figure III-12-</b>	Troisième chemin considéré.....	56
<b>Figure III-13-</b>	Courbe de vitesse maximale autorisée .....	58
<b>Figure III-14-</b>	Profile optimal de vitesse pour la troisième chemin considéré avec la courbe de vitesse maximale autorisée et les courbes obtenues par intégration vers l'avant et vers l'arrière de $\ddot{\theta} = (\ddot{\theta}_1, \ddot{\theta}_2)$ et de $\ddot{\theta} = (\ddot{\theta}_1, \ddot{\theta}_2)$ respectivement.....	59
<b>Figure III-15-</b>	Les couples articulaires optimaux obtenus pour le troisième chemin considéré. ....	60

# Introduction générale

---

Le domaine de la robotique a fait l'objet d'un développement fulgurant. Par sa nature pluridisciplinaire, le domaine implique diverses branches : L'électronique, la mécanique, les mathématiques et l'automatique. L'application des robots ne cessera d'affirmer sa large polyvalence. Sitôt après avoir été appliqués dans le domaine industriel, ils ont vite investi d'autres domaines, passant ainsi à la médecine, à la recherche spatiale, sur les sites radioactifs (notamment pour la prévention et la protection) dans l'entretien, le secours, l'agriculture, la réhabilitation, la construction, les applications sous marines, l'exploitation minière, etc. [1].

Pour effectuer leurs tâches programmées, les robots manipulateurs s'appuient sur une structure mécanique articulée (robots manipulateurs) ou avec locomotion (robots mobiles, robots marcheurs). La génération et le contrôle de mouvement constitue, ainsi, une grande partie du domaine de la robotique [1,2]. Elle consiste à planifier un mouvement de corps solides (les segments du robot) à partir d'une position initiale à une position finale à travers une collection d'obstacle statiques, ce problème de planification géométrique d'un chemin est coûteux de point de vu calcul. Une extension de cette formulation prend en considération des problèmes supplémentaires qui sont hérités des limitations mécaniques, dynamiques et des incertitudes. Parmi ces limitations on peut citer les limitations sur les variables articulaires et sur leur première et seconde dérivée, les limitations sur les couples que peut générer les actionneurs ainsi que les équations dynamiques elles mêmes du robot. Ceux-ci compliquent plus le développement de planificateurs automatiques. La solution n'étant pas unique, on procède généralement à choisir la plus adaptée (optimale) selon un critère donné.

Dans ce travail, nous avons fait l'étude et la programmation d'un algorithme de planification de trajectoire à temps optimale. Le chemin géométrique de l'outil est spécifié au départ et on procède au calcul de l'évolution dans le temps de l'outil sur ce chemin. L'idée de base sur laquelle est fondé cet algorithme est que pour une configuration donnée il y a toujours une vitesse limite maximale à partir de laquelle les actionneurs seront incapables de maintenir l'outil sur la trajectoire spécifiée, c'est-à-dire seront incapables de générer les couples nécessaires. Alors en se basant sur cette idée, on procède au calcul de l'accélération nécessaire pour chaque position, ensuite on utilise le profile obtenu pour le calcul de la trajectoire.

Nous avons appliqué cet algorithme à un robot à trois degrés de liberté et un ensemble de chemins typiques de l'outil. Les résultats obtenus montrent la simplicité de la méthode et conduisent à une trajectoire à temps optimale.

Notre présente étude commence par livrer des généralités et des définitions spécifiques au domaine de la robotique. Ensuite nous donnons une brève présentation de la cinématique et de la dynamique des robots manipulateurs, ainsi que l'explication des diverses méthodes de génération de la trajectoire.

Nous enchaînons au deuxième chapitre par un rappel sur la commande optimale en général et sur l'optimisation paramétrique.

Au dernier chapitre, sont présentés les différents calculs effectués pour appliquer cette commande à un exemple (robot à trois degrés de liberté) ainsi que les résultats de simulation.

Enfin, nous clôturons ce mémoire avec une conclusion générale.

## I.1 GENERALITES

### I.1.1 Introduction

Ce document traite des robots manipulateurs que l'on retrouve principalement dans des applications industrielles. Ces robots sont habituellement constitués d'une série de segments relativement rigides dont les articulations sont commandées par des actionneurs. Le dernier segment est muni d'un effecteur (pince, main articulée, outil,...) servant à la réalisation de la tâche à accomplir. La conception ou l'utilisation de robots manipulateurs peut faire appel à plusieurs disciplines du génie : la mécanique, l'électricité, l'hydraulique, la pneumatique, l'électronique, l'informatique, la commande, la vision par ordinateur, etc.

Dans ce chapitre, nous allons principalement nous intéresser à la géométrie, la cinématique, la dynamique et la commande des robots manipulateurs.

### I.1.2 Historique

Les premiers programmes de recherche ont été lancés durant les années 1940 pour développer des robots de type maître-esclave pouvant manipuler les charges radio actives. Durant les années 1950 le couplage mécanique entre l'opérateur humain et le bras manipulateur a été remplacé par la partie électrique et hydraulique

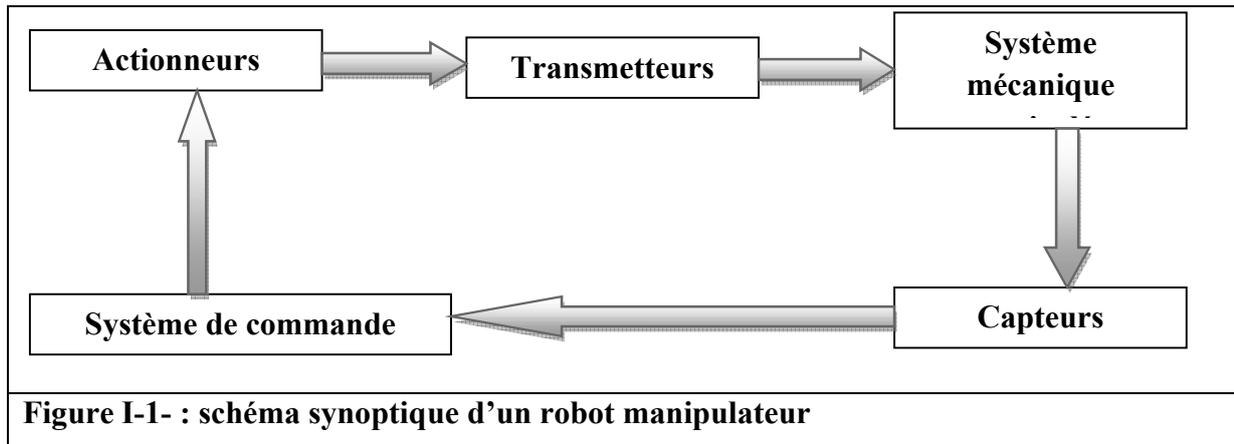
### I.1.3 Définitions générales

#### I.1.3 .a Définition du robot

Un robot est un système mécanique articulé commandé en position, reprogrammable polyvalent (à usages multiples) à plusieurs degrés de liberté. Capable de manipuler les matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches.

Ces machines polyvalentes sont généralement étudiées pour effectuer de la même façon des fonctions cycliques et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel. [1]. [2]

## I.1.4 Schéma synoptique d'un robot manipulateur



- **Système mécanique articulé :**

Le système mécanique articulé c'est la partie qui exécute les taches du robot

- **Actionneurs :**

Produisent l'effort nécessaire à transmettre aux différentes articulations du robot

- **Transmetteurs**

Assurent le transfert d'effort l'adaptation du mouvement des actionneurs du robot

- **Capteurs**

Dits proprioceptifs lorsqu'ils mesurent l'état interne du robot

(Positions et vitesses des articulations) et extéroceptifs lorsqu'ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle).

- **Système de commande**

Synthétise les consignes des asservissements pilotant les actionneurs, à partir de la fonction de perception et des ordres de l'utilisateur.

S'ajoutent à cela :

- L'interface homme-machine à travers la quelle l'utilisateur programme les tâches que le robot doit exécuter,

- Le poste de travail, ou l'environnement dans lequel évolue le robot.

## I.1.5 Classification des robots

On retiendra pour notre part 3 types de robot :

### **Les manipulateurs :**

- Les trajectoires sont non quelconques dans l'espace,
- Les positions sont discrètes avec 2 ou 3 valeurs par axe,
- La commande est séquentielle.

### **Les télémanipulateurs :**

Appareils de manipulation à distance (pelle mécanique, pont roulant), apparus vers 1945 aux USA :

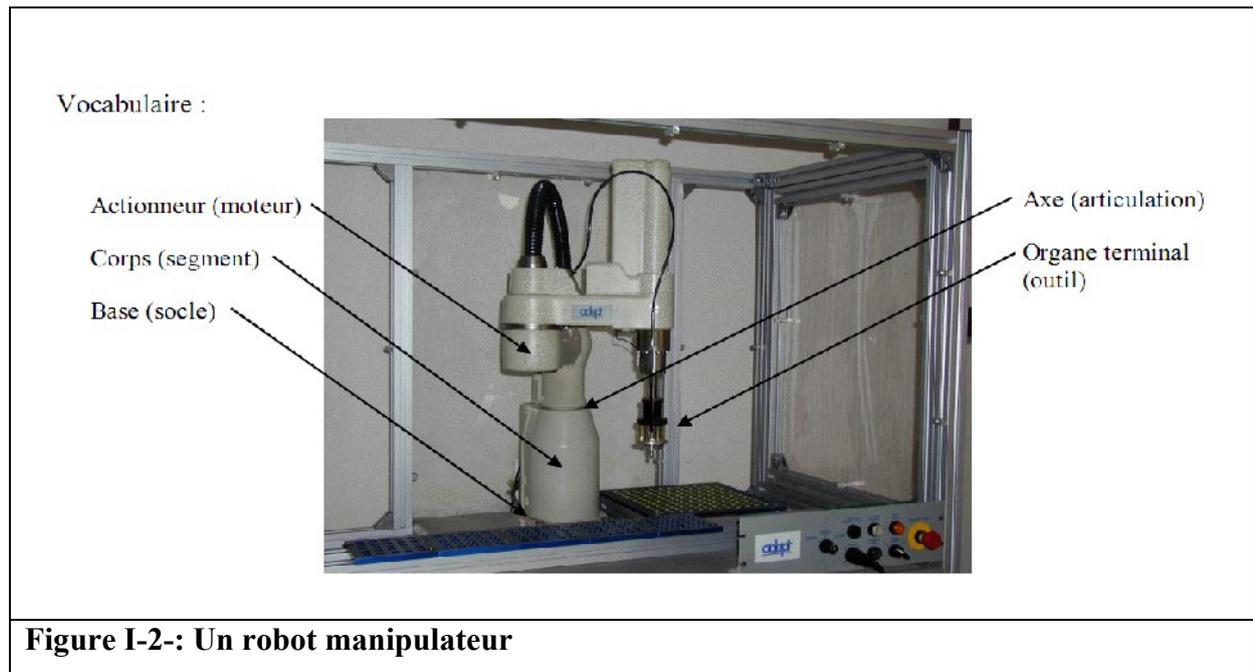
- Les trajectoires peuvent être quelconques dans l'espace,
- Les trajectoires sont définies de manière instantanée par l'opérateur, généralement à partir d'un pupitre de commande (joystick).

### **Les robots :**

- Les trajectoires peuvent être quelconques dans l'espace,
- L'exécution est automatique,
- Les informations extéroceptives peuvent modifier le comportement du robot.

## I.1.6 Constituants d'un robot

Un robot manipulateur est constitué d'un système mécanique articulé et d'un ensemble d'organes associés.



## a. Actionneurs

Les actionneurs utilisent fréquemment des moteurs électriques à aimant permanent, à courant continu, à commande par l'induit (la tension n'est continue qu'en moyenne car en général l'alimentation est un hacheur de tension à fréquence élevée ; bien souvent la vitesse de régime élevée du moteur fait qu'il est suivi d'un réducteur, ce qui permet d'amplifier le couple moteur). On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas. Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent hydrauliques, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique).

## b. L'organe terminal (effecteur)

## d. Articulations

Une articulation lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre. Soit  $m$  le nombre de degré de liberté résultant, encore appelé mobilité de l'articulation. La mobilité d'une articulation est telle que :

$$0 \leq m \leq 6$$

Lorsque  $m = 1$  ; ce qui est fréquemment le cas en robotique, l'articulation est dite simple : soit rotoïde, soit prismatique. [1]

## d.1 Articulation rotoïde

Est une articulation de type pivot réduisant le mouvement entre les 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 de cet axe, et est représentée par le symbole **R**.

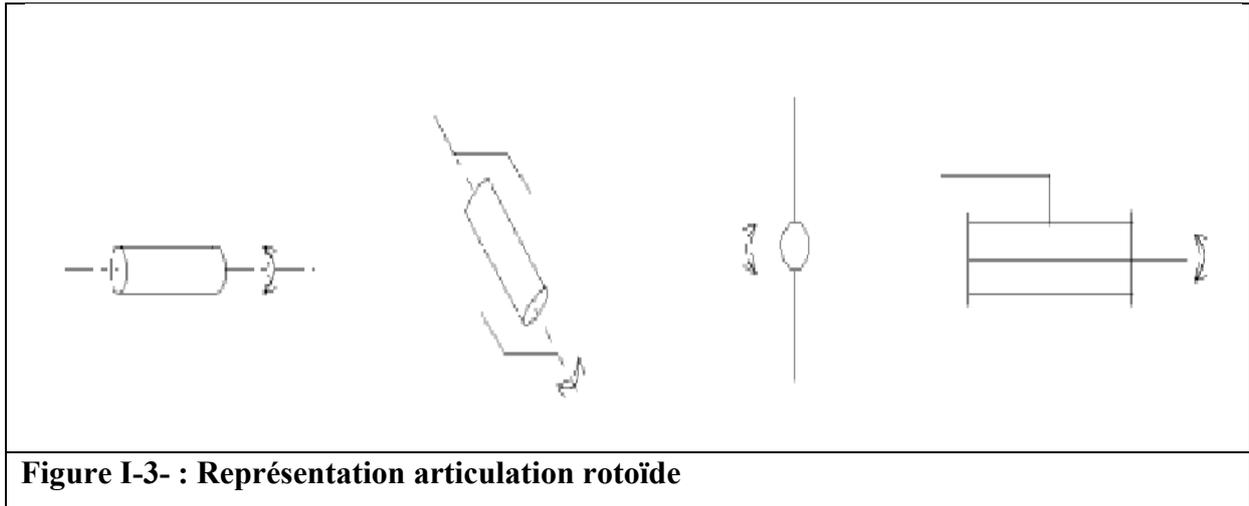


Figure I-3- : Représentation articulation rotoïde

## d.2 Articulation prismatique

Est une articulation de type glissière réduisant le mouvement entre les deux corps à une translation le long d'un axe commun

La situation relative entre les deux corps est mesurée par distance le long de cet axe, et représentée par le symbole **P**.

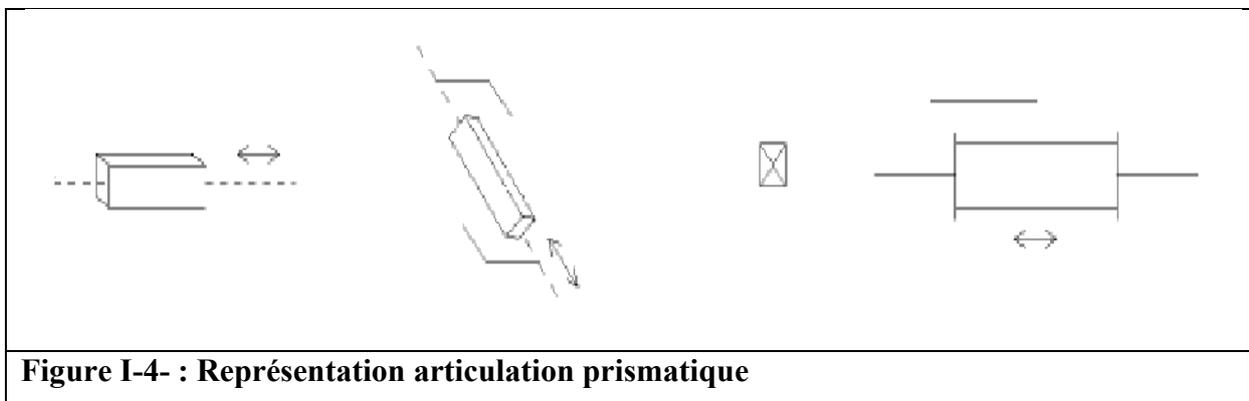


Figure I-4- : Représentation articulation prismatique

## e. Élément porteur

Sa motorisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés.

## f. Espace opérationnels

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).

## g. 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 présences d'obstacles.

### I.1.7 Composantes d'une cellule robotisée

#### a. 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 ;

#### b. 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 ;

## d. 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.

## c. L'interface homme-machine :

A travers laquelle l'utilisateur programme la tâche que le robot doit exécuter.

## d. Configurations singulières

Pour tous les robots, qu'ils soient redondants ou non, il se peut que dans certaines configurations dites singulières, le nombre de degrés de liberté nécessaire à l'organe terminal soit inférieur à la dimension de l'espace opérationnel. [12]

La configuration singulière ou singularités se traduit, physiquement, par la nullité du déterminant de la matrice Jacobéenne. [12]

## e. Choix du nombre degrés de liberté d'un robot

Soit  $R_s$  un repère attaché à un corps rigide (S). Le nombre de degrés de liberté maximum de ce corps est le nombre de paramètres indépendants nécessaires pour déterminer complètement à tout instant, la transformation géométrique  $T$  permettant le passage du repère de référence  $R_o$  au repère  $R_s$  attaché à l'objet.

Soit un manipulateur à  $n$  segments rigides reliés par des articulations. Le nombre de degrés de liberté du porteur est le nombre de paramètres indépendants nécessaires pour exprimer la transformation  $T$  faisant passer du repère de base ( $R_o$ ) du manipulateur au repère du dernier segment ( $R_{n-1}$ ) sur lequel est fixé, l'outil:  $T: R_o \rightarrow R_{n-1}$

## I.1.8 Caractéristiques d'un robot

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Voici quelques paramètres à prendre, éventuellement, en compte :

- *L'espace de travail* : défini comme l'ensemble des points atteignables par l'organe terminal. Tous les mouvements ne sont pas possibles en tout point du volume de travail. L'espace de travail également appelé espace de travail maximal, est le volume de l'espace que le robot peut atteindre *via* au moins une orientation de l'organe terminal. L'espace de travail dextre est le volume de l'espace que le robot peut

atteindre avec toutes les orientations possibles de l'organe terminal. Cet espace de travail est un sous-ensemble de l'espace de travail maximal.

