

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



Faculté de Technologie
Département d'Automatique,
Télécommunication et d'Electronique

Mémoire fin d'études

Pour l'obtention du diplôme de Master en Automatique
Spécialité : Automatique et système

Thème

**Planification de trajectoire a temps optimal pour
le robot SCARA**

Réalisé par :

M^{elle} DJAFRI Leila

Encadré par :

M^r HADDAR Hocine

Membre de jury :

M^r LEHOUCHE .H

M^{eme} IDJDARENE.

Année Universitaire : 2017/2018

Remerciements

En préambule à ce mémoire

Je tiens remercier ALLAH qui m'a aidé et qui m'a donné la patience et le courage durant ces années d'études.

Je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Je tiens à remercier sincèrement mon promoteur Mr Haddar H de m'avoir fait l'honneur d'assurer l'encadrement de mon travail et pour la confiance qu'il a témoigné, pour sa disponibilité, ses efforts et ses encouragements qui m'ont permis de mener à bien ce travail

Je tiens à remercier les membres de jury d'avoir consacré de leur temps à la lecture de ce manuscrit et d'accepter et d'évaluer ce travail

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.

Merci à tous

Dédicace

Je dédie ce modeste travail :

À ma grande mère Baya

À ma mère et à mon père

À mes oncles et leur famille

À mes chères sœurs et frères

*À mon promoteur Mr Haddar .H pour son soutien et ses conseils
les plus précieux*

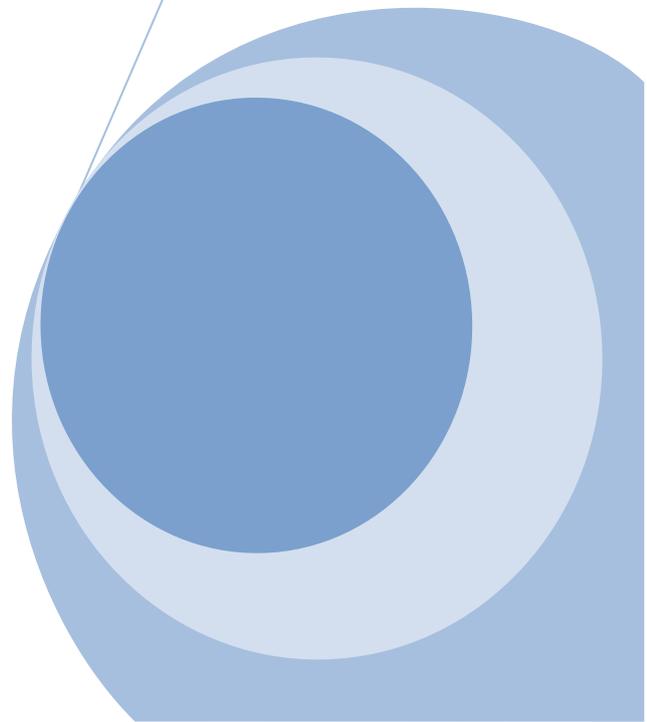
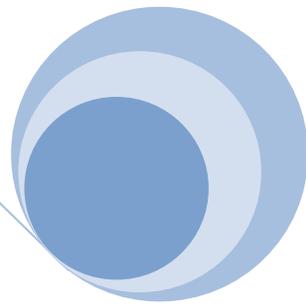
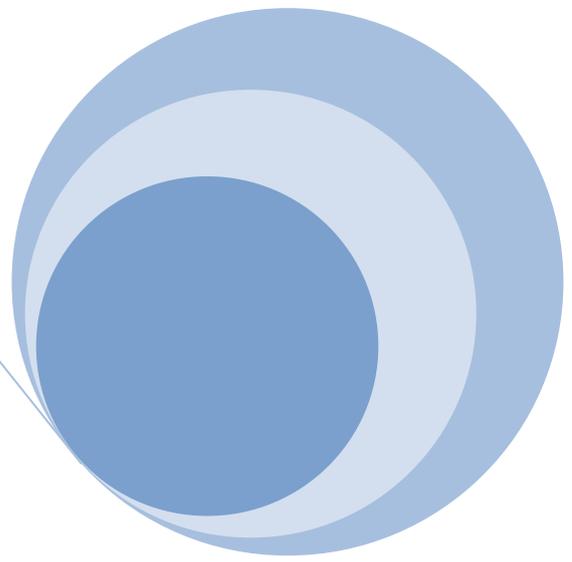
À mes chères amies

À ceux qui me sont chers

*À toute la promotion deuxième année Master Automatique et
système et informatique industriel 2017/2018.*

Leila

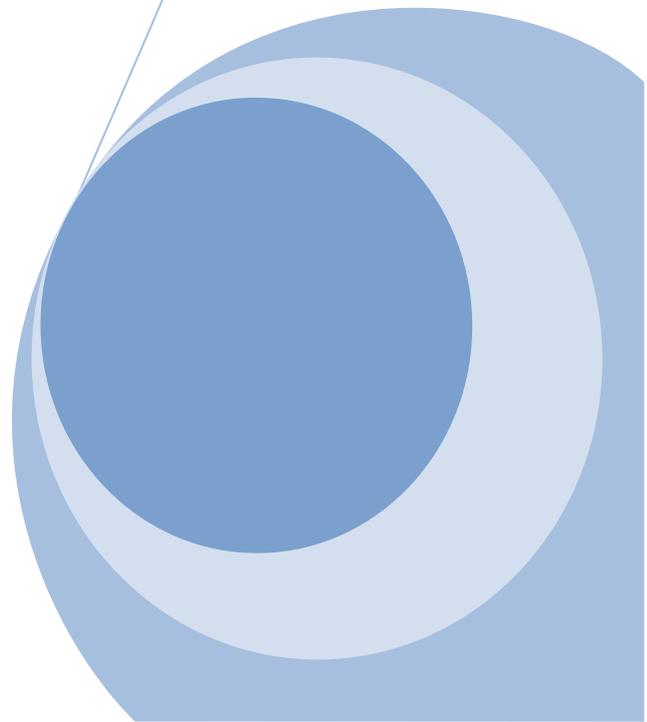
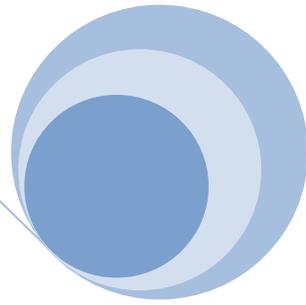
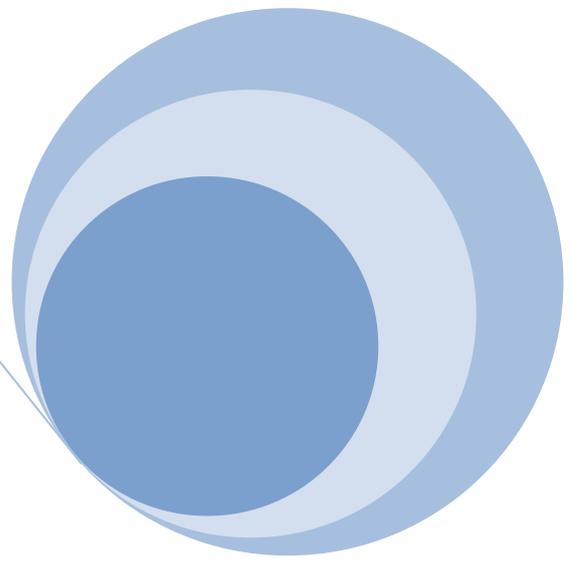
Liste des figures



Liste des figures

Figure I.1	Structure SCARA.....	4
Figure I.2	Représentation d'une articulation rotoïdes	5
Figure I.3	Représentation d'une articulation prismatique	5
Figure I.4	Morphologies possibles des robots	7
Figure I.5	Représentation d'un point.....	8
Figure I.6	Translation pure d'un repère R_f par rapport à un repère R_i	10
Figure I.7	Modèle géométrique directe et inverse	13
Figure II.1	Planification du chemin.....	29
Figure II.2	Méthodes par décomposition cellulaire (méthode exacte) et constitution du graphe.....	31
Figure II.3	Méthode par décomposition cellulaire (méthode approchée)...	31
Figure III.1	Spécification du chemin de l'organe terminal.....	36
Figure III.2	Robot SCARA	47
Figure III.3	Courbe de vitesse maximale et les points de commutation $\alpha \rightarrow \beta$ pour la trajectoire considérée.....	50
Figure III.4	Evolution des paramètres d'inertie le long de la MVC.....	51
Figure III.5	Courbes obtenues par intégration directe de $\beta_{min}(s, \dot{s})$ et intégration vers l'arrière de $\alpha_{max}(s, \dot{s})$ en plus de la MVC (Maximum Velocity Curve) et les points de commutations.....	52
Figure III.6	Profile de vitesse à temps optimal obtenu après la résolution du problème de planification.....	54
Figure III.7	Couples articulaires du robot SCARA le long de la trajectoire spécifié	55
Figure III.8	Couples articulaires des trois articulations exprimés en fonction du temps.....	56
Figure III.9	Coordonnées articulaires le long de la trajectoire optimale...	57

Liste des abréviations



La liste des abréviations

CO: Commande optimale.

CSA: Concept d'espace des configurations.

DH : Convention de Denavit-Hartenberg.

d.d.l: degré de liberté.

HJB: Hamilton-jacobéen-Bellman.

ISO: système international.

LQ: Linéaire quadratique.

MG: Modélisation géométrique.

MGD : Modèle géométrique direct.

MGI : Modèle cinématique inverse.

MDH: Convention de Denavit-Hartenberg modifiée.

MC: Modèle cinématique.

MCD: Modèle cinématique direct.

MCI: Modèle cinématique inverse.

MD: Modèle dynamique.

MDI: Modèle dynamique inverse.

MVC: La courbe de vitesse maximale (Maximum Velocity Curve).

NLTPBP: A non linear two-point boundary problem.

PPP: Morphologies (prismatiques, prismatiques, prismatiques).

PRP: Morphologies (prismatiques, rotoïdes, prismatiques).

PPR: Morphologies (prismatiques, rotoïdes, prismatiques).

RUR: les robots universels de Rossum.

RRR: Morphologies (rotoïdes, rotoïdes, rotoïdes).

RPR: Morphologies (rotoïdes, prismatiques, rotoïdes).