- *La charge utile* : transportable (de quelques kilos à quelques tonnes), à déterminer dans les conditions les plus défavorables (en élongation maximum).
- *L'architecteur du S M A* : (structure mécanique articulée) le choix est guidé par la tâche à réaliser
- *Les vitesses et accélérations maximales* : conditionnent les temps de cycle.
- *L'erreur absolue de positionnement* : correspondant à l'erreur entre le point souhaité (réel) et le point atteint, elle est calculée via le modèle géométrique inverse du robot cette erreur est due au modèle utilisé à la quantification de la mesure de position à la flexibilité du système mécanique. En général, l'erreur absolue de positionnement, également appelée précision, est d'ordre de 1 mm
- *Répétabilité justesse* la répétabilité est une mesure de la capacité du robot de pouvoir retourner se positionner au même point de façon répétitive.

## I.2. MATRICES DE TRANSFORMATION ENTRE VECTEURS REPERES

### I.2.1 Introduction

En robotique, on associe à tout élément du poste de travail un ou plusieurs repères.

Ces repères sont généralement définis de telle sorte que leurs axes et leurs origines correspondent à des directions et à des points privilégiés ayant un rôle fonctionnel lors de l'exécution de tâches :

De direction ou d'insertion ou centre de gravité d'une pièce par exemple, mais aussi axe d'une articulation ou extrémité d'un effecteur. Ils permettent de situer dans l'espace les objets fixes de l'environnement (distributeurs, dispositif de bridage....) ainsi que le corps mobiles constitutif du robot ou transportés par lui. [1]

La notion de transformation de repère est fondamentale, elle nous permet de :

- Exprimer les situations des différents corps du robot les uns par rapport aux autres.
- Spécifier les situations que doit prendre le repère associé à l'organe terminal du robot pour réaliser une tâche donnée.
- Contrôler et écrire les efforts mis en jeu lorsque le robot interagit avec son environnement ces transformations consistent principalement en translation et rotation.

Dans cette partie nous représenterons une relation qui nous permet de décrire de façon homogène les différents systèmes des coordonnées.

## I.2.2 Coordonnées homogènes

Les différentes transformations géométriques nécessitent l'utilisation de différents opérateurs matriciels :

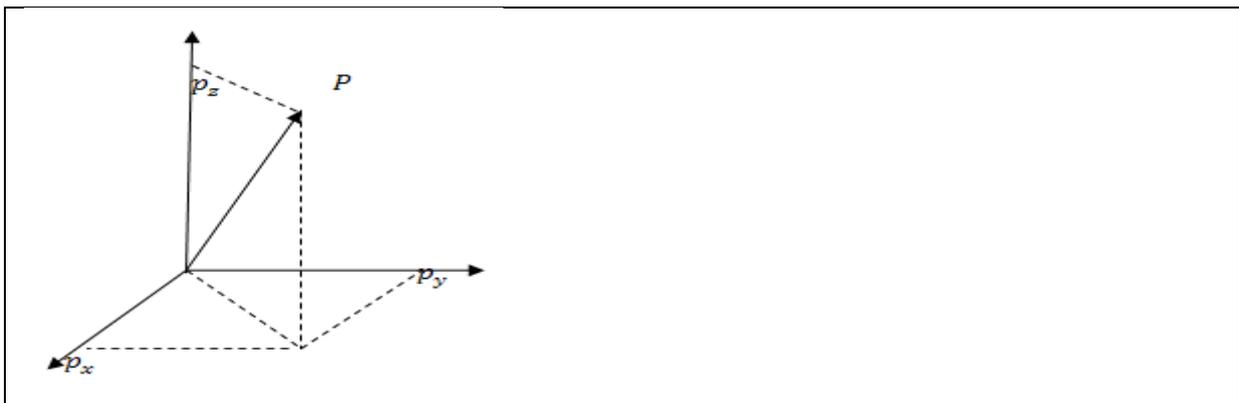
- l'addition et la soustraction pour les translations ;
- la multiplication matricielle pour les autres opérations (changement d'échelle, rotation, etc)

L'utilisation des coordonnées homogènes permet de réaliser toutes ces opérations à l'aide d'un seul opérateur : la multiplication. Ceci facilite le développement d'opérations composées et le développement du modèle géométrique d'un robot manipulateur.

## I.2.3 Représentation d'un point dans l'espace

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

$$\begin{bmatrix} w \cdot p_x \\ w \cdot p_y \\ w \cdot p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad (I.5)$$



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

## I.2.4 Représentation d'une direction

La représentation d'une direction (vecteur libre) se fait aussi par quatre composantes, mais la quatrième est nulle, indiquant 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 :

=

0

### I.2.5 Représentation d'un plan

On représente le plan par l'équation  $ax + by + cz + d = 0$  par la vectrice ligne :

=

Pour tout  $p$  situé dans le plan, le produit matriciel  $M \cdot p$  Est nul.

### I.2.6 Transformation des repères

Une transformation de repère  $R_i$  à un autre repère  $R_j$  est une matrice  $4 \times 4$ . cette transformation

Est définie par  ${}^i T_j$  et peut représenter une translation et/ou une rotation

$${}^i T_j = \begin{bmatrix} {}^i s_j & {}^i n_j & {}^i a_j & {}^i p_j \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Où :  ${}^i s_j$ ,  ${}^i n_j$ , et  ${}^i a_j$  représentent respectivement les vecteurs unitaires suivant les axes  $x_j$ ,  $y_j$ , et  $z_j$  du repère  $R_j$  exprimés dans le repère

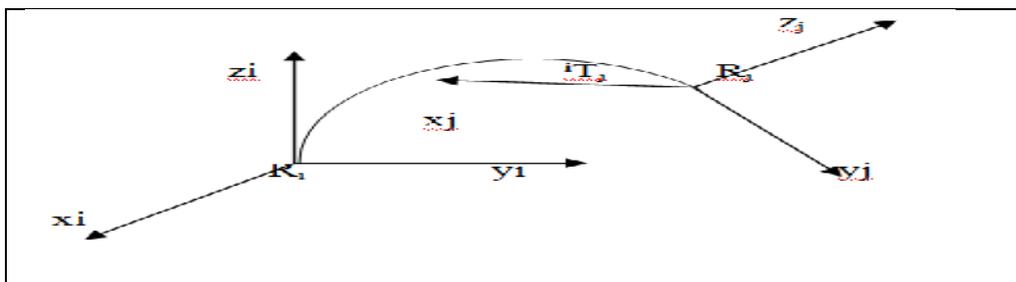


Figure I-6-: matrice de transformation  ${}^i T_j$  de repère  $R_i$ , au repère  $R_j$ ,

I.2.7 Matrice de translation

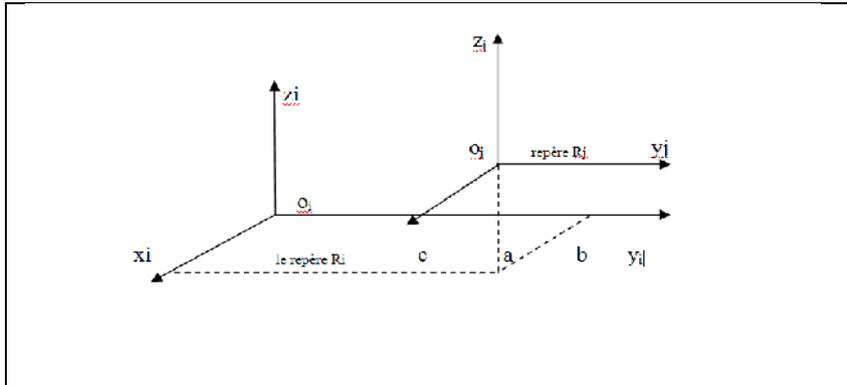


Figure I-6:- translation pure.

Soit de repère  $(x_i, y_i, z_i)$   $(x_j, y_j, z_j)$  et soient  $a, b, c$ , les coordonnées de l'origine  $O_j$  du repère  $R_j$  dans le repère  $R_i$ . Lorsque les axes des deux repère  $R_i$  et  $R_j$  sont en parallèle, alors la matrice de transformation  $A$  est donnée par la matrice de translation suivante :

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, (x_j, y_j, z_j) = \begin{pmatrix} x_i + a \\ y_i + b \\ z_i + c \end{pmatrix}$$

I.2.8 Matrice de rotation

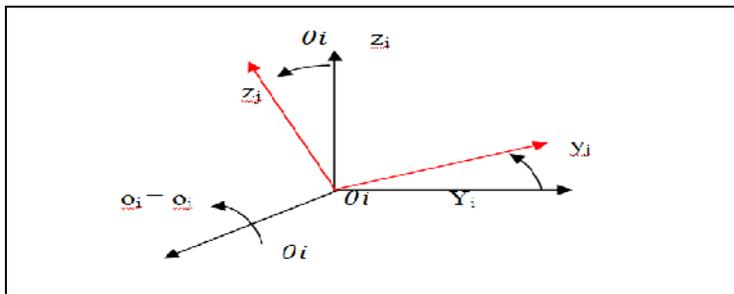


Figure I-8:- rotation autour de l'axe x.

Soient les deux repères  $R_i$  et  $R_j$  précédemment définis. Lorsque leurs origines coïncident ( $O_i = O_j$ ), et que le repère  $R_j$  effectue une rotation d'un angle  $\theta_i$  autour d'un ou plusieurs axes de  $R_i$  alors cette rotation peut être décomposée en rotations élémentaires de différents angles autour des axes de  $R_i$ . La matrice de transformation est obtenue par le produit de ces trois matrices de rotation :

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

- $$G_z(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Rotation d'un angle  $\alpha$  autour de  $Oz$  :

$$G_z(\alpha) = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Rotation d'un angle  $\alpha$  autour de  $Ox$  :

$$G_x(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

### I.3 MODELISATION D'UN BRAS MANIPULATEUR

#### I.3.1 Modélisation des robots manipulateurs

Jusqu'à présent, les robots industriels sont surtout employés pour servir comme générateurs 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 polyarticulé. 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. Le but de la modélisation dans la robotique est de fournir les équations mathématiques qui permettent de programmer, commander et exécuter les tâches désirées [4].

#### I.3.2 Modèle géométrique

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 vu de décrire ses paramètres géométriques : position et orientation [1] [3].

La variable articulaire  $\theta_i$  associée à la  $i$ -ième articulation et soit  $\theta_i$ , soit  $\dot{\theta}_i$ , selon que cette articulation est rotoïde ou prismatique, ce qui se traduit par la relation :

$$\dot{\theta}_i = \dot{\theta}_i + \theta_i \quad (I.1)$$

Avec:

- $\alpha = 0$  si l'articulation est rotoïde ;
- $\alpha = 1$  si l'articulation est prismatique ;
- $\alpha' = 1 - \alpha$ .

La matrice de transformation définissant le repère dans le repère est donnée par :

$$= (R) \times (T) \times (A) \times (O) , \tag{I.2}$$

Ce qui donne :

$$= \begin{bmatrix} \cos & -\sin & 0 & \\ \cos & \sin & \cos & \cos & -\sin & -\sin \\ & \sin & \sin & \cos & \cos & -\cos \\ 0 & & 0 & & 0 & 1 \end{bmatrix}$$

Où :  $(R, T)$  et  $(A, O)$  sont des matrices homogènes (4x4) représentant respectivement une rotation et une translation le long de  $\mathbf{u}$ .

### I.3.2.1 Modèle géométrique direct

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 [1].

Le (MGD) peut être représenté par la relation :

$$= ( ) \tag{I.3}$$

où :

- : le vecteur des coordonnées opérationnelles ;
- : le vecteur des coordonnées articulaires.

### I.3.2.2 Le modèle géométrique inverse

Le modèle géométrique inverse 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 constitue ce que l'on appelle le modèle géométrique inverse (MGI) [1] [2].

Le (MGI) peut être représenté par la relation :

$$= ( ) \tag{I.4}$$

### I.3.3 Description géométrique (Paramètres de Denavit-Hartenberg)

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**. 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  $n$  corps notés  $C_1, \dots, C_n$  et de  $n-1$  articulations.

Le corps  $C_1$  désigne la base du robot et le corps  $C_n$  est 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  $x_j$  est porté par l'axe de l'articulation  $j$  ;
- L'axe  $z_j$  est porté par la perpendiculaire commune aux axes  $x_{j-1}$  et  $x_j$ .

Si les axes  $x_{j-1}$  et  $x_j$  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  $x_{j-1}$  et  $x_j$  correspondant à une rotation autour de  $z_{j-1}$  ;
- $a_j$  : distance entre  $x_{j-1}$  et  $x_j$  le long de  $z_{j-1}$  ;
- $\beta_j$  : angle entre les axes  $z_{j-1}$  et  $z_j$  correspondant à une rotation autour de  $x_j$  ;
- $d_j$  : distance entre  $x_{j-1}$  et  $x_j$  le long de  $z_j$ .

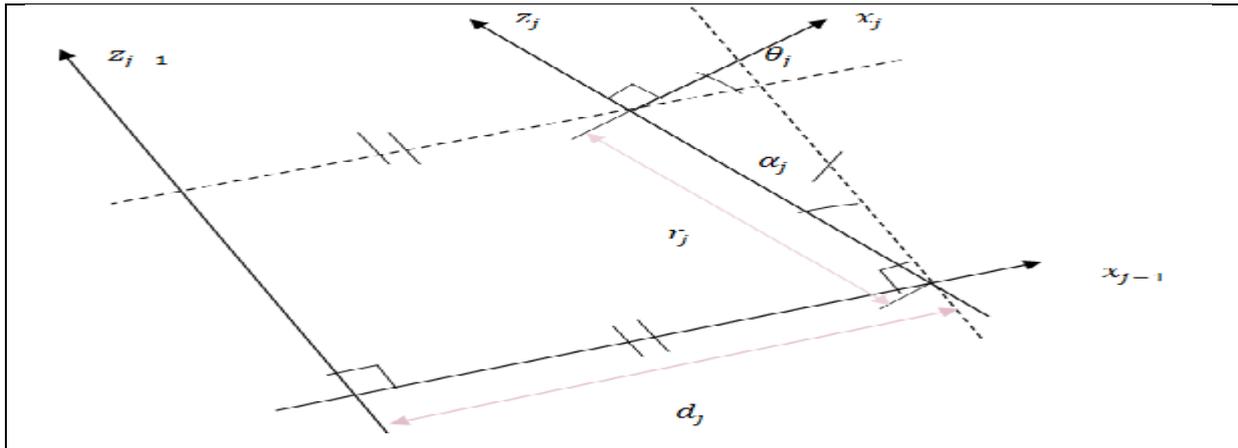


Figure I-9- : Paramètres géométriques dans le cas d’une structure ouverte simple.

I.3.4 Modélisation cinématique

Le modèle géométrique permet la commande du robot en position mais ses limites apparaissent dès qu’il est question de maîtriser la vitesse. On définit alors le modèle cinématique qui consiste en la détermination d’une relation entre les dérivées par rapport au temps des variables généralisées. Ce modèle est obtenu par linéarisation du modèle géométrique directe [2] [9].

I.3.4.1 Modèle cinématique directe

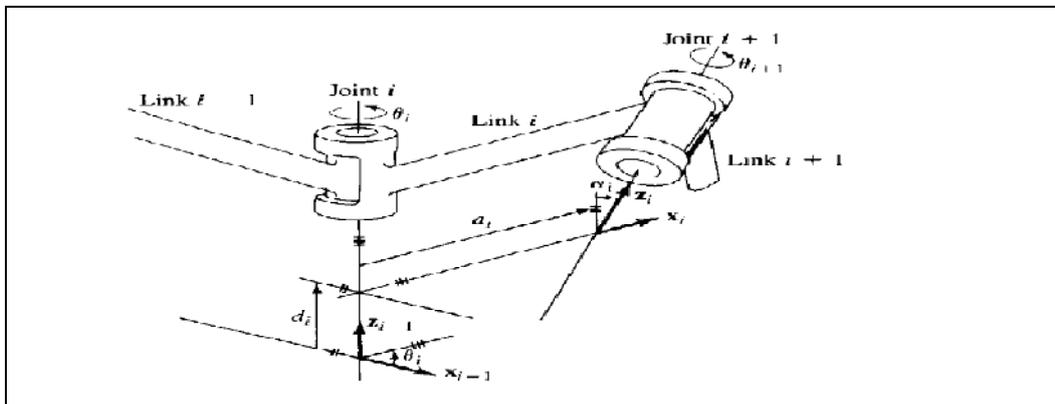


Figure I-10- : Orientation des référentiels associés à un système articulé.

Le modèle cinématique directe (MCD) d’un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires [1] [2]. Il est noté :

$$\dot{x} = J(\theta) \dot{\theta} \tag{I.5}$$

où :

$J(\theta)$  : désigne la matrice jacobienne de dimension  $(m \times n)$  du mécanisme, égale à  $\frac{\partial x}{\partial \theta}$  et fonction de la configuration articulaire  $\theta$ .

La même matrice jacobienne intervient dans le calcul du modèle différentiel direct qui donne les variations élémentaires des coordonnées opérationnelles en fonction des variations , soit :

$$\dot{q} = J \dot{p} \quad (I.6)$$

### I.3.4.2 Modèle cinématique inverse

L'objectif du modèle cinématique inverse (MCI) est de calculer, à partir d'une configuration donnée, les vitesses articulaires  $\dot{q}$  qui assurent au repère terminal une vitesse opérationnelle  $\dot{p}$  imposée. Cette définition est analogue à celle du modèle différentielle inverse : ce dernier permet de déterminer la différentielle articulaire correspondante à une différentielle des coordonnées opérationnelles spécifiées. Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires [1].

Dans le cas régulier la matrice Jacobienne est carrée et son déterminant est non nul, et la relation qui donne le modèle géométrique inverse est :

$$\dot{q} = J^{-1} \dot{p} \quad (I.7)$$

### I.3.5 Modélisation dynamique

Le modèle dynamique est la relation entre les couples (et/ou les forces) appliqués aux actionneurs et les positions, vitesses et les accélérations articulaires [4].

#### I.5.1 Modèle dynamique direct

Le modèle dynamique direct est celui qui exprime les accélérations articulaires en fonction des positions, vitesses et couples des articulations.

Il est alors représenté par la relation :

$$\ddot{q} = g(q, \dot{q}; \Gamma, \dots) \quad (I.8)$$

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 ;

- $\ddot{\theta}$ : vecteur des accélérations articulaires ;
- $F$  : vecteur représentant l'effort extérieur (forces et moment) qu'exerce le robot sur l'environnement.

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.3.5.2 Modèle dynamique inverse

On convient d'appeler modèle dynamique inverse, ou tout simplement modèle dynamique, la relation de la forme :

$$F = M(\theta) \ddot{\theta} + C(\theta, \dot{\theta}) \dot{\theta} + G(\theta) \quad (I.9)$$

Le modèle dynamique inverse est utilisé surtout pour la commande.

Pour obtenir le modèle dynamique, il existe plusieurs formalismes ; les plus utilisés sont le formalisme de Newton-Euler et le formalisme de Lagrange.

### I.3.6 Le formalisme de Lagrange

Le formalisme de 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 :

$$F = - \frac{\partial L}{\partial \theta} - \frac{\partial L}{\partial \dot{\theta}} \dot{\theta} \quad (I.10)$$

avec :

- L : lagrangien du système égale à E-U
- E : énergie cinétique totale du système
- U : énergie potentielle totale du système

### I.3.7 La fonction polynomiale cubique

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 :

$$\theta(t) = a_0 + a_1.t + a_2.t^2 + a_3.t^3 \quad (I.11)$$

Ainsi on obtient la vitesse articulaire :

$$\dot{\theta}(t) = a_1 + 2.a_2.t + 3.a_3.t^2 \quad (I.12)$$

Et aussi l'accélération articulaire :

$$\ddot{\theta}(t) = 2.a_2 + 6.a_3.t \quad (I.13)$$

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 :

$$\theta(0)=\theta_0, \theta_d(t_f)=\theta_f$$

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

$$\dot{\theta}(0)=0, \dot{\theta}(t_f)=0$$

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

$$\begin{cases} \theta_0 = a_0 \\ \theta_f = a_0 + a_1.t_f + a_2.t_f^2 + a_3.t_f^3 \\ a_1 = 0 \\ 0 = a_1 + 2.a_2.t_f + 3.a_3.t_f^2 \end{cases} \quad (I.14)$$

En résolvant le système d'équations on obtient :

$$\begin{cases} a_0 = \theta_0 \\ a_1 = 0 \\ a_2 = (3/t_f^2) (\theta_f - \theta_0) \\ a_3 = -(2/t_f^3) (\theta_f - \theta_0) \end{cases} \quad (I.15)$$

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

### I.3.8 Polynôme de degré cinq

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 :

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (I.16)$$

Ainsi on obtient la vitesse articulaire :

$$\dot{\theta}(t) = a_1 + 2 a_2 t + 3 a_3 t^2 + 4 a_4 t^3 + 5 a_5 t^4 \quad (I.17)$$

Et aussi l'accélération articulaire :

$$\ddot{\theta}(t) = 2 a_2 + 6 a_3 t + 12 a_4 t^2 + 20 a_5 t^3 \quad (I.18)$$

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 :

$$\left\{ \begin{array}{l} \theta_0 = a_0 \\ \dot{\theta}_0 = a_1 \\ \ddot{\theta}_0 = 2.a_2 \\ \theta_f = a_0 + a_1.t_f + a_2.t_f^2 + a_3.t_f^3 + a_4.t_f^4 + a_5.t_f^5 \\ \dot{\theta}_f = a_1 + 2.a_2.t_f + 3.a_3.t_f^2 + 4.a_4.t_f^3 + 5.a_5.t_f^4 \\ \ddot{\theta}_f = 2.a_2 + 6.a_3.t_f + 12.a_4.t_f^2 + 20.a_5.t_f^3 \end{array} \right. \quad (I.19)$$

En résolvant le système d'équations on trouve :

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

### I.3.9 Commande opérationnelle

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 Jacobien inverse) résulte des non linéarités supplémentaires.

## I.4 GENERATION DE TRAJECTOIRES

### I.4.1 Introduction

La génération de trajectoires est une étape très importante dans la commande des robots manipulateurs, elle consiste à calculer les consignes de référence en position, en vitesse, et en accélération qui décrivent le mouvement désiré du robot.

Cependant la trajectoire est l'évolution de la position, et ses dérivées temporelles en fonction du temps pour chacune des articulations.

Le mouvement le plus simple est d'aller d'un point initial à un point final, ainsi le robot est commandé de changer sa configuration initiale vers une configuration finale.

Ce type de mouvement convient aux tâches de transfert d'objets quand l'espace de travail ne comporte aucun obstacle. Pour certaines raisons telles qu'éviter les obstacles, le chemin à suivre par l'élément terminal peut être contraint par l'addition de points intermédiaires aux configurations initial et final.

En résumé, le parcours peut être planifié de différentes manières dont on distingue :

- le mouvement entre deux points avec trajectoire libre entre les points;
- le mouvement entre deux points via des points intermédiaires avec trajectoire libre entre les points intermédiaires;
- le mouvement entre deux points avec trajectoire contrainte entre les points (trajectoire rectiligne);
- le mouvement entre deux points via des points intermédiaires avec trajectoire contrainte entre les points intermédiaires ;

Dans les deux premiers cas la génération de la trajectoire peut se faire directement dans l'espace articulaire. Quant aux deux derniers, la trajectoire étant décrite dans l'espace opérationnelle, il est préférable de raisonner dans cet espace. [3] [6] [7]

### I.4.2 Trajectoire

Un mouvement (ou trajectoire) caractérise le déplacement d'un repère au cours du temps et est défini

par un chemin  $P$  et une loi d'évolution continue  $(\cdot)$ . Une trajectoire  $(\cdot)$  représentant un déplacement de  $\cdot$  à  $\cdot$  sur le chemin  $P$  est une fonction continue du temps  $t$  tel que  $\in [ \cdot, \cdot ]$  :

### I.4.3 Génération de la trajectoire dans l'espace articulaire

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.

## I.5 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 du robot en présentant les différents modèles (modèles géométriques, modèles cinématiques, modèles dynamiques). Ensuite, on a présenté quelques notions sur la commande des robots manipulateurs et une méthode pour la génération de trajectoires : la méthode de l'interpolation polynomiale.

L'utilisation du 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.

## II.1 Introduction

Les robots manipulateurs en milieu industriel réalisent des tâches de façon répétitive à l'intérieur d'un espace sécurisé pour ces opérations. Les contraintes de performance sont généralement liées aux temps d'exécution de ces tâches, avec un maximum d'économie d'énergie. Pour surmonter ces contraintes, on a le choix entre plusieurs techniques de commande. [9]

On distingue

### ➤ 1 Commande dans l'espace libre

- Commande par articulation.
- Commande jacobienne.
- 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.2 Problème de la commande optimale :

L'objectif de la commande optimale est de déterminer une commande optimale  $u^*(t)$  ou  $u^*(\theta)$ , qui force le système à satisfaire des contraintes physiques tout en minimisant ou maximisant un indice de performance [12],[13].

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 :

$$\dot{x}(t) = f(x(t), u(t), t) \quad (\text{II.1})$$

où  $x \in \mathbb{R}^n$  est le vecteur d'état,  $u \in \mathbb{R}^m$  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.

$$(\ ) = \quad (II.2)$$

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 :

$$= \quad , \quad +f \quad ((\ ), (\ ), (\ )) \quad (II.3)$$

Où : et sont des fonctions scalaires, continûment différentiables par rapport à toutes leurs variables. Ces fonctions sont sélectionnées compte tenu des objectifs 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.2.1 Rapport de la commande optimale :

Le problème de la commande optimale sans contraintes peut être énoncé comme suit : Etant donné , , , , , et , il s'agit de trouver  $^*(\ )$  ou  $^*(\ , \ )$  qui minimise l'indice de performance :

$$= \quad , \quad +f \quad ((\ ), (\ ), (\ )) \quad (II.4)$$

Sachant que :

$$(\ ) = \quad ((\ ), (\ ), (\ )) \quad (\ ) = \quad (II.5)$$

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 para-métrisation.

En général, il n'est pas possible de résoudre analytiquement le problème (II.4), (II.5). 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 :

$$J = \int_0^T (x^T Q x + u^T R u) dt \tag{II.6}$$

Sachant que :

$$\dot{x} = A x + B u \tag{II.7}$$

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

$$u^*(t) = -R^{-1} B^T (P x(t) + p(t)) \tag{II.8}$$

Où  $P$  est la solution de l'équation de Riccati.

### II.2.2 Contrôlabilité

On veut répondre à la question suivante : étant donné le système (II.1), où peut-on aller en temps  $T$  en faisant varier le contrôle ? On rappelle tout d'abord la notion d'ensemble accessible. L'ensemble accessible en temps  $T$  pour le système (II.1), noté  $\mathcal{A}(x_0, T)$ , est l'ensemble des extrémités au temps  $T$  des solutions du système partant de  $x_0$  au temps  $t = 0$ . [13]

### II.2.3 Existence de trajectoires optimales

Au problème de contrôle s'ajoute un second problème, celui de la minimisation. Parmi toutes les solutions reliant  $x_0$  à  $x_f$ , trouver une trajectoire qui minimise la fonction coût  $J(x_0, u)$

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. [13]

### II.3 Conditions nécessaires d'optimalité

#### II.3.1 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 . [12]

Pour résoudre le problème (II.4)-(II.5), nous utiliserons les multiplicateurs de Lagrange,  $(\lambda) \in \mathbb{R}$ . Par conséquent l'indice de performance est donné par :

$$J = J(x(t), u(t)) + \int_0^T [\lambda(t) (f(x(t), u(t), t) - \dot{x}(t)) - \mu^*(t))] dt \tag{II.9}$$

Introduisons le Hamiltonien défini par :

$$H(x(t), u(t), \lambda(t), t) = J(x(t), u(t), t) + \lambda(t) (f(x(t), u(t), t) - \dot{x}(t)) - \mu^*(t))$$

Nous pouvons, ainsi, écrire (3) sous la forme :

$$J = J(x(t), u(t)) + \int_0^T [\lambda(t) (f(x(t), u(t), t) - \dot{x}(t)) - \mu^*(t))] dt \tag{II.10}$$

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

$$\int_0^T \lambda(t) (-\dot{x}(t)) dt = -\lambda(T)x(T) + \lambda(0)x(0) - \int_0^T \dot{\lambda}(t) x(t) dt \tag{II.11}$$

Cependant (4) devient :

$$J = J(x(t), u(t)) - \lambda(T)x(T) + \lambda(0)x(0) + \int_0^T [\lambda(t) (f(x(t), u(t), t) - \dot{x}(t)) - \mu^*(t)) - \dot{\lambda}(t) x(t)] dt \tag{II.12}$$

Le problème original, (II.4)-(II.5), a été converti à un problème de minimisation, (II.11), sans contraintes.

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

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial x} = -\left[ \frac{\partial J}{\partial x} + \lambda(t) \frac{\partial f}{\partial x} - \dot{\lambda}(t) \right] \tag{II.13}$$

avec la condition finale :

$$\lambda(T) = -\frac{\partial J}{\partial x(T)} \tag{II.14}$$

#### II.3.2 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 :

$$\begin{aligned} \dot{\lambda} &= - \frac{\partial H}{\partial \lambda} = \left( \dots \right) \\ \left( \lambda^*, u^*, x^*, t \right) &\leq \left( \lambda^*, u^*, x^*, t \right) \end{aligned} \tag{II.15}$$

Pour tout  $t \in [t_0, t_f]$  et pour tout  $(x, u) \in \mathbb{C} \subset \mathbb{R}^n$ . Donc la commande optimale doit minimiser le Hamiltonien défini par (II.10).

L'inégalité (II.12) 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 variatio-nnel. Il faut aussi signaler que l'approche variatio-nnel 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.

## II.4 Méthode de résolutions du problème de commande optimale

### II.4.1 Méthodes indirectes :

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

#### II.4.1.1 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.4.1.2 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.4.2 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.4.2.1 Méthodes de discrétisation :

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

$$t_0 < t_1 < t_2 < \dots < t_N = t_f \quad (\text{II.16})$$

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 :

$$X = (x_0, \dots, x_N, u_0, \dots, u_{N-1}) \quad (\text{II.17})$$

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.

$$z = (z_1, z_2, \dots, z_n) \quad (\text{II.18})$$

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

#### II.4.2.2 Méthodes de para-métrisation :

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

##### - Para-métrisation de la Commande :

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

$$u(t) = \sum_{i=1}^n \alpha_i \phi_i(t) \quad = 1, 2, \dots, n, \quad (\text{II.19})$$

où les  $\alpha_i$  sont les paramètres inconnus et les  $\phi_i(t)$  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.

##### - Para-métrisation 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 :

$$x(t) = \sum_{i=1}^n \phi_i(t) \alpha_i = 1, 2, \dots, n, \quad (\text{II.20})$$

$$u(t) = \sum_{i=1}^n \psi_i(t) \beta_i = 1, 2, \dots, n, \quad (\text{II.21})$$

où les  $\alpha_i$ ,  $\beta_i$  sont des paramètres inconnus et les  $\phi_i(t)$ ,  $\psi_i(t)$  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.

#### - Para-métrisation d'Etat :

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

$$x(t) = \sum_{i=1}^n \phi_i(t) \alpha_i = 1, 2, \dots, n, \quad (\text{II.22})$$

Les  $\alpha_i$  sont les paramètres inconnus et les  $\phi_i(t)$  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.5 Méthodes générales d'optimisation :

Maintes fois aborde 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. [14]

### II.5.1 Définition de l'optimisation :

Optimiser une fonction continue  $f : E \rightarrow \mathbb{R}$  consiste à déterminer tous les  $s \in S$  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 :

$$\Leftrightarrow \exists \delta > 0, \forall s \in S, f(s) \geq f(s_0) \text{ si } \|s - s_0\| < \delta$$

Par contre,  $s_0$  est un minimum global de  $f$  sur  $S$  si son image par  $f$  est supérieure ou égale 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.5.2 Optimisation locale :

#### II.5.2.1 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 "descente" 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.5.2.2 Algorithme génétique :

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

Lorsque l'on procède à l'élimination de certains éléments de  $S'$ , 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 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.5.3 Optimisation globale :

#### II.5.3.1 Méthode des variations locales :

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

Étant donné un pas  $\rho > 0$  et un point  $(x_1, \dots, x_n) \in S$ , on choisit, parmi les trois points  $\{(x_1 - \rho, \dots, x_n), (x_1, \dots, x_n), (x_1 + \rho, \dots, x_n)\}$ , 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.

Cette méthode possède les deux avantages suivants :

- 1) Si  $f$  est continue et  $f(x) = +\infty$  ( $f(x) = -\infty$ ), alors un point stationnaire est atteint en un nombre fini d'étapes.
- 2) Si  $f$  est différentiable et que la limite de  $f$  a la même propriété qu'en 1), alors la solution tend vers un maximum local lorsque  $\rho$  tend vers 0.

#### II.5.3.2 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.5.3.3 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 separe l'intervalle en deux parties, comme c'était le cas dans la méthode branch and bound.

### II.5.4 Optimisation stochastique :

On considère par ailleurs toujours le problème de minimum suivant :

$$\min_{x \in \mathbb{R}^n} f(x) \quad (\text{II.23})$$

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

#### II.5.4.1 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. 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 en est que puisque le point duquel part la demi-droite est un minimum local, le gradient y est nul. C'est pour cela que la direction de la demi-droite est choisie aléatoirement.

### II.5.6 Considération du gradient :

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 a 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 sûr, 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. [14]

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  $\mathbf{x}_0$  (vecteur initial) et on calcule le vecteur gradient :

$$\mathbf{g}_k = -\nabla f(\mathbf{x}_k) \text{ au point } \mathbf{x}_k \quad (\text{II.24})$$

On effectue la recherche linéaire du scalaire  $\alpha_k$  rendant  $f(\mathbf{x}_k + \alpha_k \mathbf{g}_k)$  minimum dans la direction  $\mathbf{g}_k$  ('Linear Search'), soit :

$$\alpha_k = \arg \min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{g}_k) \quad (\text{II.25})$$

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 :

$$F(x) = \frac{1}{2} x^T H x + g^T x + c \tag{II.26}$$

où

$$g = -\nabla F(x) \text{ est le gradient}; \tag{II.27}$$

et

$$H = \nabla^2 F(x) \text{ est la deuxième dérivée ou le Hessien} \tag{II.28}$$

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 :

$$\nabla F(x) = Hx + g = 0 \Rightarrow x = -H^{-1}g \tag{II.29}$$

Quand on utilise une forme quadratique pour approximer une fonction générale non linéaire, il serait logique de prendre un pas de  $h = -H^{-1}g$  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  $h = -H^{-1}g$ , 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 que l'on puisse faire sans information supplémentaire.

**II.6 Conclusion**

Dans ce chapitre nous avons présenté les différentes méthodes de commande optimale ont été rappelées brièvement. Et en suit un rappel sur la commande optimale avec les diverses

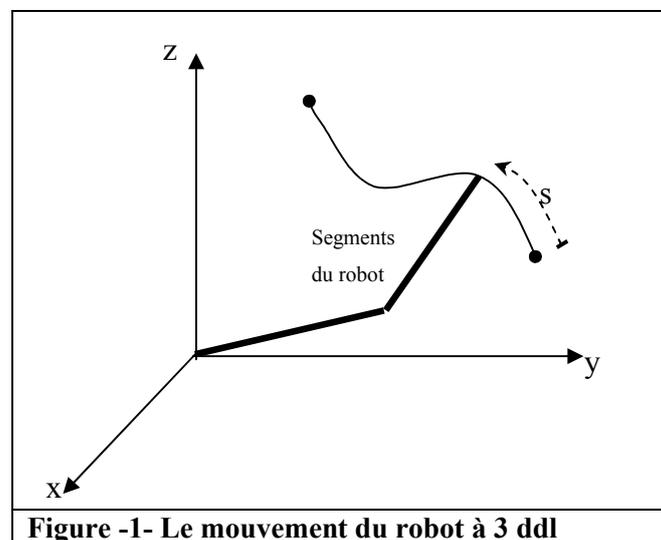
méthodes de résolution du problème de la commande optimale : méthodes directes, méthodes indirectes. Problème d'optimisation et méthodes de sa résolution y sont expliqués.