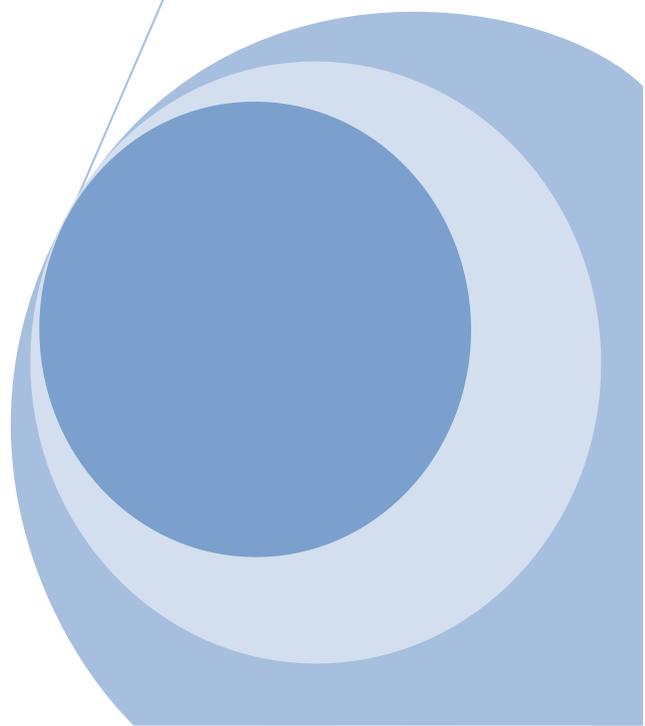
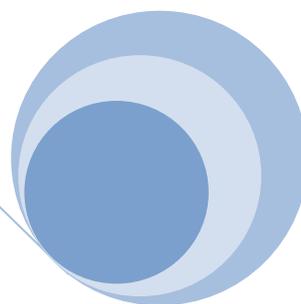
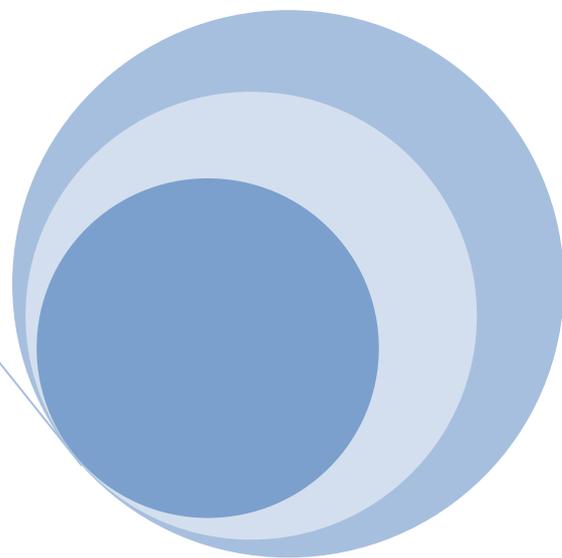
RRP: Morphologies (rotoïdes, rotoïdes, prismatiques).

RPP: Morphologies (rotoïdes, prismatiques, prismatiques).

SCARA: Selective Compliance Articulated Robot Arm.

TM: Problème à temps minimal.

Liste des notations



Liste des notations

c_i	Corps de robots
m	Mobilité de l'articulation ou nombre de degrés de liberté résultant
R	Articulation rotôdes
P	Articulations prismatiques
(P_x, P_y, P_x)	Coordonnées cartésiennes d'un point
W	Facteur d'échelle
(u, v)	Vecteur unitaire
$((u_x, u_y, u_z), v)$	Coordonne d'un vecteur unitaire
(Rot_x, Rot_y, Rot_z)	Matrice des rotations
\det	Déterminant d'une matrice
(X, Y)	Variable opérationnelles
T	Matrice de transformation homogène
$Trans(a, b, c)$	Matrice de translation pure
$\theta, (q)$	Vecteur des variables articulaires
(J_r, J)	Matrice Jacobéenne
Γ	Vecteur des couples / Force des actionnaires
f_e	Vecteur représente les efforts extérieurs
i	Indice
τ	Formule de Lagrange Euler
m_i	masse
F_i	Forces
${}^{ci}I_i$	Tenseur d'inertie ou centre de masse
$({}^i n_i, {}^i f_i)$	Forces et couples d'inertie
M	Matrice de masse et inerties
C	Matrice des termes centrifuges et de Coriolis
G	Vecteur de termes de gravité
$(q, \text{ou } \theta)$	Position articulaire
$(\dot{q}, \text{ou } \dot{\theta})$	Vitesse articulaire
$(\ddot{q}, \text{ou } \ddot{\theta})$	Accélération articulaire
s	Paramètre de distance le long de chemin
\dot{s}	La première dérivée de la distance du point de la position initiale
\ddot{s}	la deuxième dérivée de la distance du point de la position initiale
P_o	Position initial
P_f	Position final
t	Temps
$(N_i \text{ ou } C \text{ ou } \tau)$	Le couple
r	Vecteur de position de l'outil
\dot{r}	Différentiation par rapport au temps
t_f	Position du centre de masse
q_{goal}	point de destination
$\alpha_i(s, \dot{s})$	Accélération minimale
$\beta_i(s, \dot{s})$	Accélération maximale
$\tau_{\min}(\theta, \dot{\theta})$	Le couple minimale
$\tau_{\max}(\theta, \dot{\theta})$	Le couple maximale

SOMMAIRE

Liste des figures.....	
Liste des abréviations.....	
Liste des notations.....	
Introduction générale	1

Chapitre I

Concepts de base sur la robotique

I.1-Introduction	3
I.2 Généralités	3
I.2.1 Historique.....	3
I.2.2 Définition d'un robot.....	3
I.2.3 Constituants mécaniques du robot manipulateur.....	4
I.2.4 Les différents types des robots.....	6
I.2.4.1 Robots mobiles	6
I.2.4.2. Robots manipulateurs.....	6
I.2.5 Morphologie des robots manipulateurs.....	6
I.2.6 Les caractéristiques d'un robot.....	7
I.2.7 Orientations et matrice de rotations.....	8
I.2.7.1 Représentation d'un point dans l'espace.....	8
I.2.7.2 Représentation d'une direction.....	8
I.2.7.3 Matrice de rotation	9
I.2.7.4 Matrice de translation	9
I.2.7.5 Translation homogènes (matrice 4*4).....	10
I.3 Modélisation des robots.....	10
I.3.1 Modélisation géométrique (MG).....	10
I.3.1.1 Modèle géométrique direct(MGD).....	10
I.3.1.2 Modèle géométrique inverse (MGI).....	13
I.3.2 Modèle cinématique(MC).....	13
I.3.2.1 Modèle cinématique directe(MCD).....	14
I.3.2.2 Modèle cinématique inverse (MCI)	14
I.3.3 Modèle dynamique (MD).....	15
I.3.3.1 Modèle dynamique inverse (MDI).....	15
I.3.3.2 Modèle dynamique directe(MDD).....	19
I.4 Conclusion	20

Chapitre II

Commande optimale et planification de trajectoire

II.1 Introduction.....	21
II.2 La commande optimale.....	21
II.2.1 Définitions	21
II.2.2 Objectifs de la commande optimale.....	21
II.2.3 Des notions sur la commande optimale.....	21
II.2.3.1 Les contraintes de la commande optimale.....	22

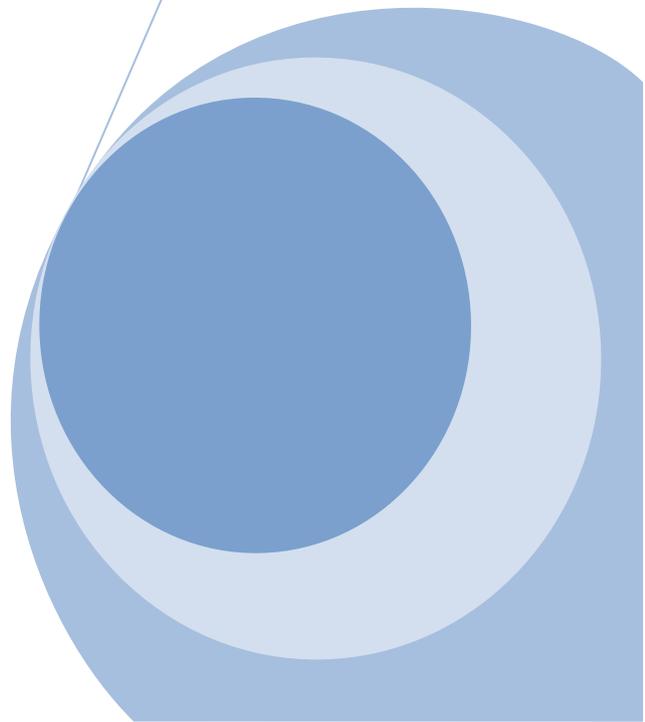
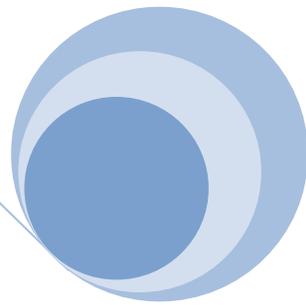
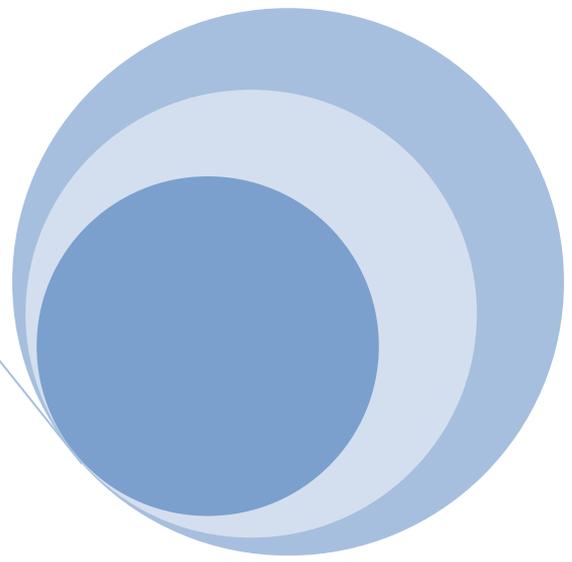
II.2.4 Les méthodes de la résolution de problème de commande optimale.....	22
II.2.4.1 Les méthodes indirectes	23
II.2.4.1.1 Multiplication de Lagrange (équation d'Euler_Lagrange).....	23
II.2.4.1.2 Le principe de Pontryagin.....	23
II.2.4.1.3 Méthode en boucle fermé.....	24
II.2.4.1.4 Méthode en boucle ouverte	24
II.2.4.2 Les méthodes directes	25
II.2.4.2.1 La méthode de discrétisation	25
II.3.4.1.1 La méthode de paramétrisation	26
II.3 Planification de trajectoire	27
II.3.1 Définition	27
II.3.2 Les différents types de planification de trajectoire	28
II.3.2.1 Planification de mouvements fins	28
II.3.2.2 Planification de chemin (trajectoire).....	28
II.3.3 Méthodes de planification de trajectoire (chemin).....	29
II.3.3.1 Les méthodes globales	29
II.3.3.1.1 Méthode par décomposition cellulaire	29
II.3.3.1.2 Méthodes de résolution de type rétraction (Roadmaps).....	32
II.3.3.1 Les méthodes locales	32
II.3.3.2.1 Méthode du potentiel.....	33
II.3.3.2.2 La méthode des contraintes	34
II.4 Conclusion	34

Chapitre III Planification de trajectoire à temps optimal pour le robot SCARA

III.1 Introduction	35
III.2 Formulation du problème de commande a temps optimale.....	35
III.2.1 Le chemin définit	35
III.3 Etapes de calcul des vitesses ($\dot{\theta}(s, \dot{s})$) et accélérations ($\ddot{\theta}(s, \dot{s}, \ddot{s})$).....	38
III.3.1 Calcul des vitesses ($\dot{\theta}(s, \dot{s})$) articulaires	38
III.3.2 Calcul d'accélération ($\ddot{\theta}(s, \dot{s}, \ddot{s})$) articulaires.....	38
III.3.3 Les expressions de l'accélération minimale et maximale	39
III.4 Solution du problème de paramétrisation a temps optimal du chemin	42
III.5 Résultats de simulation	46
III.6 Conclusion.....	57
Conclusion générale.....	58

Bibliographie

Introduction générale



Introduction générale

Pour qu'un robot puisse accomplir une tâche de haut niveau donnée, telle que la saisie d'un objet à partir d'un convoyeur et son emplacement sur une étagère, cette tâche doit être interprétée sous forme de commande de bas niveau que le système d'exploitation du robot peut comprendre. Ce processus global connu sous le nom de planification de mouvement peut être décomposé en un nombre d'étapes allant de la définition de la tâche dans l'espace de travail et son interprétation dans l'espace de configurations sous forme d'un chemin qui définit l'ensemble de configurations que doit visiter le robot pour réaliser la tâche désirée sans entrer en collision ni avec des obstacles ni avec lui-même. Cette étape s'appelle planification de chemin et ne prend en considération que les contraintes géométriques spécifiées. Ensuite, vient l'étape de paramétrisation de ce chemin avec le temps. Ici on détermine la variation des variables articulaires avec le temps en prenant en considération les limites des vitesses articulaires, des couples articulaires, la minimisation d'énergie ou bien la minimisation du temps de traversée. Cette étape s'appelle planification de trajectoires. Après vient l'étape de conversion de la trajectoire en un ensemble d'instructions dans le langage du robot. Cette étape s'appelle synthèse d'instructions. Elle est suivie de l'étape de synthèse des commandes qui fait la conversion de ces instructions en des commandes de bas niveau telles que les courants électriques appliqués aux moteurs.