### III.1. Introduction

La planification de trajectoire est une partie primordiale dans le software dédié à l'exploitation d'un robot manipulateur. Généralement, il n'est pas suffisant de se contenter de la première solution faisable, mais les exigences actuelles sur les robots imposent de chercher parmi les solutions faisables la plus adaptée ou la solution optimale. Dans le domaine de la robotique, plusieurs critères ont été utilisés pour évaluer la solution optimale, (temps minimal, minimisation des efforts fournis par les actionneurs, minimisation des surcharges). En plus de ces critères la trajectoire obtenue doit répondre aux contraintes imposées par les dynamiques du robot, les limites sur les couples fournis par les actionneurs, les limites sur les variables articulaires ...

Dans ce chapitre nous allons faire une étude détaillée d'une méthode de planification de trajectoire à temps optimal en partant d'un chemin prédéfini de l'organe terminal. Ensuite, nous allons appliquer la méthode à un robot à 3 degrés de liberté et quelques exemples de chemins de l'outil. [18] [19]

### III.2. Formulation du problème de commande à temps optimal



Pour illustrer, nous allons utiliser la figure ci-dessus. Le mouvement du robot doit faire déplacer l'outil de la position initiale (en repo) vers une position finale qui est aussi une position de repo et ceci en poursuivant le chemin décrit par la courbe. Ici, chaque point sur le chemin de l'outil est paramétré par le paramètre  $s$  qui donne la distance à partir du point de repo initial. Les dynamiques de l'organe terminal sont supposées négligeables par rapport à celles des segments ce qui est le cas pour certains robots manipulateurs. [20]

Les équations dynamiques de tels robots manipulateurs peuvent être obtenues en utilisant le formalisme de Lagrange et ont la forme suivante :

$$M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) = \tau \quad (III.1)$$

Les vecteurs  $\theta$  et  $\tau$  sont les vecteurs des variables articulaires et des couples articulaires respectivement, tandis que  $M(\theta)$  est la matrice de masse du robot et  $h(\theta, \dot{\theta})$  compte pour les effets centrifuges, de Coriolis, de gravité et de frottement. [21]

Le problème de contrôle à temps optimale est comme suit : Etant données les équations du modèle dynamique (3.1), un chemin que l'outil doit suivre (figure -1-) et les contraintes sur les couples des actionneurs de la forme :

$$\tau_{\min} \leq \tau \leq \tau_{\max}(\theta, \dot{\theta}) \quad (III.2)$$

Trouver le profile des couples  $\tau(\theta)$  qui mène le manipulateur de la position initiale  $\theta_0$  à la position finale  $\theta_f$  en un temps minimal.

Dans la méthode que nous allons utiliser, le problème de commande à temps minimal est transformé en un problème mathématique équivalent de commande optimale dans lequel la seule variable de commande est  $\ddot{s}$  ; la deuxième dérivée de la distance de la position initiale, c'est à dire l'accélération de l'outil.

Pour ce nouveau problème, nous devons déterminer pour chaque position et vitesse, sur le chemin de l'outil, les contraintes sur l'accélération linéaire correspondant aux contraintes sur les couples articulaires. Ceci impose que les angles articulaires soient calculées comme fonctions de la variable  $s$  et que les vitesses articulaires et les accélérations articulaires soient calculées comme fonctions de  $s, \dot{s}$ , c'est-à-dire :

$$\begin{aligned} \theta &= \theta(s) \\ \dot{\theta} &= \dot{\theta}(s, \dot{s}) \\ \ddot{\theta} &= \ddot{\theta}(s, \dot{s}, \ddot{s}) \end{aligned} \quad (III.3)$$

Avec un chemin de l'outil spécifié au départ en fonction de  $s$ , alors  $\theta(s)$  peut être obtenue au moins de façon implicite à partir du modèle géométrique inverse du manipulateur. Similairement pour  $\dot{\theta}(s, \dot{s})$  et  $\ddot{\theta}(s, \dot{s}, \ddot{s})$  nous n'avons pas besoin d'écrire les expressions de ces quantités, la possibilité de les calculer numériquement selon le développement cinématique suivant est suffisante.

Pour obtenir les équations de  $\dot{\theta}(s, \dot{s})$  et  $\ddot{\theta}(s, \dot{s}, \ddot{s})$  il faut noter que le vecteur position de l'outil peut être considéré comme une fonction soit des variables articulaires  $\theta$  ou de la distance le long du chemin  $s$  :

$$p = p(\theta) = p(s) \quad (III.4)$$

La différentiation de cette équation par rapport au temps donne :

$$\dot{\mathbf{r}} = \mathbf{J} \dot{\mathbf{s}} \quad (III.5)$$

Ici  $\mathbf{J}$  est le jacobéen des dérivées partiels des composantes de vecteur position de l'outil du robot par rapport aux variables articulaires et  $\dot{\mathbf{s}}$  est un vecteur tangent au chemin de l'outil.

Ainsi quand le manipulateur n'est pas dans une configuration singulière le jacobéen  $\mathbf{J}$  est inversible et on peut résoudre l'équation (III.5) pour  $\dot{\mathbf{s}}$  :

$$\dot{\mathbf{s}}(\mathbf{s}, \dot{\mathbf{s}}) = \mathbf{J}^{-1} \dot{\mathbf{r}} \quad (III.6)$$

Nous devons noter qu'en pratique, les singularités sont évitées.

Pour obtenir une équation de  $\ddot{\mathbf{r}}(\mathbf{s}, \dot{\mathbf{s}}, \ddot{\mathbf{s}})$  suffit de dériver l'équation (III.5) :

$$\ddot{\mathbf{r}} = \dot{\mathbf{J}} \dot{\mathbf{s}} + \mathbf{J} \ddot{\mathbf{s}} \quad (III.7)$$

Avec  $\dot{\mathbf{J}}$  le dérivé de la matrice jacobéenne par rapport au temps,  $\ddot{\mathbf{r}}$  est le deuxième dérivé de  $\mathbf{r}$  par rapport à  $t$ . Le premier élément du deuxième terme de l'équation (III.7) est l'accélération tangentielle de l'outil  $\ddot{\mathbf{r}}$  et le deuxième élément est l'accélération normale.

Enfin, l'expression de  $\ddot{\mathbf{s}}$  est :

$$\ddot{\mathbf{s}} = \mathbf{J}^{-1} \ddot{\mathbf{r}} - \dot{\mathbf{J}}^{-1} \dot{\mathbf{s}} \quad (III.8)$$

Maintenant nous pouvons tirer les expressions de l'accélération maximale et l'accélération minimale que les actionneurs peuvent produire, à n'importe quelle distance  $s$  et vitesse  $\dot{\mathbf{s}}$  le long du chemin.

La substitution de l'équation (III.8) dans (III.1) conduit à :

$$\mathbf{M}(\mathbf{s}) \ddot{\mathbf{s}} + \mathbf{C}(\mathbf{s}, \dot{\mathbf{s}}) \dot{\mathbf{s}} + \mathbf{h}(\mathbf{s}) = \mathbf{0} \quad (III.9)$$

C'est-à-dire :

$$\ddot{\mathbf{s}} = -\mathbf{M}^{-1}(\mathbf{s}) [\mathbf{C}(\mathbf{s}, \dot{\mathbf{s}}) \dot{\mathbf{s}} + \mathbf{h}(\mathbf{s})] \quad (III.10)$$

Si on combine l'équation (III.9) avec les contraintes imposées par les actionneurs sur les couples articulaires on obtient :

$$\tau_i \leq \tau_{i, \max} \quad ; \quad i = 1, 2, 3 \quad (III.11)$$

Ainsi :

$$-\tau_{i, \max} \leq \tau_i \leq \tau_{i, \max} \quad ; \quad i = 1, 2, 3 \quad (III.12)$$

Vue que pour tout  $s$  et  $t$  sur le chemin de l'outil,  $(x(s), y(s))$  et  $(x(t), y(t))$  peuvent être calculées par le modèle géométrique inverse et l'équation (III.6),  $\ddot{x}_i$  et  $\ddot{y}_i$  qui sont fonctions de  $\ddot{s}$  (équation (III.2)) peuvent être écrits comme fonctions de  $(\dot{x}, \dot{y})$

Alors si  $\dot{s} \neq 0$  l'équation (III.12) peut s'écrire comme suit :

$$(\ddot{x}_i, \ddot{y}_i) \leq \ddot{s} \leq (\ddot{x}_i, \ddot{y}_i); \quad i = 1, 2, 3 \tag{III.13}$$

Où :

$$(\ddot{x}_i, \ddot{y}_i)' = \frac{(\ddot{x}_i, \ddot{y}_i)}{(\dot{x}, \dot{y})} \quad (\dot{x}, \dot{y}) > 0 \tag{III.14}$$

$$(\ddot{x}_i, \ddot{y}_i)' = \frac{(\ddot{x}_i, \ddot{y}_i)}{(\dot{x}, \dot{y})} \quad (\dot{x}, \dot{y}) < 0$$

$$(\ddot{x}_i, \ddot{y}_i)' = \frac{(\ddot{x}_i, \ddot{y}_i)}{(\dot{x}, \dot{y})} \quad (\dot{x}, \dot{y}) > 0 \tag{III.15}$$

$$(\ddot{x}_i, \ddot{y}_i)' = \frac{(\ddot{x}_i, \ddot{y}_i)}{(\dot{x}, \dot{y})} \quad (\dot{x}, \dot{y}) < 0$$

Quand  $\dot{s} \neq 0$  les équations (III.14) et (III.15) donnent l'intervalle de l'accélération de l'outil  $\ddot{s}$  dans lequel les actionneurs peuvent maintenir le robot sur la trajectoire sans violer les contraintes de l'équation (III.2). Pour que l'outil reste sur le chemin prédéfini, il faut que  $\ddot{s}$  appartienne à l'intersection des intervalles  $[(\ddot{x}_i, \ddot{y}_i)'; (\ddot{x}_i, \ddot{y}_i)']$  pour lesquels  $\dot{s} \neq 0$ . Il peut arriver que les intervalles  $[(\ddot{x}_i, \ddot{y}_i)'; (\ddot{x}_i, \ddot{y}_i)']$  ne se coupent pas, ceci est le cas généralement quand la vitesse est trop grande, dans ce cas l'outil va immédiatement quitter la trajectoire. D'un autre côté, si  $\dot{s} \neq 0$  et que l'intersection des intervalles  $[(\ddot{x}_i, \ddot{y}_i)'; (\ddot{x}_i, \ddot{y}_i)']$  n'est pas vide, alors la condition nécessaire et suffisante pour que l'outil reste sur le chemin est que l'équation (III.13) soit vérifiée pour tout  $i$ .

Malgré que les équations (III.13) doivent être vérifiées pour tout  $i$ , si  $\dot{s} = 0$  pour un  $i$  donné, alors le choix de  $\ddot{s}$  ne peut pas affecter le fait que l'équation (III.13) soit vérifiée ou non pour ce  $i$ . Dans ce cas, le choix de  $\ddot{s}$  sera fait en considérant seulement l'équation (III.13) pour les autres valeurs de  $i$ . Tant que le manipulateur n'est pas dans une pose singulière, au moins un  $\dot{s} \neq 0$  ceci est la conséquence de l'expression de  $\dot{s}$  dans le terme de gauche de l'équation (III.10) vue que la matrice de masse est positive et que  $\mathbf{u}$  est un vecteur unitaire.

Une accélération admissible est une accélération pour laquelle les actionneurs peuvent maintenir l'outil sur le chemin prescrit sans violer les contraintes. Une accélération admissible doit alors vérifier :

$$(\ddot{x}_i, \ddot{y}_i)' = \max (\ddot{x}_i, \ddot{y}_i)' \leq \ddot{s} \leq (\ddot{x}_i, \ddot{y}_i)' = \min (\ddot{x}_i, \ddot{y}_i)' \tag{III.16}$$

Avec le maximum et le minimum prises pour les valeurs de  $i$  pour lesquelles  $\dot{q}_i \neq 0$ . Notons que si les intervalles  $[(t_i, t_{i+1}); (t_{i+1}, t_{i+2})]$  ne se coupent pas, alors  $\dot{q}_i > \dot{q}_{i+1}$ . Maintenir  $\dot{q}_i \leq \dot{q}_{i+1}$  telle qu'une accélération admissible existe est l'élément clé dans la solution du problème de commande à temps optimal. Enfin la version mathématique du problème de commande optimal peut être énoncée comme suit :

Etant donné  $q(0), \dot{q}(0)$  choisir  $\ddot{q}$  pour minimiser le temps final  $t_f$  pour lequel  $\dot{q}(t_f) = \dot{q}_f$  et  $q(t_f) = q_f$  soumis à la contrainte donnée par l'équation (III.16).

Quoique qu'on a transformé le problème de commande du manipulateur en un problème, purement mathématique, de commande à temps optimal ou la commande est la variable  $\ddot{q}$ ; nous n'avons pas perdu de vue que les commandes physiques qui sont les couple articulaires. Ces derniers peuvent être calculés à partir de  $\ddot{q}$ , en utilisant l'équation (III.10).

### III.3. Solution du problème de la commande à temps optimal :

L'idée de base derrière la solution du problème de commande à temps optimale de robots manipulateurs est de choisir l'accélération  $\ddot{q}$  pour obtenir la plus grande vitesse  $\dot{q}$  possible à tous les points sans violer la condition  $\dot{q}_i \leq \dot{q}_{i+1}$ . Ceci est justifié par le fait que :

$$\dot{q}_i = \int \ddot{q}_i dt \quad (III.17)$$

Il a été montré dans [18], que pour minimiser  $t_f$ , l'accélération doit à tout moment prendre soit sa valeur maximale ou sa valeur minimale. C'est-à-dire que  $\ddot{q} = \ddot{q}_{max}$  ou  $\ddot{q} = \ddot{q}_{min}$ . Ainsi, trouver la loi de commande optimale revient à trouver les instants ou positions auxquels  $\ddot{q}$  commute entre l'accélération maximale et la décélération minimale. [18] [19]

La meilleure approche pour trouver les positions de commutation est de tracer la courbe de commutation dans le plan de phase  $(q, \dot{q})$ . Nous commençons par noter que, généralement, pour chaque point  $s$ , il y a une vitesse  $\dot{q}$  de l'outil au-delà de laquelle aucune combinaison des couples articulaires ne peut maintenir l'outil sur le chemin prescrit. Pour un chemin donné de l'organe terminal, nous avons alors une courbe de vitesse maximale dans le plan de phase comme montré sur la figure -2-. En dessous de la courbe, nous avons  $\dot{q}_i \leq \dot{q}_{i+1}$ ; tel que  $\ddot{q}$  peut être choisi pour satisfaire l'équation (III.16). Au dessus de la courbe, nous avons  $\dot{q}_i > \dot{q}_{i+1}$ ; et il n'existe aucun choix admissible pour  $\ddot{q}$ . La courbe satisfait l'équation :

$$\dot{q}_i = \dot{q}_{i+1} \quad (III.18)$$

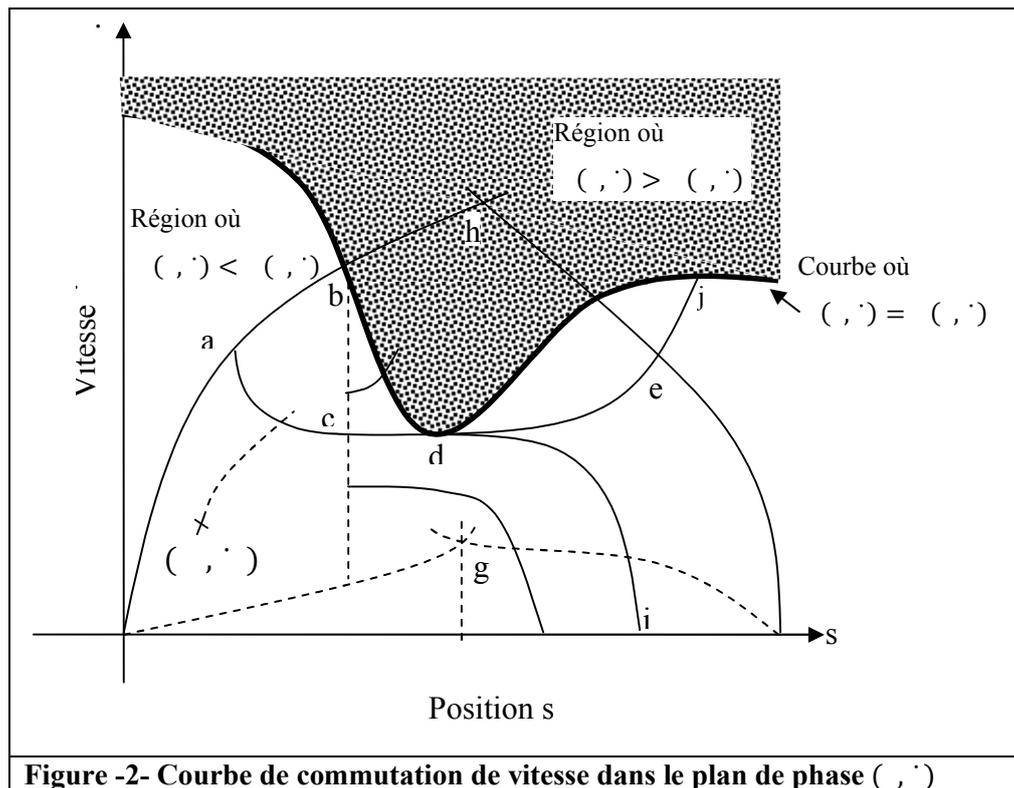


Figure -2- Courbe de commutation de vitesse dans le plan de phase  $(\dot{s}, \ddot{s})$