Dans ce travail nous traiterons le problème de la planification de trajectoires. En partant d'un chemin prés-spécifié qui détermine la position et l'orientation de l'outil le long de la tâche on calcul la trajectoire dans l'espace articulaire. Plusieurs techniques existent pour accomplir ceci. Une grande classe de ces techniques utilise la commande optimale pour résoudre ce problème. Certaines méthodes utilisent la programmation dynamique et d'autres font la conversion du problème de planification en un problème de commande optimale qui minimise un certain critère qui est dans le cas le plus fréquent le temps de traversée. Dans certaines méthodes, ce problème de commande optimale est ensuite converti en un problème d'optimisation est résolu par les méthodes d'optimisation avec contraintes.

La méthode que nous traiterons dans ce travail utilise le principe que la solution à temps optimale est Bang-Bang pour résumer la recherche de la solution à temps optimal en la recherche de l'ensemble de points de commutations entre l'accélération maximale et l'accélération minimale le long d'une certaine courbe appelée la courbe de vitesse maximale

(MVC) qui détermine la limite du domaine dans lequel la solution optimale qui vérifie les contraintes sur le couple existe dans les références[12,14,15].

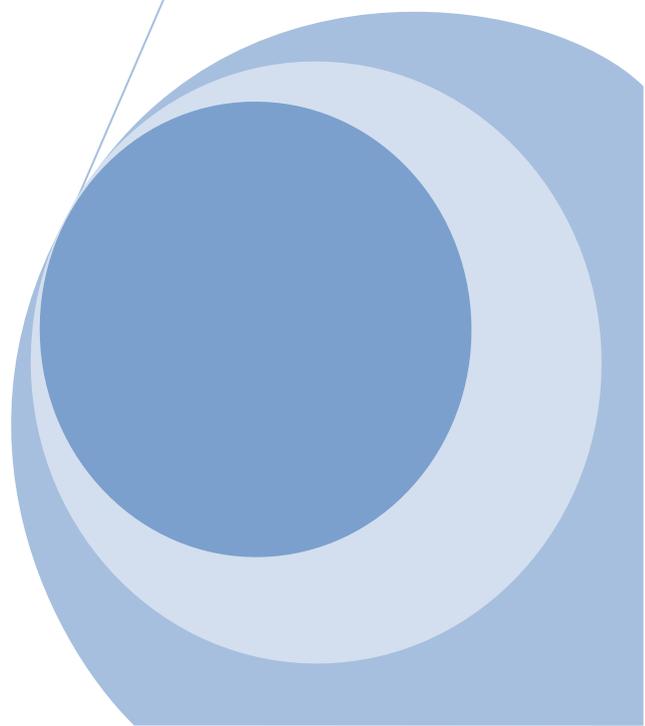
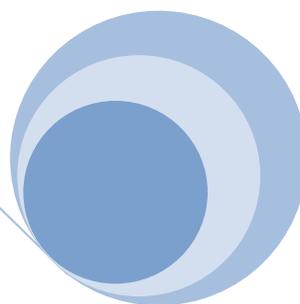
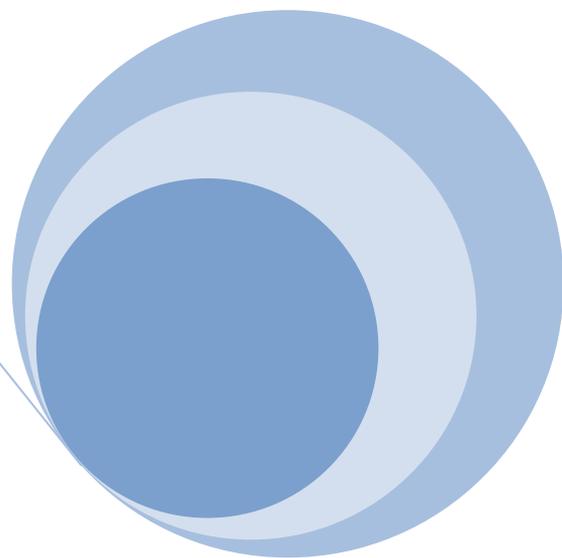
Dans le premier chapitre nous avons parlé sur des généralités de la robotique premièrement sur les définitions et historique des robots puis sur leur architecture (les constituants mécanique des robots), les différents type des robots puis leur morphologie, caractéristique, orientation et matrice de rotation finalisant par leur modélisations avec ces différents partie.

Dans le deuxième chapitre nous avons traités les deux parties l'une sur la commande optimale, nous avons parlé de son définition, objectif, notions et contraintes de cette commande, les méthodes de résolution comme méthode indirecte (Multiplicateur de Lagrange, principe de pontryagin, méthode en boucle ouvert, en boucle fermé) et les méthodes directes. L'autre partie est sur planification de trajectoire, nous avons cité les définitions, contraintes, les différents types et aussi les méthodes de planification de trajectoire.

Dans le troisième chapitre nous avons formulés le problème de commande a temps optimale a traités dans la simulation avec les divers étapes premièrement définition du chemin puis on transforme le problème a un problème mathématique dont la seul variable de commande et l'accélération maximale selon des contraintes sur les couple on calculant les accélérations minimale et maximale. Nous avons fait la solution du paramétrisation a temps optimal du chemin puis les résultats de simulation.

Nous avons mis au point un programme sous MATLAB qui calcul la trajectoire à temps optimale pour un robot SCARA qui satisfait les contraintes sur les couples articulaires. Nous avons appliqué le programme mis au point à un exemple de trajectoire et nous les résultats dans ce mémoire.

Concepts de base de la robotique



I.1 Introduction

Les robots manipulateurs sont conçus pour manipuler des objets. Dans ce chapitre nous allons faire une étude générale des concepts de base sur la robotique tel que : la représentation de la position et de l'orientation des objets dans l'espace, matrice de transformation homogène et de rotation, les définitions des principaux termes utilisés dans la robotique, cinématique, dynamique...

I.2 Généralités

I.2.1 Historique

Avant les années 50, les techniques liées à la robotique étaient trop primitives pour permettre des réalisations utiles. A la fin des années cinquante, Georges Devol invente une machine originale polyvalente et reprogrammable, ce qui a permis au robot d'acquérir une réalité industrielle. Ce robot, appelé Unimate, était capable de manipuler des objets avec des actionneurs hydrauliques.

En 1961, Le fabricant de voitures américain General Motors le premier à installer ce bras articulé dans l'une de ses usines. C'est à la fin des années 1970, que les robots industriels de première génération ont vu le jour. Les robots sont devenus tellement indispensables dans certains secteurs industriels (l'automobile par exemple), que leur emploi conditionne la survie économique des entreprises [1].

I.2.2 Définition d'un robot

Selon ISO un robot est un manipulateur multifonction commandé en position, reprogrammable, polyvalent, à plusieurs degrés de liberté capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvement variables et programmés pour l'exécution d'une variété de tâches de façon cyclique. Ils sont composés d'un ou plusieurs bras se terminant par un poignet [2].

Les robots manipulateurs contiennent plusieurs structures telles que :

- Structure ouverte simple.
- Structure arborescente.

- Structure fermée.
- Structure fermée simple.

I.2.3 Constituants mécaniques du robot manipulateur

Un robot sert à déplacer des objets et il demande une structure mécanique donnée par la (figure I.1) tel que :

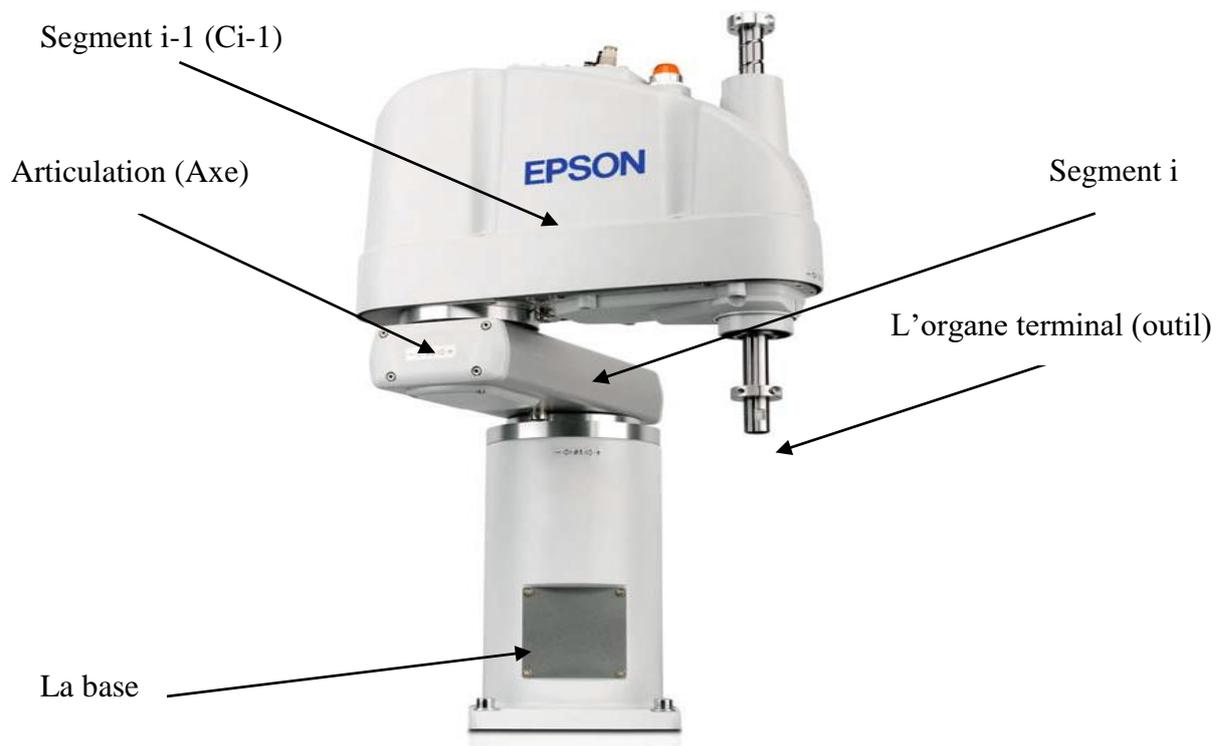


Figure I.1 Structure SCARA

- **La base** : est une surface qui supporte le poids du corps de bras manipulateur. Ceci est le même pour toutes les robots.
- **Segments** : Est un corps solides rigides peut être en mouvement par rapport à la base.
- **Articulations (Axes)** : Une articulation lie deux corps successifs en limitant le nombre de degrés de liberté de l'un par rapport à l'autre.