- 1) Une courbe délimitant les deux régions  $(\dot{s}, \ddot{s}) > (\dot{s}, \ddot{s})$  et  $(\dot{s}, \ddot{s}) < (\dot{s}, \ddot{s})$  est tracée dans le plan de phase. Pour tracer cette courbe nous avons procédé comme suit : Après avoir échantillonné la variable  $s$  dans tout l'intervalle où le chemin prescrit est défini, nous avons résolu l'équation (III.18) pour chaque valeur de  $s$ , c'est-à-dire nous avons calculé la valeur de  $\dot{s}$  qui satisfait l'équation (III.18). Pour cela nous avons utilisé la fonction `fzero` de MATLAB pour chercher une racine positive de l'équation  $(\dot{s}, \ddot{s}) - (\dot{s}, \ddot{s}) = 0$ . La fonction (f-g) doit être modifiée pour que `fzero` ne converge que vers les racines positives. A la fin de cette procédure une courbe  $\dot{s} = \dot{s}(s)$  spécifiant la vitesse maximale autorisée comme celle représentée sur la figure sera obtenue.
- 2) Pour que l'outil traverse la trajectoire désirée, la commande doit commencer par faire accélérer le robot avec une accélération maximale  $\ddot{s} = (\ddot{s}, \ddot{a})$  partir de sa position de repo initiale. Quand l'outil s'approche du point de destination la commande doit décélérer le robot en appliquant la décélération maximale pour que l'outil arrive au point avec une vitesse nulle. En se basant sur ces deux idées, on commence par intégrer la fonction  $\ddot{s} = (\ddot{s}, \ddot{e})$  avant de  $\dot{s}$  à  $s_0$  et la fonction  $\ddot{s} = (\ddot{s}, \ddot{v})$  vers l'arrière de  $\dot{s}$  à  $s_f$ . Selon le résultat obtenu, deux cas sont possibles :
  - a) Les deux courbes obtenues se coupent avant qu'elles touchent la courbe de la vitesse maximale autorisée, ceci est représenté sur la figure -2- par les courbes en pointillés

- [ ] et [ ]. Dans ce cas une seule commutation est nécessaire pour la commande à temps optimale. On applique l'accélération maximale  $\ddot{x} = ( \ , j )$  jusqu'à ce qu'on arrive au point  $\ ,$  là on commute vers la décélération maximale  $\ddot{x} = ( \ , j )$  jusqu'à ce qu'on arrive au point finale  $\ .$
- b) Les deux courbes obtenues par l'intégration de  $\ddot{x} = ( \ , è )$  avant et de  $\ddot{x} = ( \ , ' )$  vers l'arrière ne se coupent qu'après avoir sortie de la région autorisée. Ce cas est représenté sur la figure -2- par la courbe [  $h$  ]. Pour ce cas plusieurs commutations doivent être faites avant d'arriver au point de destination. Le calcul de ces instants de commutation pour obtenir une réponse à temps optimale est la phase la plus difficile de cette méthode.
- 3) Dans le cas où les courbes obtenues par intégration vers l'avant de  $\ddot{x} = ( \ , à )$  et par intégration vers l'arrière de  $\ddot{x} = ( \ , ä )$  ne se coupent qu'après avoir sortie de la région de vitesse autorisée (courbe [  $h$  ] sur la figure -2-), des commutations antérieures entre l'accélération et la décélération sont nécessaires pour empêcher le développement d'une vitesse trop grande. Pour trouver les points de commutation nous avons procédé comme suit :
- a) Nous commençons par déterminer le point d'intersection de la courbe obtenue par intégration de  $\ddot{x} = ( \ , à )$  avec la courbe de vitesse maximale que nous avons appelé  $b$  et dont les coordonnées sont dénotées  $(( \ , ' )$ .
- b) On cherche le point  $c$  situé en dessous du point  $b$  (c'est-à-dire  $\ = \ < ' )$  tel que la courbe obtenue par intégration de  $\ddot{x} = ( \ , à )$  à partir de  $( \ , ' )$  soit tangente à la courbe de vitesse maximale autorisée mais elle ne doit pas la couper. Ceci est en quelque sorte un problème à deux valeurs aux limites. Pour résoudre ce problème, le programme que nous avons fait, commence par un intervalle initial  $[ ' \ , ' ]$  et utilise la méthode de bisection pour chercher la solution désirée  $'$  qui donne une courbe tangente à la courbe de la vitesse maximale autorisée. L'idée de base derrière ceci est que si on augmente  $'$  la courbe obtenue tend toujours à s'approcher de la courbe de vitesse maximale autorisée de même si on diminue  $'$  la courbe obtenue tend toujours à s'éloigner de la courbe de vitesse maximale autorisée. Ainsi si on arrive à trouver deux valeurs de  $' < '$  la première donnant une courbe qui ne coupe pas la courbe de vitesse maximale autorisée et la deuxième donnant une courbe qui la coupe, alors on est sûr que la valeur de  $'$  optimale est située entre ces deux valeurs  $' \in [ ' \ , ' ]$ . Nous avons utilisé l'erreur relative sur  $'$  comme

critère d'arrêt. Sur la figure -2- les points  $b$ ,  $c$  sont deux exemples de points à rejeter. Une fois le point  $c$  déterminé, nous lisons à partir de la courbe les coordonnées du point de tangence  $d$ . Ensuite à partir du point  $c$ , nous intégrons  $\ddot{x} = -\left(\frac{v}{v_{max}}\right)^2$  vers l'arrière jusqu'au point initial. La courbe obtenue est utilisée pour déterminer les coordonnées du point  $a$ .

- c) Maintenant, à partir du point  $d$  on trace la courbe obtenue par intégration de  $\ddot{x} = \left(\frac{v}{v_{max}}\right)^2$  vers l'avant jusqu'au point final. Deux cas peuvent être rencontrés, le premier est que la courbe obtenue coupe la courbe de vitesse maximale autorisée avant qu'elle coupe la courbe  $[h]$ , dans ce cas il faut revenir vers l'étape (3.b) avec le point  $j$  remplaçant le point  $b$ . Le deuxième cas est celui représenté par la figure -2-, c'est-à-dire que la courbe obtenue coupe la courbe  $[h]$  (point  $e$  sur la figure) avant qu'elle coupe la courbe de vitesse maximale autorisée. Dans ce cas la solution est terminée et les points de commutation sont  $a$ ,  $d$  et  $e$ .

Le profile de vitesse qui donne la solution à temps optimal est donné par la courbe  $[v]$ .

La solution du problème de commande à temps minimal à partir de tout autre point initial situé en dessous de cette courbe  $(x_0, v_0)$  sera obtenue en intégrant vers l'avant  $\ddot{x} = \left(\frac{v}{v_{max}}\right)^2$  jusqu'à ce qu'on rencontre la courbe  $[h]$  ensuite on continue sur cette dernière.

Le programme que nous avons mis au point est structuré selon le schéma bloc de la figure --.

#### III.4. Résultats de simulations :

Le programme que nous avons mis au point a été utilisé pour le calcul de la commande à temps optimal correspondant à un ensemble de chemins que doit suivre l'outil d'un robot à trois ddl. Le bras manipulateur que nous avons considéré est représenté sur la figure suivante :

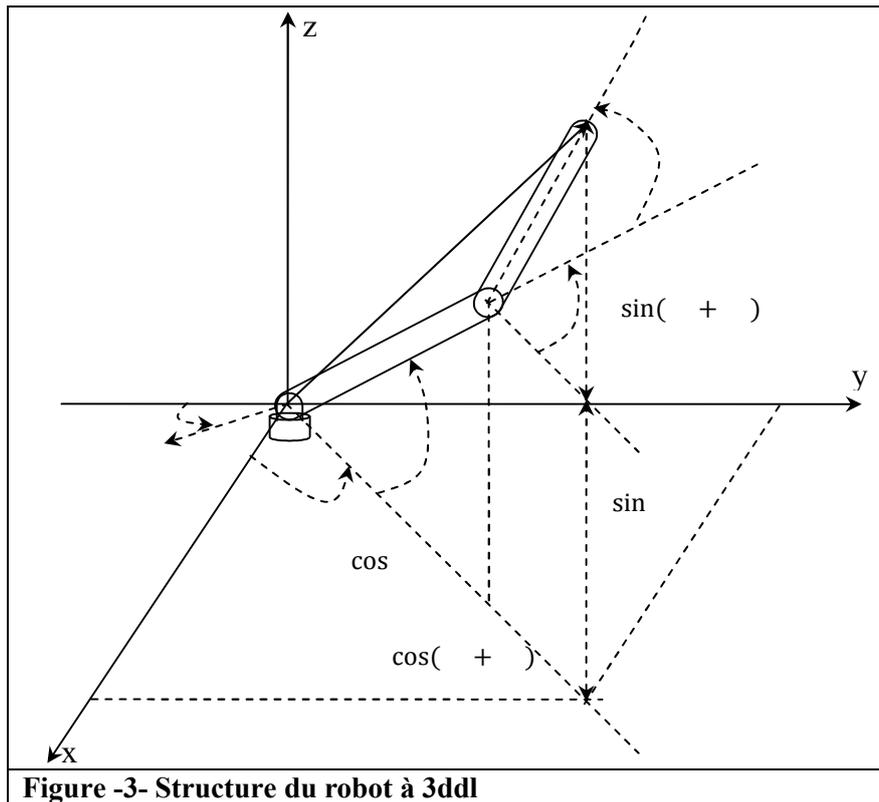


Figure -3- Structure du robot à 3ddl

C'est un robot à trois degrés de libertés (partie positionnement). Les trois articulations sont rotoïdes, les variables articulaires sont  $\alpha, \beta, \gamma$ .

### III.4.1 Cinématique du robot :

Le repère inertiel est le repère  $\{0\}$  (xyz) représenté sur la figure. Les repères associés aux segments 1, 2, 3 sont les repères  $\{1\}, \{2\}$  et  $\{3\}$ . Le repère  $\{1\}$  a un axe  $z_1$  parallèle à celui du repère inertiel. L'axe  $z_2$  du repère  $\{2\}$  est celui de l'articulation 2 ( $\beta$ ). L'axe  $z_3$  est parallèle à  $z_2$  mais l'origine du repère  $\{3\}$  est sur l'axe de rotation de l'articulation  $\{3\}$ .

Les trois matrices de transformation permettant de transformer les coordonnées des repères  $\{1\}, \{2\}$  et  $\{3\}$  vers le repère inertiel sont données par :

$${}^0T_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \quad (III.19)$$

$${}^1T_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (III.20)$$

$${}^2T_3 = \begin{pmatrix} \cos(\alpha + \beta) & -\sin(\alpha + \beta) & 0 \\ \sin(\alpha + \beta) & \cos(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (III.21)$$

La position de l'organe terminale peut être tirée directement de la figure et elle est donnée par le vecteur :

$$= \begin{bmatrix} + \\ + \\ + \end{bmatrix} \begin{bmatrix} ( + ) \\ ( + ) \\ ( + ) \end{bmatrix} \text{ ainsi } \| \| = + + 2 \quad (III.22)$$

La dernière équation permet de tirer une expression de la variable articulaire .

$$= \pm \frac{\| \|}{\| \|} \quad (III.23)$$

Ainsi selon le signe de deux solutions sont possibles pour le modèle géométrique inverse.

Pour calculer , il suffit de constater à partir de l'expression de que :

$$= [ 2( ), ( ) ] \quad (III.24)$$

Nous avons aussi

$$( ) = + ( + ) = ( + ) + \quad (III.24)$$

$$\frac{( ) + ( )}{( ) + ( )} = + ( + ) = ( + ) - \quad (III.25)$$

La solution de ces deux équations pour et donne :

$$\left\{ \begin{array}{l} = \frac{( ) ( ) ( ) ( )}{( ) ( ) ( ) ( )} \\ = \frac{( ) ( ) ( ) ( )}{( ) ( ) ( ) ( )} \end{array} \right. \quad (III.26)$$

Cette solution vérifie la condition + = 1 et est alors une solution valide. La variable articulaire est alors donnée par :

$$= \frac{2 ( ) + ( + ) \sqrt{( ) + ( )}, -( + ) ( ) + \sqrt{( ) + ( )}}{( ) + ( )} \quad (III.27)$$

Pour le calcul des pulsations angulaires , et , on peut utiliser la relation = , ou bien, on tire leurs expressions directement à partir de la figure.

Pour , la rotation s'effectue autour de l'axe z avec un angle son expression est, alors, évidente = [0 0 ] . Si on l'exprime dans le repère 1 elle reste inchangée.

Pour la pulsation : Le segment 2 fait une rotation qui est composée de deux rotations dont les axes se coupent. La première est autour de l'axe avec un angle , et la deuxième est autour de l'axe = - avec un angle .

Ainsi = - . Si on l'exprime dans le repère {2} on obtient : = .

Pour la pulsation : Les segment 3 fait une rotation qui est composée de trois rotations : La première est autour de l'axe avec un angle , et la deuxième est autour de l'axe = - avec un angle . La troisième rotation est autour de l'axe || avec un

angle , ainsi :  $\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} = - \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix}$  si on l'exprime dans le repère trois on obtient :

$$= \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix}.$$

### III.4.2 Dynamiques du robot :

L'énergie cinétique du système est donnée par :  $T = \frac{1}{2} \dot{\theta}^T M \dot{\theta}$

Nous avons :

$$= - ; \dot{T} = - - ; \dot{T} = - +$$

$$= \begin{bmatrix} + - (\dot{\theta}_1 + \dot{\theta}_2) \\ + - (\dot{\theta}_1 + \dot{\theta}_2) \\ + - (\dot{\theta}_1 + \dot{\theta}_2) \end{bmatrix} \quad (III.28)$$

$$\dot{T} = \begin{bmatrix} - \dot{\theta}_1 - - (\dot{\theta}_1 + \dot{\theta}_2)(\dot{\theta}_1 + \dot{\theta}_2) & - & + - (\dot{\theta}_1 + \dot{\theta}_2) \\ - \dot{\theta}_1 - - (\dot{\theta}_1 + \dot{\theta}_2)(\dot{\theta}_1 + \dot{\theta}_2) & + & + - (\dot{\theta}_1 + \dot{\theta}_2) \\ & \dot{\theta}_1 + - (\dot{\theta}_1 + \dot{\theta}_2)(\dot{\theta}_1 + \dot{\theta}_2) & \end{bmatrix}$$

(III.29)

$$\dot{T} = + \dot{\theta}_1 + = \dot{\theta}_1 + - (\dot{\theta}_1 + \dot{\theta}_2) \dot{\theta}_1 + (\dot{\theta}_1 + \dot{\theta}_2) \dot{\theta}_1 + - \dot{\theta}_1 + \dot{\theta}_1 + \dot{\theta}_1 + \dot{\theta}_1$$

Les tenseurs d'inertie des deux segments sont considérés diagonaux (ceci est vérifié si les segments sont symétriques par rapport aux axes du repère passant par le centre de gravité du segment. Prises par rapport à ce repère, ces tenseurs sont indépendants des variables articulaires.

$$= \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} ; \quad = \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix}$$

L'énergie cinétique due à la rotation du segment 1 est donnée par :

$$= \frac{1}{2} \dot{\theta}_1^2 = \frac{1}{2} \dot{\theta}_1^2 + \frac{1}{2} \dot{\theta}_2^2$$

De même l'énergie cinétique due à la rotation du deuxième segment est :

$$= \frac{1}{2} \dot{\theta}_2^2 = \frac{1}{2} (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2} \dot{\theta}_2^2$$

L'énergie cinétique due au déplacement du segment 1 est :

$$= \frac{1}{2} \dot{x}_1^2 = \frac{1}{2} \dot{x}_1^2 + \frac{1}{2} \dot{y}_1^2$$

L'énergie cinétique due au déplacement du segment 2 est :

$$= \frac{1}{2} \dot{x}_2^2 + \frac{1}{2} \dot{y}_2^2 = \frac{1}{2} (\dot{x}_1 + \dot{x}_2)^2 + \frac{1}{2} (\dot{y}_1 + \dot{y}_2)^2$$

L'énergie potentielle du robot est donnée par :

$$= m_1 g y_1 + m_2 g y_2 = \frac{1}{2} m_1 g y_1^2 + \frac{1}{2} m_2 g y_2^2$$

Le lagrangien du robot est donnée par :

$$L = T - V \tag{III.30}$$

Les expressions des couples articulaires peuvent être obtenues par la relation de Lagrange :

$$\tau_i = \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} \tag{III.31}$$

Enfin, les équations des dynamiques du robot sont données par :

$$= - \frac{\partial L}{\partial \theta_1} = - m_1 g y_1 - m_2 g y_2 + \dots$$





ou le chemin est paramétré par la distance  $s$ . Les équations de cette trajectoire sont comme suit :

$$= \frac{\dots \sqrt{\dots} \dots ( \dots )}{\dots \sqrt{\dots} \dots ( \dots )} \quad 0 \leq < \quad (III.38)$$

$$= \frac{\dots \sqrt{\dots} \dots ( \dots )}{\dots \sqrt{\dots} \dots ( \dots )} \quad 0 \leq < 2 * \quad (III.39)$$

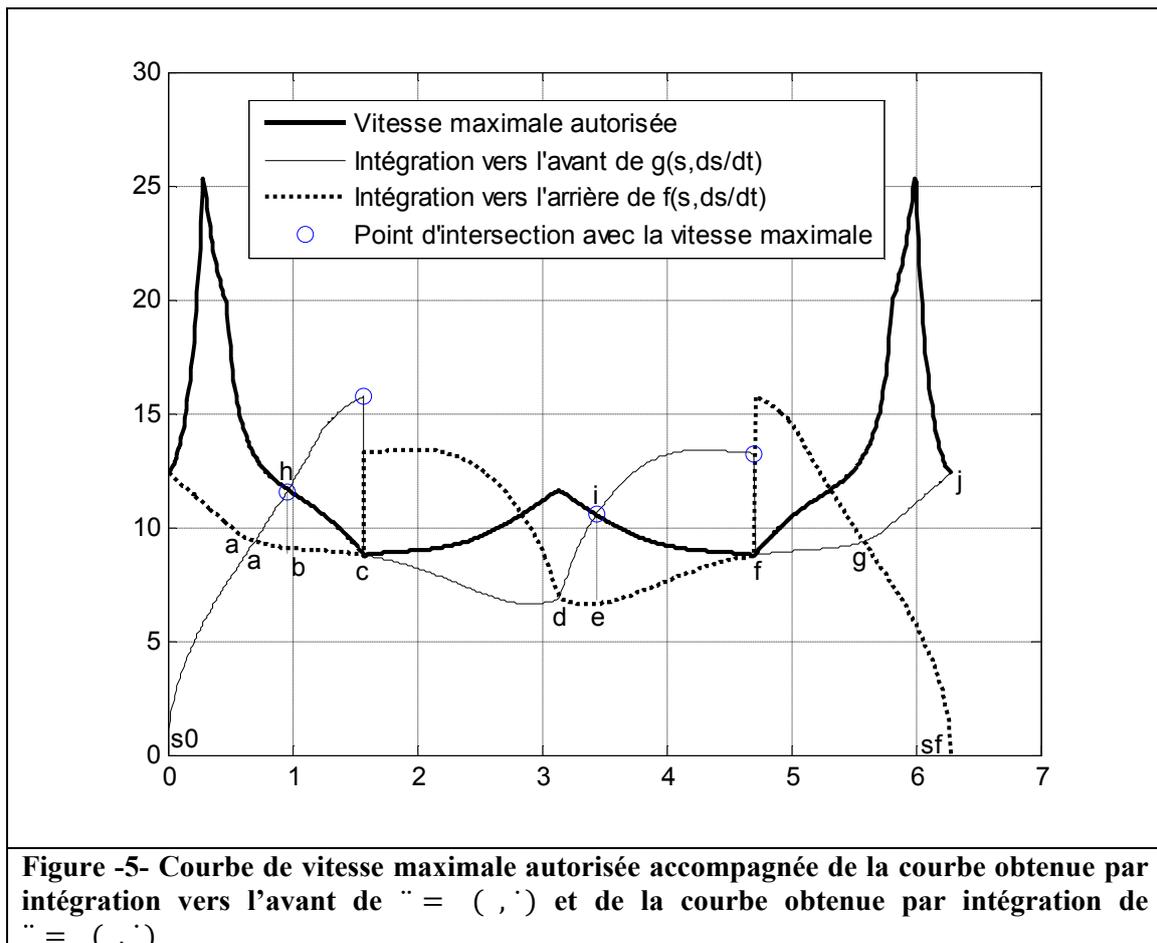
$$= 0.4 \quad 0 \leq < 2 * \quad (III.40)$$

Il faut noter que cette trajectoire ne passe pas par le point  $(0,0,0.25)$  qui nécessite une configuration singulière.