Si $m=1$, ce qui est le cas le plus fréquent en robotique l'articulation est soit rotoïdes, soit prismatique. « m : mobilité de l'articulation ou nombre de degrés de liberté résultant »

➤ **Articulations rotoïdes**

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

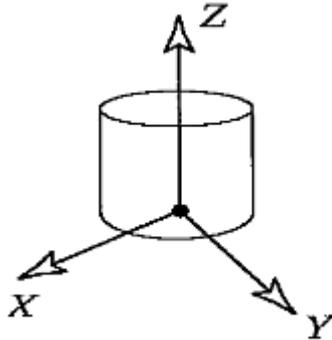


Figure I.2 Représentation d'une articulation rotoïdes.

➤ **Articulations prismatiques**

Il s'agit d'une articulation de type glissière réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe (figure I.3). On note P [2].

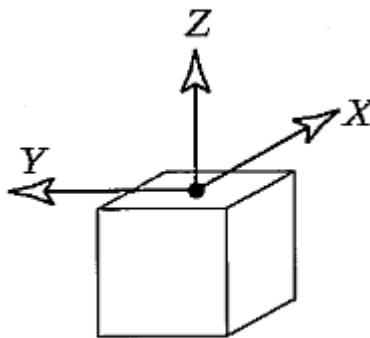


Figure I.3 Représentation d'une articulation prismatique.

- **L'organe terminal (effecteur) :** Il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multifonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable.

Un robot, enfin, peut-être multi-bras, chacun des bras portant un organe terminal différent. On utilisera indifféremment le terme organe terminal, préhenseur, outil ou effecteur pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique, exemple : pistolet pour la soudure dans les robots industriels.

I.2.4 Les différents types des robots

I.2.4.1 Robots mobiles

Les robots mobiles sont en constante évolution principalement depuis les années 2000, dans les domaines militaires (drones volants, robots sous-marins, . . .), le médical, agriculture et etc. Ils sont particulièrement demandés pour réaliser des tâches considérées comme pénibles, dangereuses ou impossibles pour les hommes.

Les robots mobiles sont des appareils automatiques capables d'effectuer certains travaux qui nécessitent le déplacement pour effectuer la tâche demandée par les programmeurs. En effet, le robot autonome doit être en mesure de prendre des décisions, quant aux mouvements à réaliser, en fonction des informations dont il dispose sur l'environnement ou lui parvenant de ses capteurs [3].

I.2.4.2 Robots manipulateurs

C'est une association d'un système mécanique articulé avec des organes différents, chacun assurant une certaine fonction. Le système ainsi obtenu doit réaliser des tâches plus ou moins diverses qu'on lui assignera à travers un algorithme.

Ils sont ancrés physiquement à leur place de travail et généralement mis en place pour réaliser une tâche précise ou répétitive. Ils sont très implantés sur les lignes d'assemblage dont le but est de prendre et de déplacer des objets (pièces ou outils). Ils ont plusieurs degrés de liberté, ils peuvent être commandés par un opérateur, une unité de commandes électronique ou un système logique [4].

I.2.5 Morphologie des robots manipulateurs

On ne considère que 2 paramètres : le type d'articulation (rotoïdes (R) ou prismatique (P)) et l'angle que font deux axes articulaires successifs (0° ou 90° ; sauf cas très particulier,

les axes consécutifs d'un robot sont soit parallèles, soit perpendiculaires). On convient d'appeler les 3 premiers d.d.l. le porteur du robot.

Les d.d.l. résiduels forment le poignet, caractérisé par des dimensions beaucoup plus petites et une plus faible masse. Les 12 morphologies possibles de porteur sont schématisées dans la figure qui suit (ces morphologies sont non redondantes (on élimine a priori les structures limitant les mouvements du porteur à des déplacements linéaires ou planaires : 3 liaisons prismatiques d'axes parallèles, par exemple, ou 3 liaisons rotoïdes d'axes parallèles)) donné par la (figure I.4) [5].

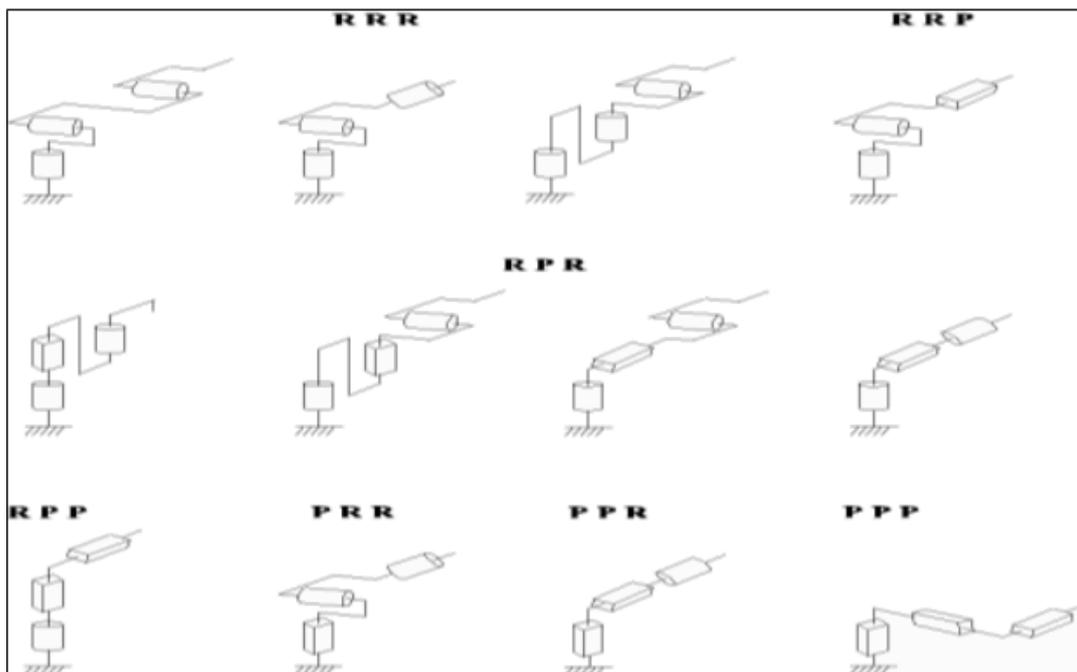


Figure I.4 Morphologies possibles des robots

I.2.6 Les caractéristiques d'un robot

La norme ISO (système internationale) 9946 spécifie les caractéristiques que les fabricants de robot doivent fournir parmi ces caractéristiques [2]:

- L'espace de travail : c'est l'espace que l'organe terminal peut atteindre.
- La charge utile ou charge maximale transportable par le robot.
- Les vitesses et accélérations maximale, qui conditionnent les temps de cycle.
- Les performances : les notions d'exactitude (écart entre une situation commandée et la moyenne des situations atteintes) et de répétabilité (dispersion des situations atteintes lorsque l'on commande successivement la même situation).

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

I.2.7 Orientations et matrice de rotations

I.2.7.1 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 $w \times p_x, w \times p_y, w \times p_z$ ou 'w' est un facteur d'échelle égale à 1 en robotique. On présente alors les coordonnées homogènes d'un point par le vecteur (figure I.5) :

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad (\text{I.1})$$

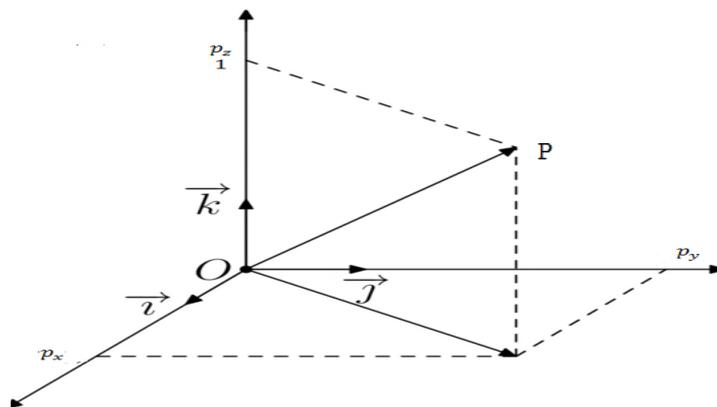


Figure I.5 Représentation d'un point.

I.2.7.2 Représentation d'une direction

La représentation d'une direction (vecteur libre) se fait aussi par quatre composantes,

La quatrième étant nulle, indique un point à l'infini. Si l'on note u_x, u_y, u_z les coordonnées cartésiennes d'un vecteur unitaire u, en coordonnées homogènes on écrit :

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

I.2.7.3 Matrice de rotation

Soient deux repères R_0 et R_1 , lorsque leurs origines coïncident

($O_0 = O_1$), et que le repère R_1 effectue une rotation d'un angle θ autour d'un ou plusieurs axes de R_0 alors cette rotation peut être décomposée en rotations élémentaires de différents angles autour des axes de R_0 , la matrice de transformation est obtenue par le produit de ces trois matrices de rotation :

Le déterminant des matrices de rotation propre orthogonal par rapport à x, y, z égale à 1 (det=1) donc $R(x_0, \theta_1)^{-1} = R(x_0, \theta_1)^T$.