Nous avons fixé les limites sur les couples délivrés par les actionneurs selon l'équation suivante :

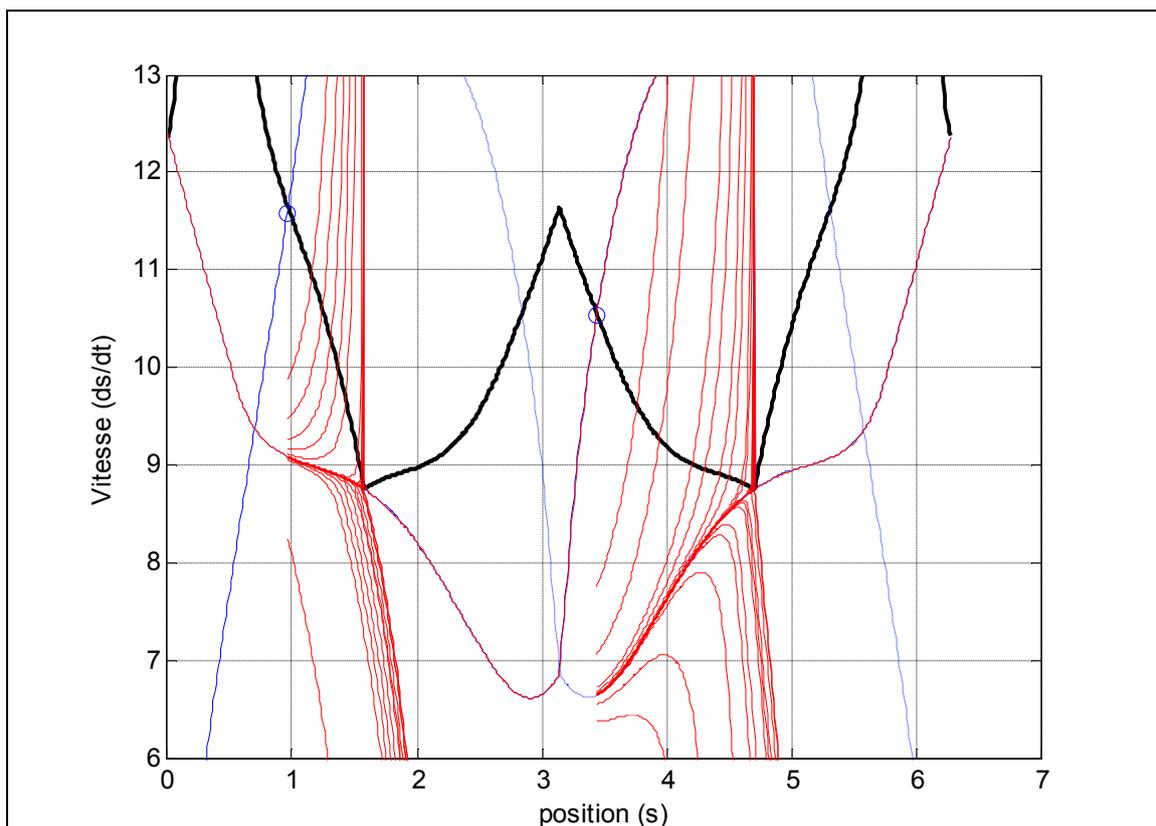
$$= -0.01 \dot{\phantom{x}} + 400 \quad ; \quad = -0.01 \dot{\phantom{x}} - 400 \quad (III.41)$$

Après implémentation de cette trajectoire et de ses dérivés par rapport à  $u$ , nous avons exécuté le programme et nous avons aboutie à la courbe de vitesse maximale autorisée suivante :



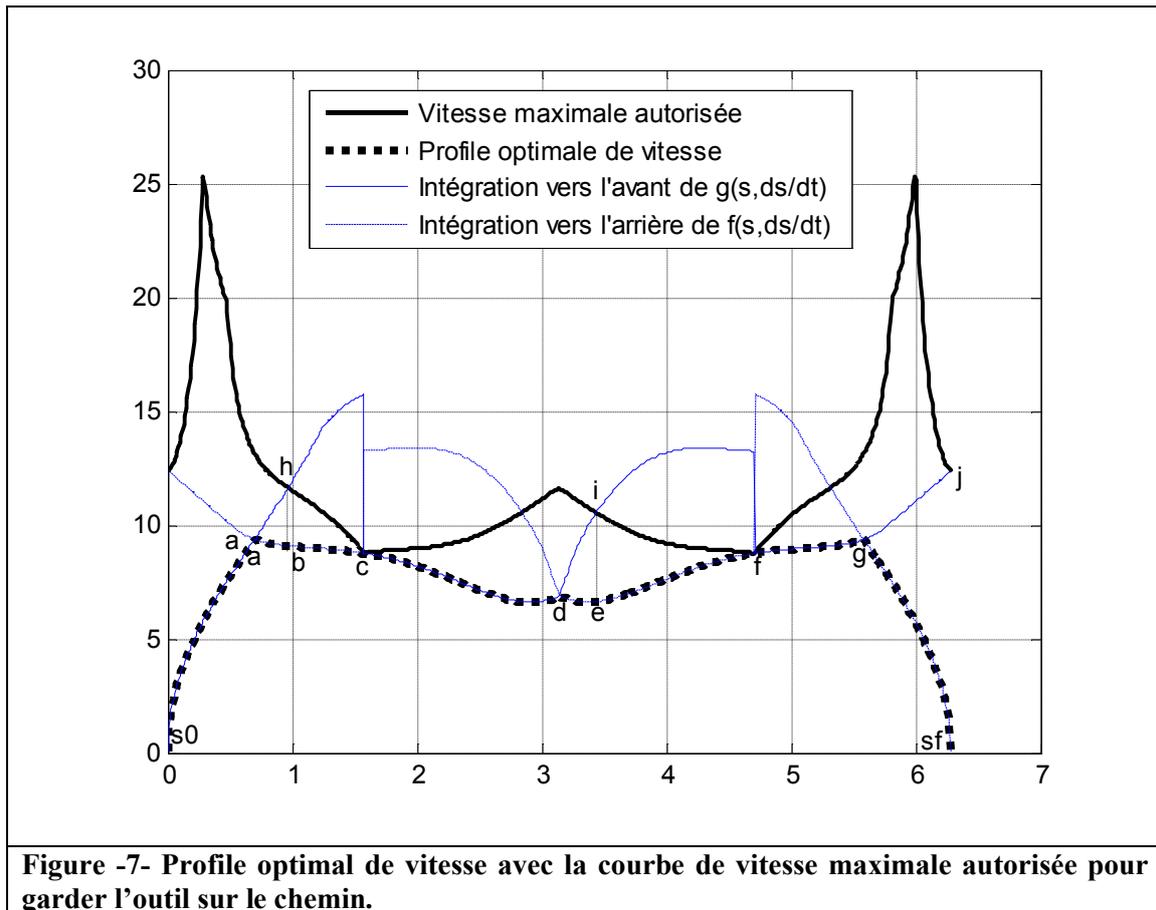
Nous remarquons qu'il s'agit ici d'un cas de plusieurs commutations. Alors nous avons procédé à la recherche des positions de commutations. A partir du premier point d'intersection avec la courbe de vitesse maximale autorisée, le programme a fait les itérations nécessaires pour résoudre le problème à deux valeurs aux limites, c'est-à-dire chercher la vitesse initiale qui donne une courbe tangente à la courbe de vitesse maximale autorisée. La figure -6- représente les itérations effectuées pour résoudre les deux problèmes à 2 valeurs aux limites rencontrés dans ce cas. Nous remarquons que ce problème est très sensible à la position du point initial et la tolérance doit être serrée pour obtenir une courbe qui est vraiment tangente à la courbe de vitesse maximale autorisée. Pour notre cas nous sommes contenté de la tolérance  $5.3295 \times 10^{-5}$ , le point initial correspondant est  $(x, y) = (0.96, 9.08700733216385)$ .

Pour le deuxième cas rencontré pendant la résolution du problème similaire, la tolérance sur la vitesse initiale est  $1.54 \times 10^{-5}$  et le point initial est  $(x, y) = (3.44, 6.6457)$ .



**Figure -6- Les itérations effectuées pour chercher les tangentes à la courbe de vitesse maximale autorisée.**

Le profile optimal, de la vitesse, obtenu pour ce cas est représenté dans figure -7- avec la courbe de vitesse maximale autorisée.



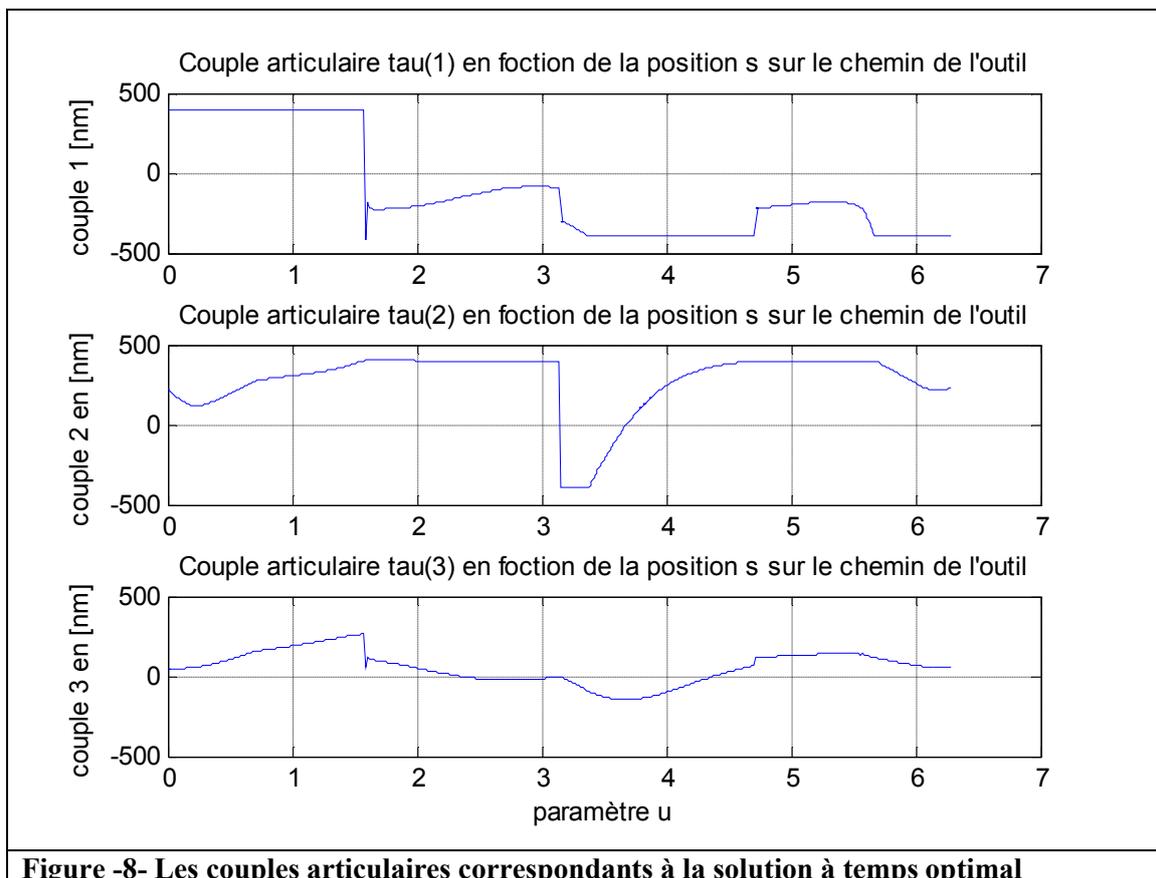
**Figure -7- Profile optimal de vitesse avec la courbe de vitesse maximale autorisée pour garder l'outil sur le chemin.**

Pour construire le profile optimal de vitesse le programme procède comme suit :

1. Au début le profile de vitesse optimale = portion [ [  $s_0$   $h$  ]
2. A chaque itération, on détecte le point d'intersection de l'intégration vers l'arrière de  $v = (s, v)$  avec la portion précédente de la courbe de l'intégration de  $v = (s, v)$  (Pour la première itération c'est le point a et pour la deuxième itération c'est le point d), ensuite le programme efface la portion inutile (portion [  $s_0$   $h$  ] pour la première itération et portion [  $s_0$   $d$  ] pour la deuxième) et rajoute la nouvelle portion calculée (pour la première itération : profile optimale de vitesse = [  $s_0$   $h$  ] + [  $h$   $a$  ] + [  $a$   $c$  ] . Tandis pour la deuxième itération c'est : profile optimale de vitesse = [  $s_0$   $h$  ] + [  $h$   $a$  ] + [  $a$   $d$  ] + [  $d$   $e$  ] + [  $e$   $g$  ]
3. A la fin, le programme détecte le point d'intersection de la dernière portion obtenue par intégration  $v = (s, v)$  vers l'arrière et la d'intégration de  $v = (s, v)$  (dans ce cas c'est le point g), ensuite il efface la dernière portion inutile (ici c'est [  $s_0$   $g$  ] ) et il rajoute la dernière portion du profile optimale (dans ce cas c'est la portion [  $g$   $sf$  ] ). Le profile optimal final de vitesse est alors [  $s_0$   $h$  ] + [  $h$   $a$  ] + [  $a$   $d$  ] + [  $d$   $e$  ] + [  $e$   $g$  ] + [  $g$   $sf$  ] .

Maintenant, après avoir résolu le problème de commande mathématique, nous pouvons obtenir les couples articulaires correspondants en appliquant l'équation (III.10). Nous obtenons alors les trois courbes de la figure -8- .

Nous remarquons qu'à chaque position au moins l'un des trois couples est à sa valeur maximale, ceci s'explique par le fait que la trajectoire doit être traversée le plus vite possible (temps minimal).



**Figure -8- Les couples articulaires correspondants à la solution à temps optimal**

Le temps final est donné par l'équation (III.17). Nous avons utilisé l'approximation trapézoïdale pour calculer l'intégral en question. La valeur obtenue est  $t_f = 0.8868$ . Il faut noter ici que dans le calcul de  $t_f$  on exclu la première et la dernière valeur de  $s$  car ces deux valeurs sont proches de zéro et leurs inverses sont  $\infty$ .

#### III.4.4. Résultat pour la deuxième trajectoire :

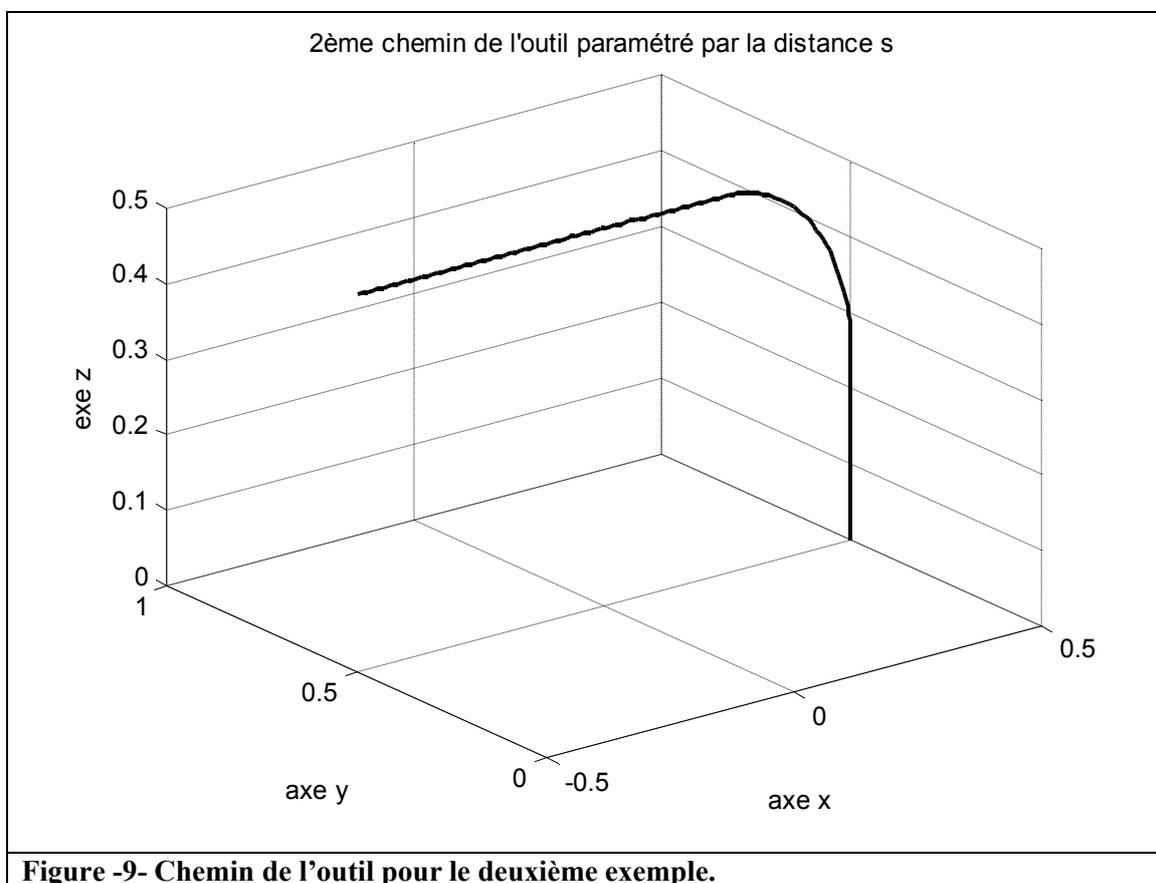
La deuxième trajectoire, que nous avons choisie, est représentée sur la figure -9-. Pour la représentation paramétrique de cette trajectoire nous avons choisi la distance  $s$  comme paramètre. Les équations sont comme suit :

$$(1) = \begin{cases} 0.25 + 0.25 \cos \frac{0.5}{0.25} s & 0 \leq s < 0.25 \\ -0.5 + 0.5 \cos \frac{0.5}{0.25} (s - 0.25) & 0.25 \leq s < 0.5 \end{cases} ; \quad (2) = 0.5 \quad (III.42)$$

$$(3) = \begin{cases} 0.25 + 0.25 \sin \frac{0.5}{0.25} s & 0 \leq s < 0.25 \\ 0.25 + 0.25 \sin \frac{0.5}{0.25} (s - 0.25) & 0.25 \leq s < 0.5 \end{cases} \quad (III.43)$$

Pour les limites sur les couples que peuvent fournir les actionneurs nous les avons prises sous la forme :  $\tau_x = -0.01 \dot{s} + 250$  ;  $\tau_y = -0.01 \dot{s} - 250$

Le chemin de l'outil est illustré par la figure -9-.



**Figure -9- Chemin de l'outil pour le deuxième exemple.**

La courbe de la vitesse maximale autorisée ainsi que le profile de vitesse correspondant à la solution à temps optimal sont représentés par la figure -10-. Nous remarquons ici que les courbes d'intégration de  $(\dot{s}, t)$  de  $(s, t)$  coupent avant de sortir de la région de vitesse autorisée. Ainsi, une seule commutation est nécessaire pour la solution à temps optimal. L'instant de commutation correspond à la position  $s = 0.88$

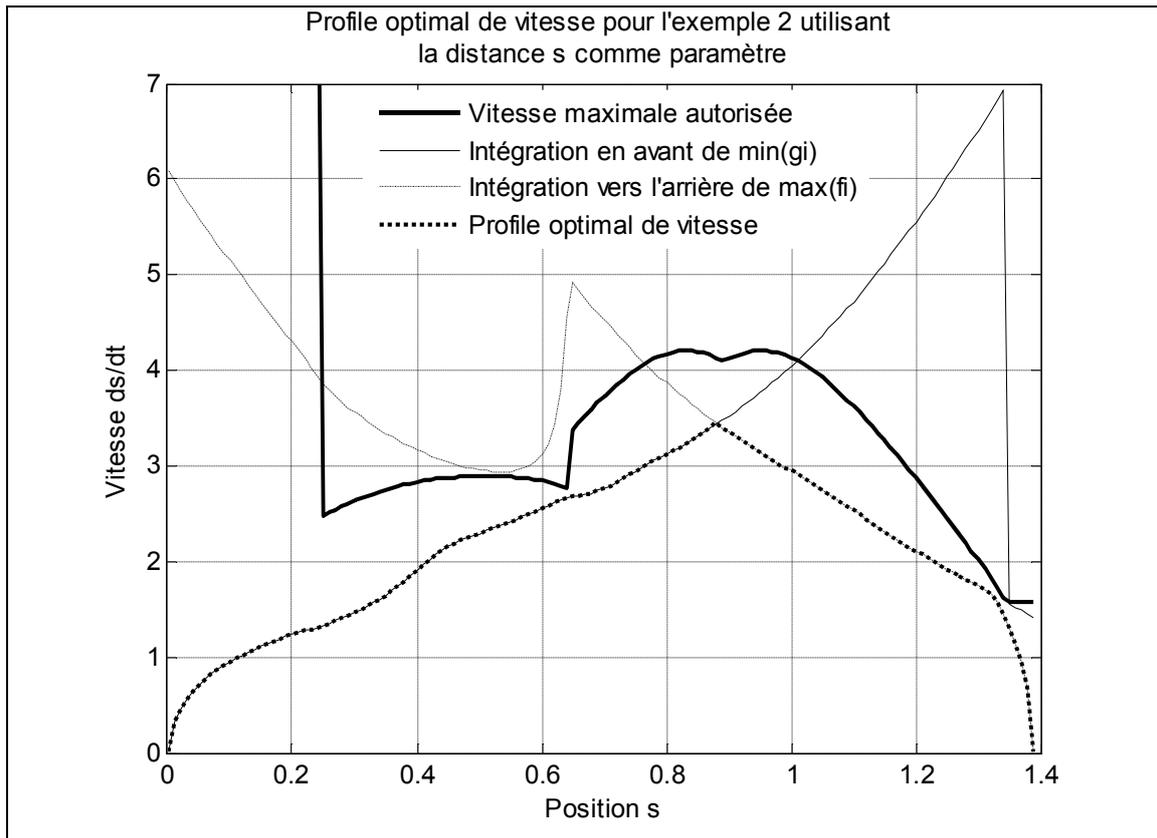


Figure -10- Profile optimal de vitesse et la courbe de vitesse maximale autorisée correspondant au chemin de la figure -9-

Les couples articulaires sont représentés sur la figure -11-.

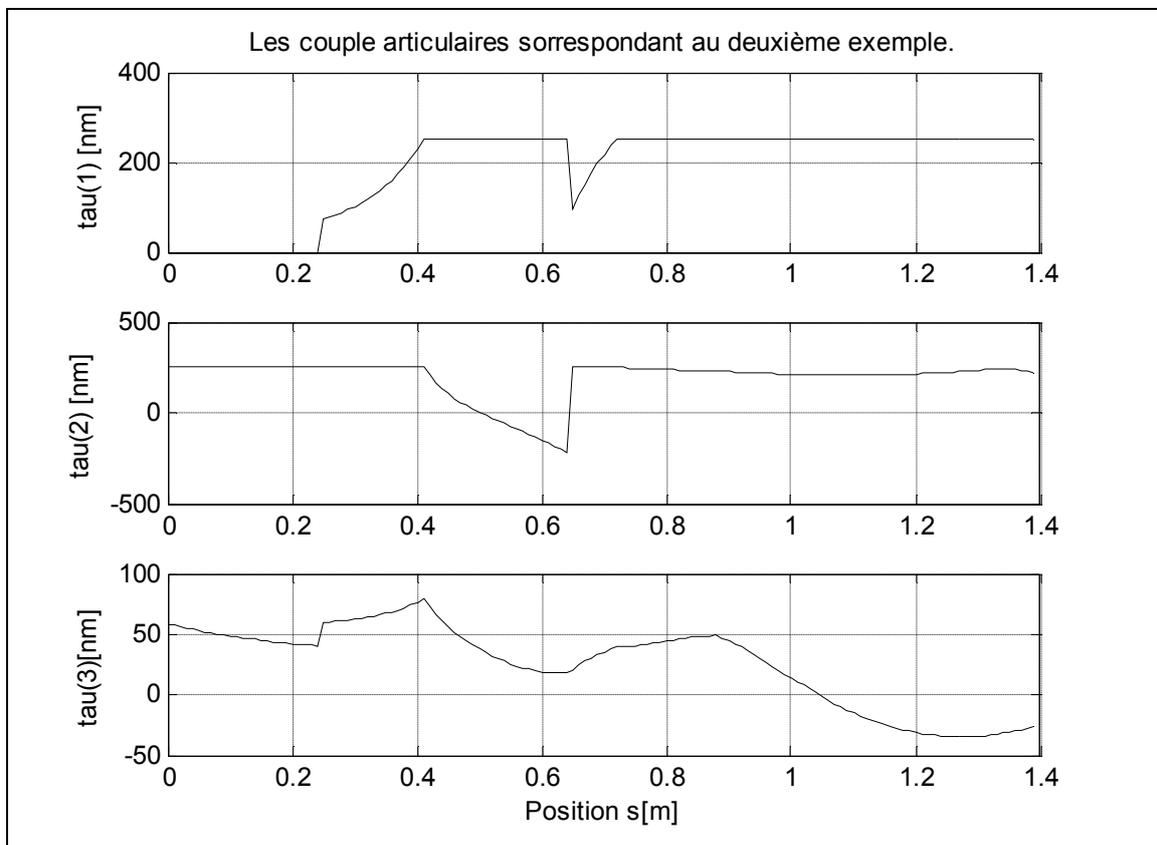


Figure -11- Couples articulaires correspondants au chemin du deuxième exemple.

Le temps optimal est  $t = 0.7810$ . La même remarque concernant les couples articulaires reste valable ici, à chaque instant au moins l'un des couples prend la valeur maximale.

### III.4.5. Résultats pour le troisième chemin considéré :

La trajectoire que nous allons considérer ici est composée de portions de droites et de cercles. Son allure est donnée par la figure ci-dessous.

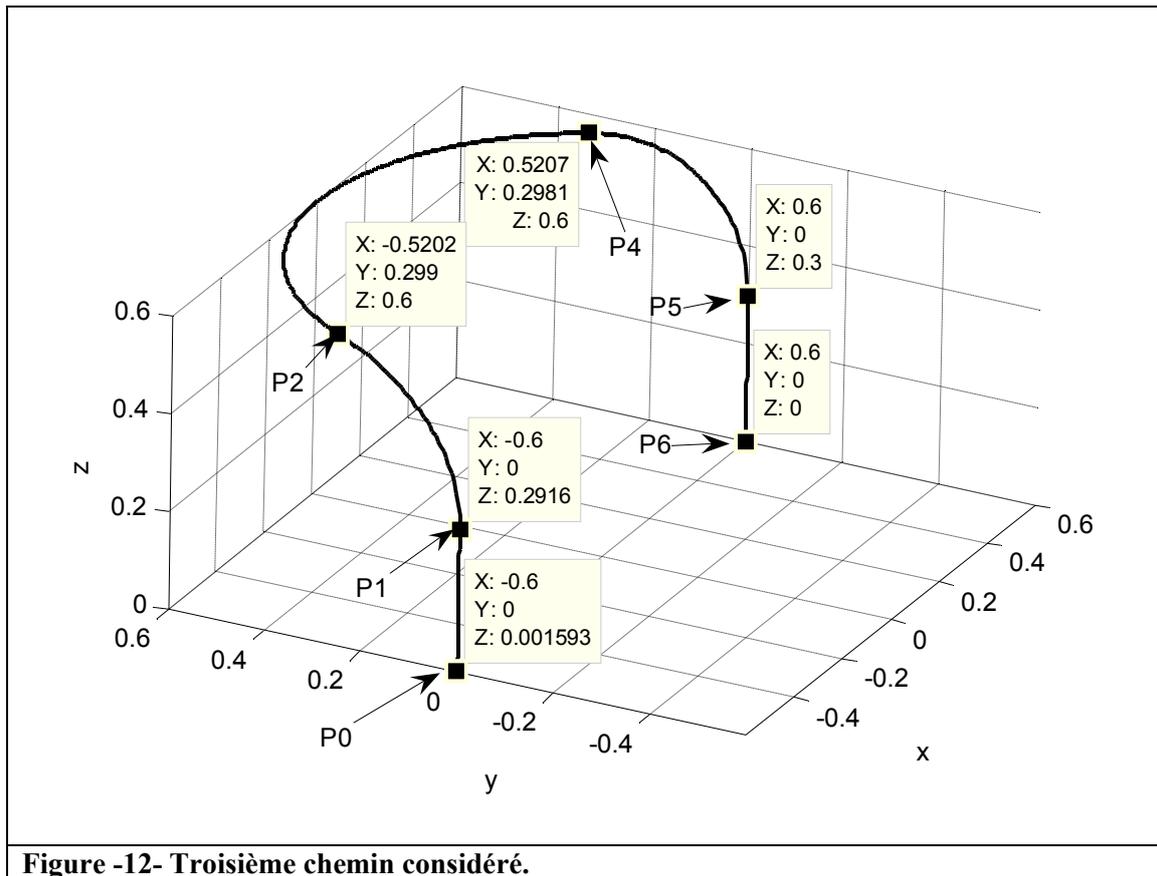


Figure -12- Troisième chemin considéré.

Les équations paramétriques de ce chemin en fonction d'un certain paramètre réel  $u$  sont comme suit :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cdot \\ \cdot \\ \cdot \end{pmatrix} \text{ tel que : } \begin{pmatrix} \cdot \\ \cdot \\ \cdot \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0.6 \cos(\cdot) \\ -0.6 \sin(\cdot) \\ 0 \end{pmatrix} \quad \begin{matrix} 0 \leq \cdot \leq 0.3 \\ 0.3 \leq \cdot \leq 0.6 \end{matrix} \quad \text{(III.44)}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0.6 \sin(\cdot) \\ 0 \\ 0 \end{pmatrix} \quad \begin{matrix} 0 \leq \cdot \leq 0.3 \\ 0.3 \leq \cdot \leq 0.6 \end{matrix} \quad \text{(III.45)}$$

$$\left( \begin{array}{c} \\ \\ \\ \end{array} \right) = \left\{ \begin{array}{l} \frac{0.3 + 0.3 - ( ) - 0.3}{0.6} \leq \leq 0.3 + \\ \frac{0.3 + 0.3 - ( ) - 0.3}{(0.6 + - )} \leq \leq (0.3 + ) \\ \frac{0.3 + 0.3 - ( ) - 0.3}{(0.6 + - )} \leq \leq (0.3 + ) \end{array} \right. \quad (III.46)$$

Si on examine le premier dérivé — le long de ce chemin, on remarque qu’il représente quelques discontinuités à certaines positions, ce qui donne des valeurs infinies pour certaines composantes de — en ces positions. Pour éviter ce problème, nous avons utilisé le B-spline pour obtenir une fonction lisse, d’ordre au moins , qui approxime cette trajectoire en fonction du même paramètre u. Le traitement reste inchangé, car on a besoin seulement d’avoir en disposition une façon de calculer numériquement la position de l’outil et de ses dérivés en fonction de u, mais on n’est pas obligé d’avoir des expressions formelles de ceux-ci.

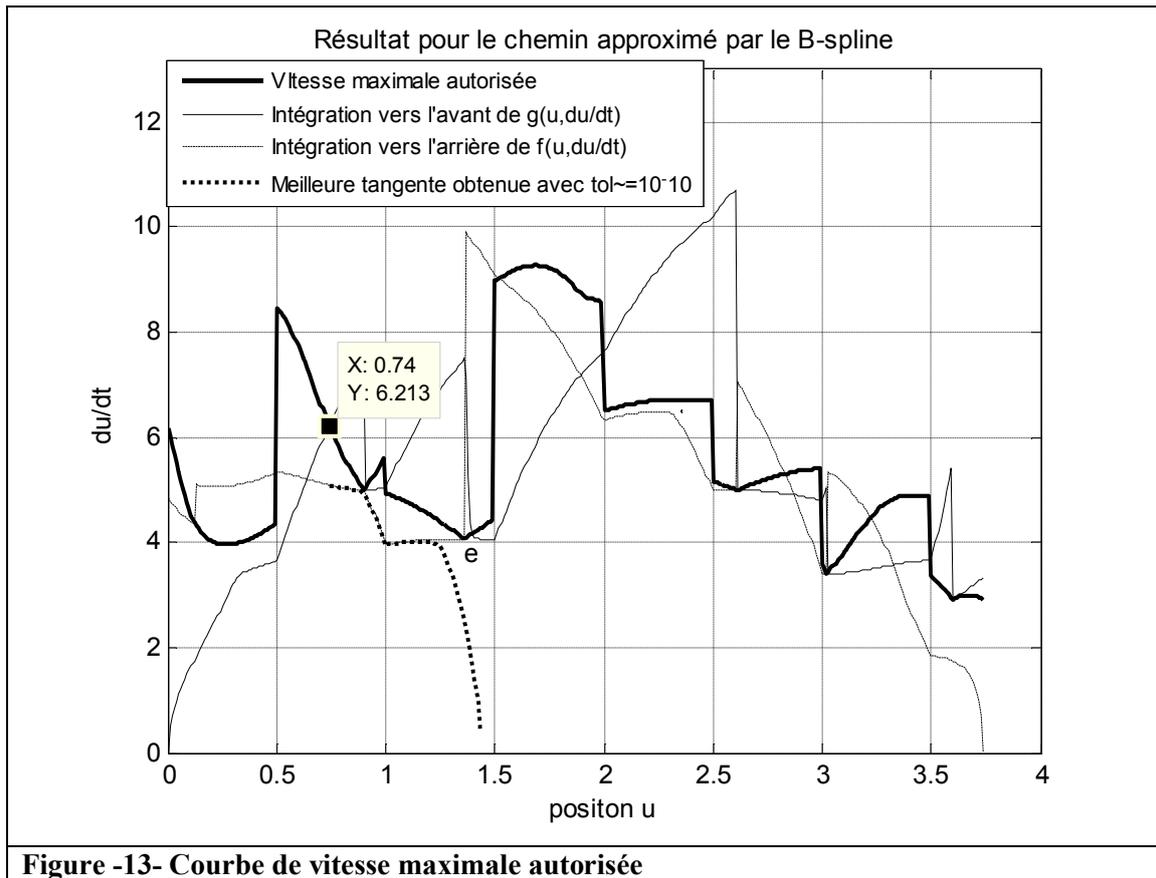
Seulement, il faut noter que l’utilisation d’une expression mathématique au lieu d’une approximation par le B-spline représente l’avantage d’un temps de calcul de la trajectoire optimal sensiblement inférieur. Ainsi il vaut mieux choisir une trajectoire lisse au départ pour ne pas être obligé d’utiliser le spline.