- Rotation d'un angle θ_1 autour de x_0 :

$$Rot(x_0, \theta_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_1 & -\sin \theta_1 & 0 \\ 0 & \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.3)$$

- Rotation d'un angle θ_2 autour de y_0 :

$$Rot(y_0, \theta_2) = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.4)$$

- Rotation d'un angle θ_3 autour de z_0 :

$$Rot(z_0, \theta_3) = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.5)$$

I.2.7.4 Matrice de translation

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

$$A = trans(p_x, p_y, p_z) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.6)$$

I.2.7.5 Transformation homogènes (matrice 4×4)

Elle inclut à la fois la rotation et la position, soit A et B deux repères avec des axes qui sont pas les mêmes [2].

$$T = T_A^B = T_B^A = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ N_{1 \times 3} & 1 \end{bmatrix}, \quad N_{1 \times 3} = [0 \quad 0 \quad 0] \tag{I.7}$$

Avec :

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

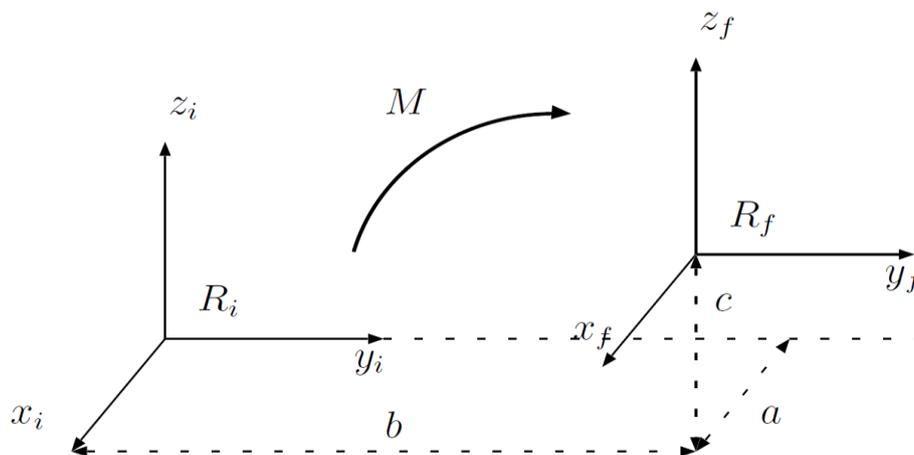


Figure I.6 Translation pure d’un repère R_f par rapport à un repère R_i

I.3 Modélisation des robots

I.3.1 Modélisation géométrique(MG)

La modélisation géométrique inclue deux modèles différents et permet de calculer la position et l’orientation de l’organe terminal et les variables articulaires connaissant la situation particulière du poignet d’un robot. Ces modèles sont :

I.3.1.1 Modèle géométrique direct (MGD)

Ce modèle permet de déterminer la position et l’orientation de l’organe terminal en fonction des variables articulaires noté par : $X(t)=f(\theta(t))$

Avec :

$\theta(t)=[\theta_1, \theta_2, \dots, \theta_n]^T \in \mathbb{R}^n$: Vecteur des variables articulaires.

$X(t)=[x_1, x_2, \dots, x_n]^T \in \mathbb{R}^m$: Vecteur des variables opérationnelles, $m \leq n$.

- **Calcul du MGD**

Le calcul de MGD peut se faire selon plusieurs méthodes la plus répandue est celle de Denavit-Hartenberg (DH). Il existe aussi la notation de Khalil et Kleinfinger qui contient un nombre minimum de paramètres par rapport à celle de DH aussi des architectures ouvertes simples et complexes.

- **Méthode de Denavit-Hartenberg (DH)**

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

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

- Les corps sont supposés parfaitement rigides. Ils sont connectés par des articulations considérées comme idéales (pas de jeu mécanique, pas d'élasticité), soit rotoïdes, soit prismatiques.
- le repère R_i est lié à la fin de corps C_i .
- l'axe Z_{i-1} est porté par l'axe de l'articulation i .
- l'axe X_i est porté par la perpendiculaire commune aux axes Z_i et Z_{i+1} et, Si les axes Z_i et Z_{i+1} et sont parallèles ou colinéaires, le choix de X_i n'est pas unique : des considérations de symétrie ou de simplicité permettent alors un choix rationnel.

Les paramètres de Denavit-Hartenberg(DH):

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

- α_i : angle entre les axes Z_{i-1} et Z_i correspondant à une rotation autour de X_i .
- d_i : Distance entre X_{i-1} et X_i le long de Z_i .
- θ_i : Angle entre les axes X_{i-1} et X_i correspondant à une rotation autour de Z_i .
- a_i : Distance entre Z_{i-1} et Z_i le long de X_i .

La matrice de passage d'une articulation :

$$T = T_{i-1}^i = R_{z,\theta_i} \times T_{z,d_i} \times T_{x,a_i} \times R_{x,\alpha_i} \quad (I.8)$$

Le MGD d'un bras obtenu à partir:

$$T_{MGD} = T_0^n = T_0^1 \times T_1^2 \times \dots \times T_{n-1}^n \quad (I.9)$$

Avec n: nombres d'articulations.

➤ **La méthode de Denavit-Hartenberg modifié(MDH)**

Nous avons:

- le repère R_i est lié au début de corps C_i associée à l'articulation i .
- pour passer de R_{i-1} vers R_i il faut suivre chemin suivant:

1. Rotation (X_{i-1}, α_{i-1}).
2. Translation (X_{i-1}, a_{i-1}).
3. Rotation (Z_i, θ_i).
4. Translation (Z_i, d_i).

Les paramètres de Denavit-Hartenberg modifié(MDH):

- a_i : Distance entre Z_i et Z_{i+1} le long de X_i .
- α_i : angle entre les axes Z_i et Z_{i+1} correspondant à une rotation autour de X_i .
- θ_i : Angle entre les axes X_{i-1} et X_i correspondant à une rotation autour de Z_i .
- d_i : Distance entre X_{i-1} et X_i le long de Z_i .

La matrice de passage d'une articulation :

$$T_{MDH} = T_{i-1}^i = R_{x,\alpha_{i-1}} \times T_{x,a_{i-1}} \times R_{z,\theta_i} \times T_{z,d_i} \quad (I.10)$$

I.3.1.2 Modèle géométrique inverse (MGI)

Le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Le modèle s'écrit :

$$\theta = f^{-1}(x) \quad (\text{I.11})$$

La détermination du modèle géométrique inverse (MGI) est un problème complexe. On doit inverser un système d'équations non linéaires. Toutefois et selon la structure du manipulateur étudié, il existe des méthodes permettant de résoudre le MGI sous une forme explicite. Nous présentons trois méthodes de calcul du MGI :

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

Pour la plupart des manipulateurs utilisés dans l'industrie, la méthode de Paul. Permet de donner les solutions du MGI sous forme explicite comme été démontré dans [6].

Ce schéma permet de démontrer la relation entre le modèle géométrique direct et inverse.

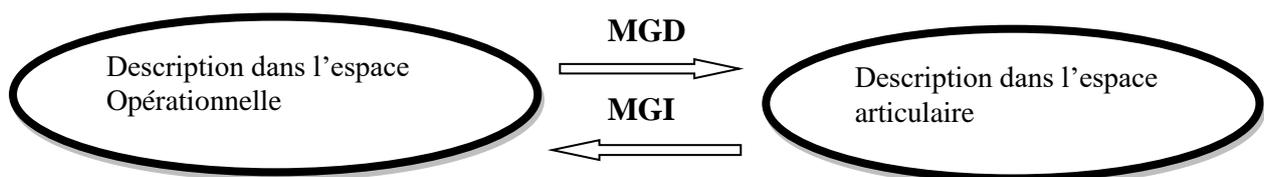


Figure I.7 Modèle géométrique directe et inverse.

I.3.2 Modèle cinématique(MC)

Le modèle cinématique permet de trouver une solution au problème géométrique inverse quelle que soit la structure du robot. Il permet aussi d'introduire la notion de vitesse de déplacement du mécanisme, pour calculer les vitesses de translation et de rotation du corps terminal en fonction des vitesses articulaires.

I.3.2.1 Modèle cinématique direct(MCD)

Le modèle cinématique complète le modèle géométrique, il permet de déterminer la vitesse de l'organe terminal dans l'espace opérationnel en fonction de la vitesse des variables articulaires. Le modèle cinématique est linéaire par rapport aux vitesses.

Le modèle est décrit par l'équation :

$$\dot{X} = J(q)\dot{q} \quad (\text{I.12})$$

Où : $J(q) \in \mathbb{R}^{n \times m}$ est la matrice jacobéenne de dimension $(m \times n)$ du mécanisme, égale à $:\frac{dX}{dq}$ et fonction de la configuration articulaire q . La même matrice jacobéenne intervient dans le calcul du modèle différentiel direct qui donne les variations élémentaires dX des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires dq , soit :

$$dX = J(q) dq \quad (\text{I.13})$$

- **La matrice jacobéenne**

C'est un opérateur de liaison des vitesses des corps leur intérêt est de calculer une solution locale des variations articulaire dq connaissant celle opérationnelles dx , aussi pour établir les relations des efforts exercés par l'organe terminal sur l'environnement, et faciliter le calcul des singularités.

I.3.2.2 Modèle cinématique inverse(MCI)

L'objectif du modèle cinématique inverse est de calculer à partir d'une configuration q données les vitesses articulaires \dot{q} qui assurent au repère terminal une vitesse opérationnelle X imposée.

L'inversion du modèle cinématique direct pour obtenir le modèle cinématique inverse, passent par la résolution d'un système d'équations linéaires. Il existe plusieurs méthodes pour la détermination de ce dernier. Ces méthodes sont soit analytiques, soit numériques.

- **Les méthodes analytiques** : Conviennent pour résoudre le problème inverse lorsque la matrice jacobéenne associée au mécanisme est régulière.
- **Les méthodes numériques** : Généralement sont plus répandue. Elles nécessitent un temps de calcul relativement important.

I.3.3 Modèle dynamique (MD)

Le modèle dynamique est la relation entre les couples et/ou forces appliqués aux actionneurs (articulations) et les variables articulaires (positions, vitesses et accélérations). On représente le modèle dynamique par une relation de la forme [6].

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

Avec:

Γ : Vecteurs des couples/forces des actionneurs, selon que l'articulation est rotoïdes ou prismatique. (dans la suite on écrira tout simplement couple).

q : Vecteur des positions articulaires.

\dot{q} : Vecteur des vitesses articulaires.

\ddot{q} : Vecteur des accélérations articulaires

f_e : Vecteur représentant les efforts extérieurs (forces et moment) qu'exerce le manipulateur sur l'environnement.

I.3.3.1 Modèle dynamique inverse(MDI)

Le modèle dynamique inverse exprime les couples exercés par les actionneurs en fonction des positions, vitesses et accélérations des articulations. Le modèle dynamique inverse est représenté par une relation de la forme:

$$\Gamma = f(\dot{\theta}, \ddot{\theta}) \quad (\text{I.15})$$

Γ : Vecteur des couple/forces des actionneurs, selon que l'articulation est rotoïdes ou prismatiques.

$\dot{\theta}$: Vecteur des vitesses articulaires.

$\ddot{\theta}$: Vecteur des accélérations articulaires.

Plusieurs formalismes ont été utilisés pour obtenir le modèle dynamique des robots. Les formalismes les plus souvent utilisés sont [7]:

- Le formalisme de Lagrange-Euler et celui de Newton-Euler.

Dans ce qui suit le paramètre θ est changé par le paramètre q selon la notation de référence utilisée.

- **Formalisme de Newton-Euler :**

Nous allons voir les équations de vitesse, d'accélération, de force et de couple nous permettant d'obtenir la dynamique inverse. Tous les calculs qui suivent sont effectués selon la méthode de Newton-Euler, de façon itérative, i.e. membre par membre. La première partie des itérations se fait de la base vers l'effecteur ; pour calculer les positions et les vitesses articulaires d'un certain membre, nous avons besoin des résultats du membre précédent. La seconde partie de l'algorithme, qui consiste à calculer les forces et couples articulaires, s'effectue de l'effecteur à la base. Ces formules sont démontrées dans [8].

D'abord, nous devons calculer les vitesses angulaires et linéaires de chacun des membres. La notation ${}^{i+1}\omega_{i+1}$ signifie la vitesse angulaire du membre $i+1$ par rapport à la base, représentée dans le référentiel $i+1$. Le vecteur ${}^{i+1}\hat{Z}_{i+1} = [0 \ 0 \ 1]$ sert à transformer un scalaire en vecteur orienté vers l'axe des Z . Pour une articulation rotative, les formules sont les suivantes :

$${}^{i+1}\omega_{i+1} = {}^{i+1}R^i \omega_i + \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (I.16)$$

$${}^{i+1}V_{i+1} = {}^{i+1}R^i ({}^iV_i + {}^i\omega_i \times {}^iP_{i+1}) \quad (I.17)$$

Pour une articulation prismatique, les équations de vitesses rotative et linéaire sont :

$${}^{i+1}\omega_{i+1} = {}^{i+1}R^i \omega_i \quad (I.18)$$

$${}^{i+1}V_{i+1} = {}^{i+1}R^i ({}^iV_i + {}^i\omega_i \times {}^iP_{i+1}) + \dot{d}_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (I.19)$$

Ensuite nous devons calculer les accélérations rotative et linéaire de l'origine de chacun des référentiels des membres. Pour une articulation relative, les formules sont les suivantes :

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}R^i \dot{\omega}_i + {}^{i+1}R^i \dot{\omega}_i \times \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (I.20)$$

$${}^{i+1}\dot{V}_{i+1} = {}^{i+1}R^i (\dot{\omega}_i \times {}^iP_{i+1} + \dot{\omega}_i \times ({}^i\omega_i \times {}^iP_{i+1}) + \dot{V}_i) \quad (I.21)$$

Nous devons aussi trouver l'accélération linéaire du centre de masse de chaque membre. Nous posons qu'un référentiel « Ci » est attaché à chaque lien avec son origine positionnée au centre de masse du membre et avec la même orientation que le référentiel « i ». L'équation suivante est valide pour les articulations rotatives et linéaires :

$${}^i\dot{V}_{ci} = \dot{\omega}_i \times {}^iP_{ci} + \dot{\omega}_i \times ({}^i\omega_i \times {}^iP_{ci}) + \dot{V}_i \quad (I.22)$$

Où ${}^iP_{ci}$ et ${}^i\dot{V}_{ci}$ sont la position et l'accélération de l'origine du référentiel de la masse du membre i par rapport au référentiel du membre i. Nous sommes maintenant en mesure d'appliquer les équations de Newton-Euler pour calculer les forces F_i et couples N_i d'inertie agissant sur chaque membre:

$$F_i = m_i \dot{V}_{ci} \quad (I.23)$$

$$N_i = {}^{ci}I_i^i \dot{\omega}_i + \dot{\omega}_i \times {}^{ci}I_i^i \omega_i \quad (I.24)$$

Où ${}^{ci}I_i$ est le tenseur d'inertie au centre de masse représenté dans le référentiel i et définissant la forme et la distribution de la masse du membre. Une fois toutes les vitesses, accélérations, forces et couples calculés pour chaque membre (de la base à l'effecteur), nous devons effectuer un calcul itératif dans le sens inverse (partant de l'effecteur vers la base). Nous calculerons les forces et couples aux joints qui résultent des forces et couples d'inertie agissant sur chaque membre et des forces et couples provenant des membres suivants (forces et couples externes):

$${}^i f_i = {}^{i+1}R^i {}^{i+1}f_{i+1} + {}^i F_i \quad (I.25)$$

$${}^i n_i = {}^i N_i + {}^{i+1}R^i {}^{i+1}n_{i+1} + {}^i P_{ci} \times {}^i F_i + {}^i P_{i+1} \times {}^{i+1}R^i {}^{i+1}f_{i+1} \quad (I.26)$$

Notons que la notation ${}^i f_i$ désigne la force exercée par le membre i-1 sur le membre i représenté dans le référentiel i. Finalement, pour une articulation rotative, le couple articulaire est:

$$T_i = {}^i n_i {}^{Ti}\hat{Z}_i \quad (I.27)$$

Pour une articulation prismatique, la force articulaire est:

$$\mathbf{T}_i = {}^i\mathbf{f}_i \hat{\mathbf{z}}_i \quad (\text{I.28})$$

Nous pouvons inclure la gravité dans la dynamique inverse en définissant la condition suivante:

$${}^0\dot{\mathbf{V}}_0 = -g \dot{\mathbf{V}} \quad (\text{I.29})$$

Où $\dot{\mathbf{V}}$ est un vecteur unitaire possédant la même orientation que la gravité g est la constante gravitationnelle (scalaire). Ceci fait en sorte que le manipulateur entier accélère dans le sens opposé de la gravité, produisant ainsi le même effet que si la gravité était présente.

➤ **Formalisme de Lagrange-Euler :**

Par la formulation de Lagrange, nous pouvons trouver les équations dynamiques d'une façon systématique. Nous devons choisir un ensemble de coordonnées généralisées décrivant la position du système.

La plupart du temps, nous utilisons directement les variables articulaires, c'est à dire les angles et longueurs décrivant la position de chacune des articulations mobiles. Le Lagrangien du système mécanique est comme suit:

Le formalisme de Lagrange décrit les équations du mouvement en termes de travail et d'énergie du robot. La forme générale représentant le mouvement d'un robot manipulateur constitué de n degrés de liberté et de n articulations, s'écrit suivant Lagrange [6] :

$$\Gamma_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{\lambda}_i} - \frac{\partial L}{\partial \lambda_i} \quad (\text{I.30})$$

Avec : $i = 1, \dots, n$

Où Γ_i sont les forces généralisées associées aux coordonnées généralisées λ_i .

Pour un manipulateur avec chaîne cinématique ouverte, un choix naturel de coordonnées généralisées est le vecteur de variables articulaires: $\lambda_i = q_i$. Dans ce cas, les forces généralisées seront les couples/forces articulaires.

La dynamique inverse peut s'écrire sous la forme Lagrange-Euler de la façon suivante:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + J_r(q) F_e \quad (I.31)$$

où τ est le vecteur de forces/couple articulaires, $M(q)$ est la matrice des masses et inerties (définie positive et symétrique), $C(q, \dot{q})$ est la matrice des termes centrifuges et de Coriolis (peut être la matrice de Cristoffel, qui correspond à un choix particulier), $G(q)$ est le vecteur des termes de gravité et $J_r(q)$ est le Jacobiéen de force et F_e , le vecteur de forces externes. Le Jacobiéen de force est le Jacobiéen de vitesse transposé.

I.3.3.2 Modèle dynamique directe (MDD)

La formulation de Lagrange-Euler peut être utilisée pour calculer la dynamique directe en réécrivant son équation de la façon suivante:

$$\ddot{q} = M^{-1}(q) [\tau - C(q, \dot{q})\dot{q} - G(q) - J_r(q)F_e] \quad (I.32)$$

Pour calculer la dynamique directe avec l'algorithme de Newton-Euler, nous pouvons

Procéder de la manière suivante. Nous avons déjà le vecteur de couple τ (entrée de l'algorithme). Nous pouvons obtenir les termes $C(q, \dot{q})\dot{q} + G(q) + J_r(q) F_e$ en calculant la dynamique inverse avec \ddot{q} nulle, q et \dot{q} étant connues (valeurs de l'itération précédente). Une des façons possibles de trouver $M(q)$ consiste à évaluer cette matrice une colonne à la fois. Pour ce, nous devons calculer la dynamique inverse n fois, où n est le nombre d'articulations. L'entrée pour chacun des calculs de dynamique inverse est la suivante:

q du cycle précédent.

$\dot{q} = 0$ pour toutes les articulations.

$\ddot{q}_i = 1$ où i est le numéro de la colonne évaluée.

$\ddot{q}_k = 0$ pour $k = 1$ à N sauf i .

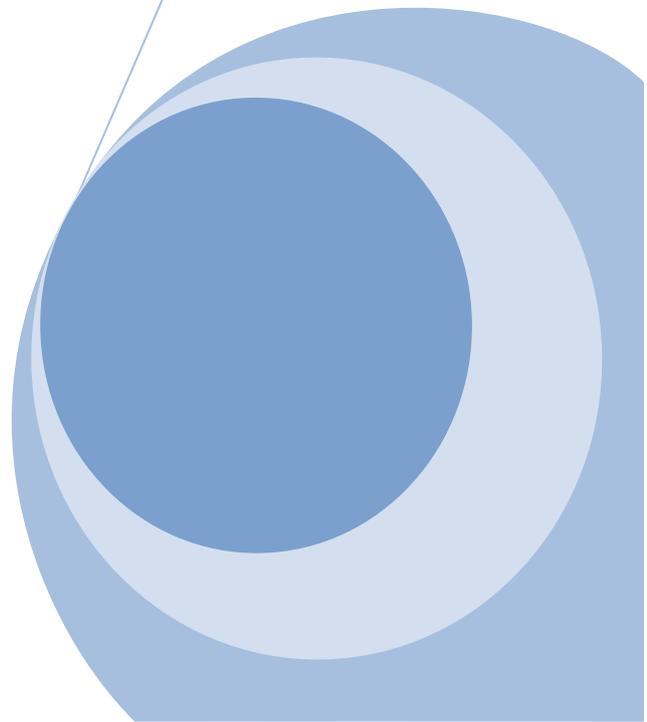
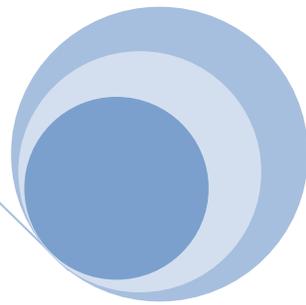
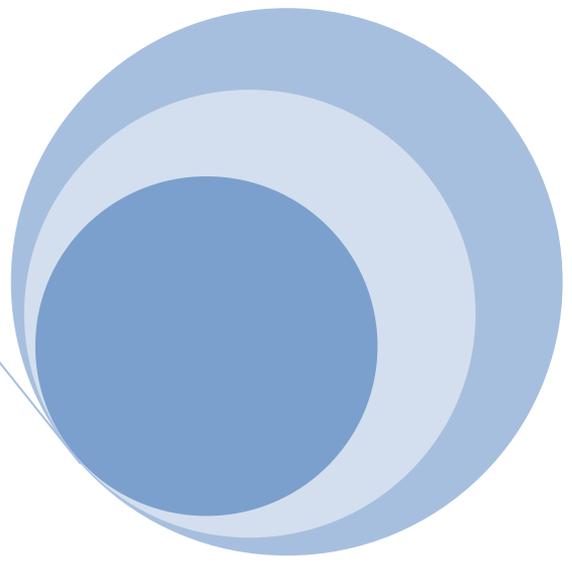
$g = 0$.

Avec (2.32), nous sommes en mesure de calculer l'accélération articulaire \ddot{q} . Pour obtenir \dot{q} et q , il suffit d'intégrer deux fois \ddot{q} par rapport au temps [7].

I.4 Conclusion

Durant ce chapitre nous avons abordé des généralités sur la robotique telle que les constituants de robot manipulateur, les types, les différentes morphologies qui existent et les caractéristiques d'un robot puis à la fin de ce chapitre, nous avons traité la modélisation des robots avec certains détails.

**Commande optimale et planification
de trajectoire**



II.1 Introduction

La commande des systèmes est un domaine très vaste. Les solutions apportées ont pour objectif une meilleure utilisation des systèmes commandés. La commande optimale constitue une partie importante. Son principe est de commander le système de façon à minimiser un certain critère de performance. Elle trouve application dans tous les systèmes traités dans le domaine de la commande. La robotique n'en fait pas l'exception. Dans ce travail nous nous intéressons à l'application de la commande optimale au domaine de planification de trajectoire pour robot manipulateurs. Dans ce chapitre, nous donnons un aperçu sur la commande optimale et sur la planification de trajectoire des robots manipulateurs.