Les limites sur les couples que peuvent fournir les actionneurs sont sous la forme :

$$= -0.01 \dot{\phantom{u}} + 400 \quad ; \quad = -0.01 \dot{\phantom{u}} - 400$$

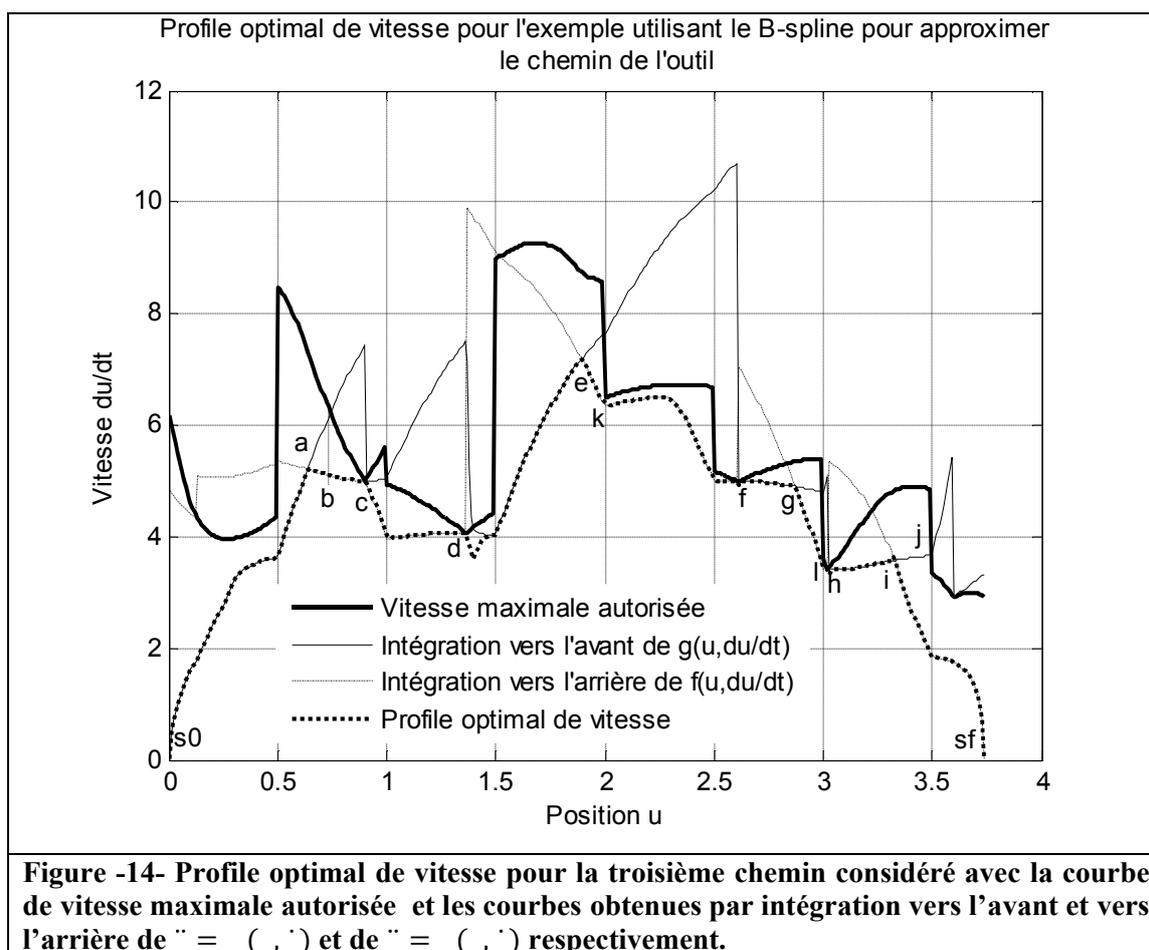
La courbe de vitesse maximale autorisée que nous avons obtenue pour cet exemple est représentée par la figure -14-.

Ensuite, nous avons procéder au calcul du profile optimal de vitesse selon l’algorithme décrit précédemment. Le résultat obtenu est donné dans la figure -15-. A ce point des remarques importantes doivent être faites :



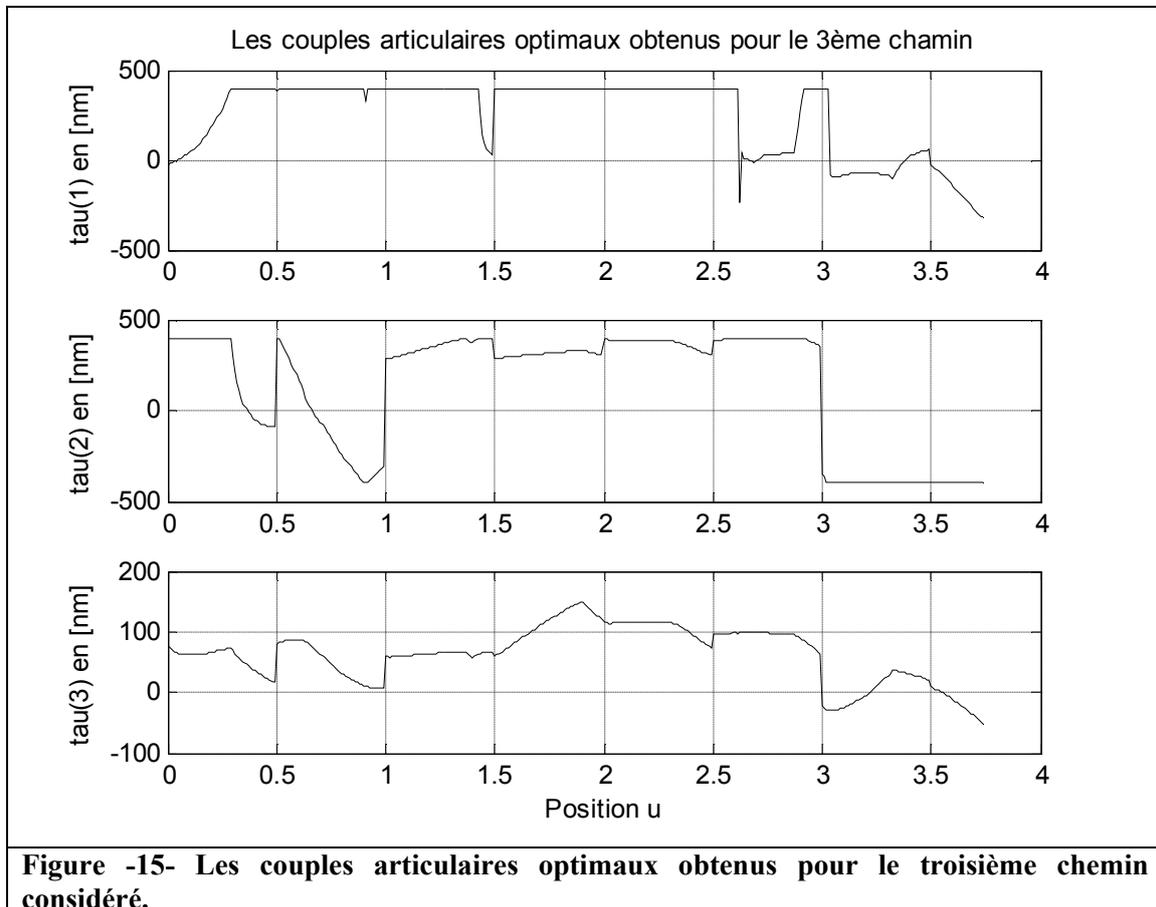
1. A partir des applications (effectuées durant ce travail) de cet algorithme, le profile de vitesse optimale est composé de portions très proches aux courbes obtenues par intégration vers l'avant de  $\ddot{u} = \dots$  et vers l'arrière de  $\ddot{u} = \dots$  avec seulement une petite différence. Ceci constitue un grand atout pour cet algorithme, car on peut détecter la position des points de tangence  $(u, v)$ ,  $(u, v)$  (figure -15-) à partir de ces courbes, ce que nous allons faire dans les points suivants.
2. Les points de tangence sont très voisins des points auxquels la courbe obtenue par intégration vers l'arrière de  $\ddot{u} = \dots$  coupe la courbe de vitesse maximale autorisée (ici c'est les points d, f et h sur la figure -15-).
3. Les positions des points de tangence  $(u, v)$  étant détectées de cette façon, alors au lieu de calculer  $(\dot{u}, \dot{u})$  en procédant à la résolution du problème à deux valeurs aux limites à partir des positions des points d'intersection de la courbe obtenue par intégration vers l'avant de  $\ddot{u} = \dots$  avec la courbe de vitesse maximale (les points b, k et l sur la figure -15-) qui sont généralement loin des points de tangence, il faut procéder autrement pour éviter que le cumul des erreurs d'intégration empêche la résolution efficace de ce problème. Un problème que nous

avons rencontré pour le calcul du premier point de tangence dans cet exemple, en partant du point b figure -15- et en faisant la recherche par intégration vers l'avant de  $\ddot{u} = ( \cdot )$ . L'algorithme de bisection donne la courbe représentée en pointillés sur la figure -14-, avec une tolérance relative sur  $\dot{u}$  de l'ordre de  $10^{-4}$ . En plus, l'exécution d'itérations supplémentaires ne conduit pas à une amélioration sensible. Cependant la courbe obtenue est loin d'être la tangente désirée. Alors au lieu de faire ceci nous avons procédé comme suit : Puisque ce qui importe est de trouver une courbe tangente et pas la façon de la trouver, alors, on peut commencer la recherche de la tangente au point qui précède juste l'intersection de la courbe  $f(u, \dot{u})$  avec la courbe de vitesse maximale autorisée (c'est à dire points d, f et h sur la figure -15-). Une fois une valeur de  $\dot{u}$  (respectivement  $\dot{u}$  et  $\dot{u}$ ) qui donne une courbe tangente à la courbe de vitesse maximale autorisée est obtenue, on procède au calcul du reste de la courbe par intégration vers l'arrière de  $\ddot{u} = ( \cdot )$  à partir du point  $(u, \dot{u})$  (respect  $(u, \dot{u})$  et  $(u, \dot{u}_h)$ ) jusqu'à la position  $u$  (respect  $u$  et  $u$ ). Cette façon de procéder est nettement meilleure en termes de précision et en termes de temps de calcul.



Les couples articulaires, pour ce troisième chemin, sont donnés par la figure -16-. La même remarque peut être faite, c'est que à chaque instant au moins l'un des trois actionneurs est entrain de fournir le couple maximum ceci est dû au fait qu'il faut exécuter la trajectoire en un temps optimal.

La valeur du temps optimale pour ce cas est  $t_{opt} = 1.0474$  s.



### III.5. Conclusion :

Dans ce chapitre nous avons appliqué un algorithme de planification de trajectoire à temps optimal à un robot à trois degré de liberté. L'algorithme nécessite seulement la connaissance d'une représentation paramétrique du chemin que doit suivre l'outil ou une façon de la calculer numériquement ainsi que le modèle du robot. La représentation paramétrique du chemin que doit suivre l'outil doit être lisse au moins d'ordre 3. Ce chemin doit éviter les singularités du robot. Si une expression mathématique du chemin de l'outil n'est pas disponible alors on peut l'approximer en utilisant le B-spline.

# Conclusion générale

---

Dans ce travail, nous avons traité le problème de planification de trajectoire à temps optimal avec chemin contraint pour un robot à trois degrés de liberté. Nous avons mis au point un programme qui permet de calculer les profils des couples articulaires qui permettent de déplacer l'outil du robot d'un point initial vers un point final selon un chemin défini est ceci dans un temps optimal.

L'algorithme implémenté nécessite que :

1. Le chemin de l'outil y compris son orientation soit spécifié.
2. Les angles articulaires peuvent être calculés en termes de la position sur le chemin de l'outil.
3. Les équations dynamiques du robot sont disponibles
4. Les valeurs maximales et minimales des couples qui peuvent être produits par les actionneurs sont connus comme fonction des angles et des vitesses articulaires.

L'algorithme commence par trouver la courbe de commutation entre l'accélération et la décélération maximales dans le plan de phase, ensuite calcul les couples qui en résultent.

On remarque que pour les couples optimaux et à un instant donnée, aux moins un actionneur est entrain de fournir le couple maximum, ceci s'explique par le faite que le robot doit faire la trajectoire spécifiée en un temps minimale.

Nous avons constaté, aussi, que les profils obtenus des couples articulaires présentent des points de discontinuités. En pratique, les dynamiques des robots rendent impossible l'implémentation de ces couples. Pour un moteur à courant continu typique la constante de temps entre la tension d'entrée et le couple mécanique fourni par le moteur va de 0.0001sec à 0.025sec [20], ces valeurs sont, en général, assez petites pour que l'effet de ces dynamiques soit négligé. Si ce n'est pas le cas un contrôleur additionnel doit être incorporé pour obtenir les couples désirés.

## **Résumé**

Dans ce travail, nous avons fait l'étude et la programmation d'un algorithme de planification de trajectoire à temps optimale. Le chemin géométrique de l'outil est spécifié au départ et on procède au calcul de l'évolution dans le temps de l'outil sur ce chemin. L'idée de base sur laquelle est fondé cet algorithme est que pour une configuration donnée il y a toujours une vitesse limite maximale à partir de laquelle les actionneurs seront incapables de maintenir l'outil sur la trajectoire spécifiée, c'est-à-dire seront incapables de générer les couples nécessaires. Alors en se basant sur cette idée, on procède au calcul de l'accélération nécessaire pour chaque position, ensuite on utilise le profile obtenu pour le calcul de la trajectoire.

Nous avons appliqué cet algorithme à un robot à trois degrés de liberté et un ensemble de chemins typiques de l'outil. Les résultats obtenus montrent la simplicité de la méthode et conduisent à une trajectoire à temps optimale.

Notre présente étude commence par livrer des généralités et des définitions spécifiques au domaine de la robotique. Ensuite nous donnons une brève présentation de la cinématique et de la dynamique des robots manipulateurs, ainsi que l'explication des diverses méthodes de génération de la trajectoire.

Nous enchaînons au deuxième chapitre par un rappel sur la commande optimale en général et sur l'optimisation paramétrique.

Au dernier chapitre, sont présentés les différents calculs effectués pour appliquer cette commande à un exemple (robot à trois degrés de liberté) ainsi que les résultats de simulation.

Enfin, nous clôturons ce mémoire avec une conclusion générale.

## BIBLIOGRAPHIE

- [1] **Richard . P. Paul**, « robot manipulator Mathematics, programming, and Control » MIT Press, Cambridge, 1981
- [2] **Wisama.K, Dombre. E** « Modélisation Identification et Commande des robots , » 2<sup>e</sup> édition, Hermes Science Publications, Paris, 1999
- [3] **Craig. J**, « introduction to robotics , mechanics and control», Troisième Edition, prentice Hall 2005.
- [4] **Bobrow,J.E.,Dudowsky,S. end Gibson, J.S.** (San Francisco),On the optimal control of robotic manipulators with actuator constraints. » Proc.American control conference 2:782-787. 1983
- [5] **Benaissa. S, Hamache. A** « Commande en position-orientation d'un robot manipulateur dans l'espace opérationnel. » Mémoire d'ingénieur, Ecole nationale polytechnique, 2004.
- [6] **Abderrahmani. A, Zerguini. Y** « Commande floue robuste appliquée à un bras de robot manipulateur »,Mémoire d'ingénieur, Université A. Mira Bejaïa, Algérie, 2001 .
- [8] **Azoui.N**, « commande non linéaire d'un bras manipulateur » Mémoire de magister en électronique Option Robotique, université de Batna, 2008 /2009.
- [9] **Liegeois. A**, «Modélisation et commande des robots manipulateurs » Techniques de l'ingénieur, Université de Montpellier, référence S77730
- [10] **M, Zennache**« Commande neuro-floue d'un bras manipulateur» Mémoire d'ingénieur, Université A. Mira Bejaïa, Algérie, 2000.
- [11] **Wisama.K** ;« Commande des robots manipulateurs », Hermes Science Publications, Paris, 2002.
- [12] **Dombre. E** ; « Analyse et Modélisation des robots manipulateurs », Hermes Science Publications, Paris, 2001. Paris, 2002.
- [13] **Abdelkader Benmisra**« Programmation des robots industriel et application sur le robot manipulateur», Mémoire de Magistère, Université de Saad Dahleb de Blida, Algérie, 2007
- [14] **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.
- [15] **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.
- [16] **Emmanuel Trélat**, « commande optimale», Note du cours, 2007/2008.
- [17] **Roux Patrick**, « Nombre aléatoires et optimisation non linéaire», Mémoire d'ingénieur, école polytechnique fédérale de Lausanne, 2003/2004.
- [18] **J.E. Bobrow, S. Dubowsky end J.S Gibson** «The International Journal of Robotics Research Time-Optimal Control of Robotic Manipulators Along Specified Paths» 1985
- [19] **Bobrow,J.E.,Dudowsky,S. end Gibson, J.S** «On the optimal control of robotic manipulators with actuator constraints. » *Proc.American control conference 2:782-787.* 1983
- [20] **Dubowsky, S. and Shiller, Z.** »Optimal dynamic trajectories for robotic manipulators ». Proc. V CISM-IFTOMM Symp. «Theory and Practice of Robots and Manipulators. » 1984
- [21] **Bobrow, J. E.** « Optimal control of robotic manipulators. » Ph.D. dissertation, University of California, Los Angeles 1982