II.2 La commande optimale

II.2.1 Définitions

De point de vue formel, un problème de commande optimale est un problème de minimisation ou de maximisation d'une fonctionnelle. Elle est basée sur un modèle mathématique du système, une formulation de l'indice de performance qui appelé critère en automatique et de contraintes.

II.2.2 Objectifs de la commande optimale

L'objectif de la commande optimale est d'introduire une commande $u^*(t)$ ou $u^*(x, t)$ en vue d'améliorer le comportement du système. Pour concevoir la commande optimale, on doit tout d'abord choisir un modèle mathématique pour le système à commander. Ce modèle décrit, par un ensemble d'équations d'état, le comportement du système qui est donnés par :

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

Ensuite, vient la formulation de l'indice de performance J qu'on doit minimiser ou maximiser en respectant les contraintes imposées.

II.2.3 Des notions sur la commande optimale

Le problème de la commande optimale sans contraintes peut être énoncé comme suit : étant donné f , x_0 , t_0 , t_f , φ , et L , il s'agit de trouver $u^*(t)$ ou $u^*(x, t)$ qui minimise l'indice de performance ou critère (en automatique selon le domaine).

Chapitre II Commande optimale et planification de trajectoire

Les critères les plus utilisés sont :

- Problème à temps minimal (TM)

$$J = \int_{t_0}^T dt \quad (\text{II.2})$$

- Cas linéaire quadratique(LQ)

$$J = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t) dt \quad (\text{II.3})$$

Où $L(x(t), u(t), t)$ est une fonction non linéaire aussi décrit le coût à chaque instant sur la trajectoire $x(t)$ et $\varphi(x(t_f), t_f)$ représente la fonction de coût terminal.

Avec

$$\dot{x}(t) = f(x(t), u(t), t) \quad (\text{II.4})$$

Ou

$$x(t_0) = x_0$$

Ce problème peut être résolu par plusieurs méthode.

II.2.3.1 Les contraintes de la commande optimale

Si on parle des contraintes donc on parle de problème de cette commande, pour obtenir un résultat qui satisfait notre difficulté d'optimalité il existe des conditions nécessaires peuvent être déduites par des méthodes de calcul des variations point par point qui donnent une solution au problème posé.

Pour la formulation des contraintes, il faut noter leur diversité lors de la commande d'un processus : soit sur le temps de simulation: libre ou imposé, sur la valeur de la commande par exemple $-1 < u(t) < 1$, sur l'état du système,... [9].

II.2.4 Les méthodes de la résolution de problème de commande optimale

La résolution de problèmes de commande optimale peut se faire en plusieurs étapes :

On commence par la position du problème (II-4), ensuite, on choisit de critère à minimiser (II-3) puis on détermine la commande $u^*(t)$ ou $u^*(x, t)$ sur l'intervalle $[t_0, T]$ qui transfère le système décrit par (II-4) le long d'une trajectoire optimale $x^*(t)$ qui minimise le critère(II-3).

II.2.4.1 Les méthodes indirectes

Ces méthodes basées sur la transformation du problème de la commande optimale à un problème de programmation non linéaire.

II.2.4.1.1 Multiplicateur de Lagrange (équation d'Euler-Lagrange)

Pour résoudre le problème (II.3) et (II.4), nous utiliserons les multiplicateurs de Lagrange, $\lambda(t) \in R_n$. Par conséquent l'indice de performance est donné par :

$$J_A = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} [L(x(t), u(t), t) + \lambda^T(t) (f(x(t), u(t), t) - \dot{x})] dt \quad (\text{II.5})$$

Introduisons le Hamiltonien H défini par :

$$H(x(t), u(t), \lambda(t), t) = L(x(t), u(t), t) + \lambda^T(t) f(x(t), u(t), t) \quad (\text{II.6})$$

Nous pouvons, ainsi, écrire (II.5) sous la forme :

$$J_A = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} [H(x(t), u(t), \lambda^T(t), t) - \int_{t_0}^{t_f} \lambda^T \dot{x}(t)] dt \quad (\text{II.7})$$

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

$$\int_{t_0}^{t_f} \lambda^T \dot{x}(t) = \lambda^T(t_f) x(t_f) - \lambda^T(t_0) x(t_0) - \int_{t_0}^{t_f} \dot{\lambda}^T x(t) \quad (\text{II.8})$$

Cependant (II.7) devient :

$$J_A = \varphi(x(t_f), t_f) - \lambda^T(t_f) x(t_f) + \lambda^T(t_0) x(t_0) + \int_{t_0}^{t_f} [H(x(t), u(t), \lambda^T(t), t) + \dot{\lambda}^T x] dt \quad (\text{II.9})$$

Le problème original, (II.3) et (II.4), a été converti à un problème de minimisation, (II.9), sans contraintes.

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

$$\dot{\lambda}^T(t) = -\frac{\partial H}{\partial x} = -\lambda^T \frac{\partial f}{\partial x} - \frac{\partial L}{\partial x} \quad (\text{II.10})$$

Avec la condition finale :

$$\lambda^T(t) = -\frac{\partial \varphi}{\partial x} \Big|_{t=t_f} \quad (\text{II.11})$$

II.2.4.1.2 Le principe de Pontryagin

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

Chapitre II Commande optimale et planification de trajectoire

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

$$\dot{\lambda}^T = -\frac{\partial H}{\partial x} \quad (\text{II.12})$$

$$\lambda^T(t_f) = \left(\frac{\partial \varphi}{\partial x}\right) \quad (\text{II.13})$$

$$H(x^*, x^*, u^*, \lambda^*, t) \leq H(x^*, x^*, u, \lambda^*, t) \quad (\text{II.14})$$

Pour tout $t \in [t_0, t_f]$ et pour tout $u(t) \in U \subset \mathbb{R}_m$. Donc la commande optimale doit minimiser le Hamiltonien défini par (II.7).

L'inégalité (II.9) 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 de variations.

- **Cas système non linéaire**

Il faut aussi signaler que l'approche variation-el et le principe de minimum mènent à un problème connu comme « A non linear two-point boundary value problem : NLTPBVP » qui est difficile à résoudre de façon analytique.

II.2.4.1.3 Méthode en boucle fermé

Parmi les méthodes proposées pour obtenir la commande optimale en boucle fermée on citera 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 NLTPBVP.

La seconde approche, pour obtenir cette commande, 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.

II.2.4.1.4 Méthode en boucle ouverte

Parmi elles. 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 standards pour résoudre les problèmes de la commande optimale.

II.2.4.2 Les méthodes directes

Il existe plusieurs techniques pour résoudre le problème de la commande optimale. Pour réaliser une tâche il peut y avoir plusieurs solutions puis reste le choix de la bonne solution qui minimise ou maximise un critère, c'est le cas de CO. Ces méthodes offrent quelques avantages comme:

L'existence d'algorithmes pour résoudre les problèmes de programmation non linéaire.

La possibilité de traiter facilement différents types de contraintes.

Le problème de commande optimale peut être converti en un problème de programmation mathématique soit en utilisant la technique de discrétisation, soit en se servant de la technique de paramétrisation.

II.2.4.2.1 La méthode de discrétisation

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

$$t_0 < t_1 < t_2 \dots < t_n < t_f$$

Où les points t_i sont appelés nœuds.

Objectif de cette méthode est de discrétiser les variables d'état et les variables de contrôle ; nous avons donc la séquence des valeurs inconnues des variables d'état et de contrôle suivante :

$$Z = (x_0, x_1, \dots, x_n, u_0, u_1, \dots, u_{n-1}) \quad (\text{II.15})$$

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 être résolu par les techniques de la programmation non linéaire.

L'inconvénient majeur de cette approche est la dimension élevée du vecteur z .

Une autre approche possible serait de discrétiser la variable de contrôle seule

$$z = (u_0, u_1, \dots, u_{n-1}) \quad (\text{II.16})$$

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

II.2.4.2.2 La méthode de para-métrisation

Cette technique de para-métrisation offre la possibilité d'être appliquée de trois manières différentes :

1. 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 :

$$u_k(t) = \sum_{i=0}^N c_i^{uk} \varphi_i(t) \quad (\text{II.17})$$

Avec $K=1,2,\dots,m$.

c_i^{uk} : Les paramètres inconnus.

φ_i : Dénotent un ensemble approprié de fonctions formant une base de l'espace de commande à dimension finie.

Les variables d'état sont obtenues en fonction des paramètres inconnus en intégrant les équations d'état du système et en substituant les variables approximatives de la commande et les variables d'état correspondantes dans l'indice de performance aussi le problème de la commande optimale est ainsi converti en un problème de programmation non linéaire à paramètres statiques, plus facile à résoudre que le problème initial.

Pour l'approximation des variables de la commande plusieurs fonctions ont été utilisées 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 d'approximation nécessite l'intégration des équations d'état, ce qui représente un processus coûteux en termes de temps de calcul.

2. Para-métrisation d'état-commande

Cette approche est basée sur l'approximation des variables d'état et de commande à la fois par une séquence de fonctions connues avec des paramètres inconnus :

$$u_k(t) = \sum_{i=0}^N c_i^{uk} \varphi_i(t) \quad K=1,2,\dots,m \quad (\text{II.18})$$

$$x_j(t) = \sum_{i=0}^N c_i^{uj} \varphi_i(t) \quad j=1,2,\dots,m \quad (\text{II.19})$$

Où

c_i^{uk} et c_i^{xj} : des paramètres inconnus et φ_i : un ensemble de fonction appropriées.

En utilisant cette méthode, le problème de la commande optimale se convertit en un problème de programmation mathématique non linéaire.

L'inconvénient principal de cette approche est le nombre important de paramètres que l'on doit déterminer. Cependant, suivant cette approche, nous aboutirons à un problème de programmation non linéaire à dimensions très élevées en termes de nombre de paramètres inconnus et du nombre de contraintes d'égalité.

3. Para-métrisation d'état

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

$$x_j(t) = \sum_{i=0}^N c_i^{uk} \varphi_i(t) \quad (\text{II.20})$$

Avec $j=1,2,\dots,m$

Dans cette technique, les variables de commande sont obtenues à partir des équations d'état.

Cette technique possède des avantages :

On peut se passer du besoin d'intégrer les équations d'état.

- Les paramètres inconnus sont moins nombreux en regard d'autres technique.
- Les paramètres peuvent être manipulés directement.

Néanmoins cette technique présente quelques inconvénients :

- Il est difficile de traiter les systèmes non linéaires du fait qu'on ne trouve pas systématiquement des variables de commande comme fonction de variables d'état.
- Cette technique n'a pas de voie d'application systématiquement aux problèmes de la commande optimale à aspect général quand le nombre de variables d'état et de commande est différent [10].

II.3 Planification de trajectoire

II.3.1 Définitions

Chapitre II Commande optimale et planification de trajectoire

La planification elle sert à donner les moyens au robot de trouver une suite d'actions à appliquer sur le monde, pour le faire passer de l'état initial à un état qui satisfait le but à atteindre tout en obéissant à un modèle dynamique et en évitant les obstacles. Ainsi, pour qu'un robot suit une trajectoire bien définie il faut respecter les contraintes du robot et celle de l'environnement.

1. Contraintes relatives au robot

Concernant sa géométrie, sa cinématique et sa dynamique et même l'architecture (type) du robot : Pouvant correspondre à un système articulé d'objets rigides (bras manipulateur), un robot mobile, une main à plusieurs doigts, ou encore à plusieurs systèmes de robots à coordonner tels que des bras manipulateurs ou des robots mobiles.

2. Contraintes de l'environnement

Concernant essentiellement la non-collision aux obstacles, la prise en compte d'interactions de contact avec le robot.

II.3.2 Les différents types de planification de trajectoire

Il existe deux types de planifications de trajectoire tel que:

II.3.2.1 Planification de mouvements fins

C'est une série de petits mouvements qui représentent des essais, des erreurs et des corrections de faibles amplitudes.

II.3.2.2 Planification de chemin (trajectoire)

Dans l'univers géométrique, il est indispensable de pouvoir déterminer à l'avance le parcours du robot ou la trajectoire d'un bras manipulateur afin d'éviter les collisions comme illustré dans la (figure II.1) [11].

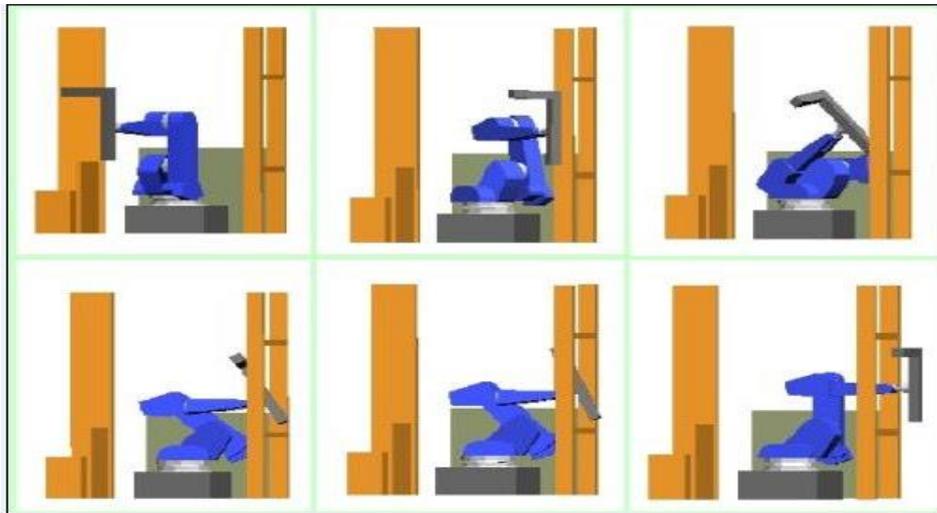


Figure II.1 Planification du chemin.

II.3.3 Méthodes de planification de trajectoire (chemin)

Depuis une vingtaine d'années, plusieurs techniques de planification ont été proposées, mais très peu de ces techniques sont efficaces à résoudre de façon satisfaisante le problème de la planification, notamment dans l'industrie.

En effet, selon l'environnement, la planification de trajectoire peut être classée en deux méthodes:

II.3.3.1 Les méthodes globales

Sont basées sur la connaissance de l'ensemble de l'espace libre. La construction de cet espace libre est très coûteuse en temps de calcul, ce qui limite en pratique leur utilisation.

La plupart, de ces méthodes sont basées sur le concept d'espace des configurations (CSA) du robot introduit par 'Lozano- Perez'. Le principe de cette méthode est basé sur le mouvement solution qui correspond à une séquence continue de configurations sans collision entre deux configurations, initiale et finale.

II.3.3.1.1 Méthodes par décomposition cellulaire

Chapitre II Commande optimale et planification de trajectoire

Elle consiste à partitionner l'espace de configurations libres (CSA) du robot en un ensemble de régions (cellules) adjacentes. La description de la décomposition obtenue est représentée dans un graphe de connectivité, dont les nœuds correspondent aux différentes régions et les arcs aux relations d'adjacences entre elles.

Le problème de planification, entre deux configurations situées dans deux cellules différentes est résolu en quatre étapes:

Etape 1 : Partitionner l'espace de configurations libres (CSA) à un ensemble de cellules adjacentes.

Etape 2 : Créer le graphe de connectivité reliant les différentes cellules adjacentes de (CSA). Généralement, on utilise le point milieu de la frontière de deux cellules adjacentes.

Etape 3 : Identifier les cellules dans les quelles se trouvent la configuration initiale et la configuration finale.

Etape 4 : Recherche de la solution (chemin sans collision) à partir de l'enveloppe définit par la liste de cellules adjacentes trouvées dans (Etape 2) reliant les deux configurations initiale et finale.

L'approche par décomposition a été adoptée dans plusieurs travaux et a conduit à la mise en œuvre de nombreuses méthodes. Celles-ci sont généralement classées en deux catégories :

1. Méthodes exactes

Une décomposition exacte donne un ensemble de polygones subdivisant l'espace des configurations libres (figures II.2 (a) et (b)). A partir de cette représentation, on crée un graphe reliant les différentes composantes adjacentes de l'environnement (figure II.2 (c)). Ensuite on identifie les cellules dans les quelles se trouvent la configuration initiale et la configuration finale.

Le problème de planification consiste alors en un problème de recherche dans un graphe. Pour cela, on choisit un algorithme de parcours (A*, depth-first, breadth-first) permettant d'optimiser la distance parcourue. Finalement, il faut relier les cellules de graphe solution par un chemin sans collision. Généralement, on utilise le point milieu de la frontière de deux cellules adjacentes (figure II.2 (d)).

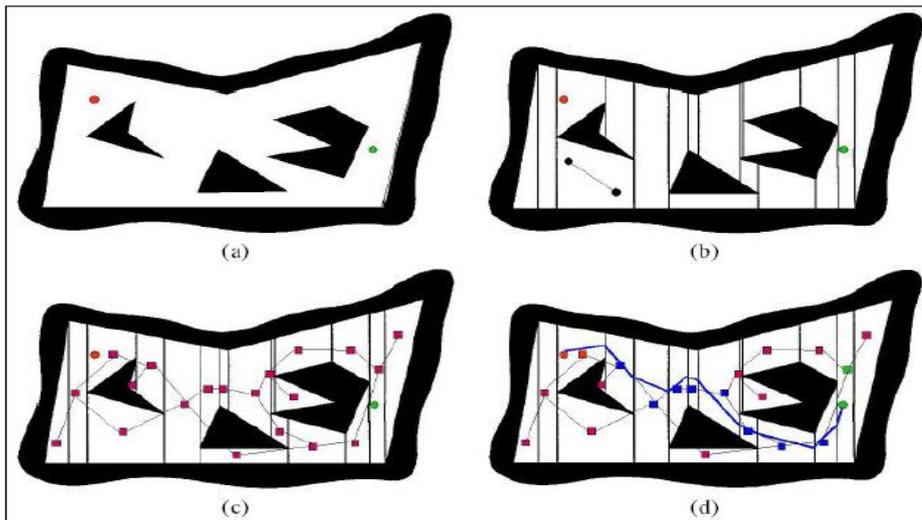


Figure II.2 Méthodes par décomposition cellulaire (méthode exacte) et la constitution du graphe.

2. Méthodes approchées

Des méthodes dites approchées sont généralement appliquées. Elles se distinguent des précédentes (méthodes exactes) par la simplicité de la structure des cellules utilisées pour la décomposition et l'approximation faite de l'espace libre du robot.

La structure de celui-ci est généralement capturée dans une représentation hiérarchique en cellules élémentaires identiques et permettant l'adaptation de la taille de celles-ci à la géométrie des zones à recouvrir. Plusieurs types de cellularisation de l'espace du robot sont généralement utilisées en pratique : représentation rectangulaires, découpage en tranches ou encore en polyèdres.

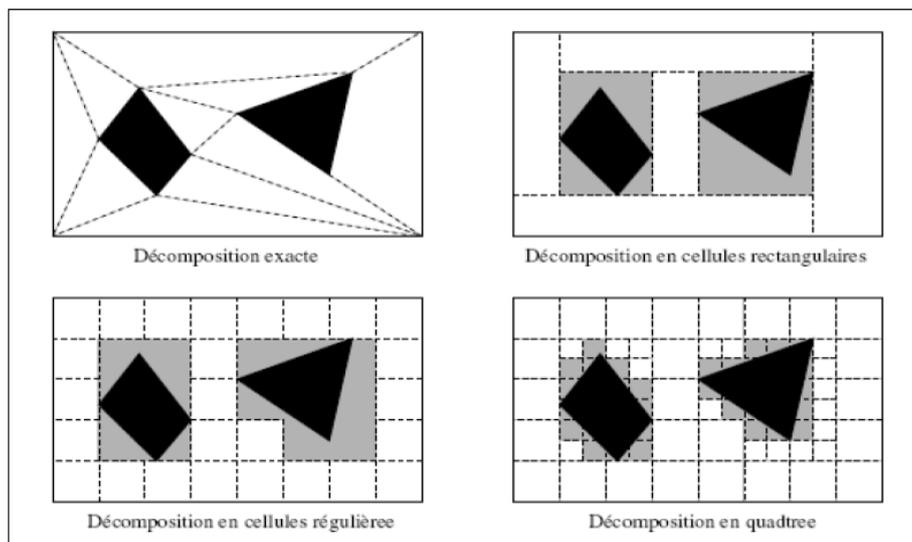


Figure II.3 Méthodes par décomposition cellulaire (méthode approchée).

En comparant les deux approches de décomposition, il ressort que les méthodes exactes sont plus complètes en théorie que les méthodes approchées. Toutefois, la complexité de leur mise en œuvre, les rend moins appropriées en pratique que les méthodes approchées même si celles-ci restent parfois limitées à des espaces de dimension réduite [11].

II.3.3.1.2 Méthodes de résolution de type rétraction (Roadmaps)

La seconde approche pour la résolution du problème de planification, consiste à ramener la recherche du mouvement du robot dans un espace de plus faible dimension que celle de l'espace admissible. Cela consiste à représenter la connectivité de l'espace libre du robot par un réseau de courbes unidimensionnelles pouvant être entièrement dans l'espace libre ou des contacts du robot.

La planification de mouvement entre deux configurations données est alors résolue en trois étapes:

Etape 1: Détermination d'un chemin ramenant le robot de sa configuration initiale à un point situé sur l'une des courbes de réseau de connectivité.

Etape 2: Détermination d'un chemin entre la configuration initiale et le réseau.

Etape 3: Extraire le chemin reliant les deux configurations initiale et finale.

L'approche a été adoptée dans plusieurs travaux et a conduit à la mise en œuvre de nombreuses méthodes (Graphe de visibilité, Diagramme de Voronoï et l'échantillonnage aléatoire).

II.3.3.2 Les méthodes locales

Les méthodes locales se caractérisent par la connaissance locale de l'environnement. A chaque pas de déplacement l'algorithme détermine l'existence ou non de collision, s'il y a une collision, la trajectoire est modifiée localement. Ce type de méthodes permet de résoudre des problèmes dans un temps raisonnable.

Leur principe consiste à déterminer les déplacements du robot en ne considérant qu'une représentation locale de l'environnement et à prévoir la planification comme un problème d'optimisation.

II.3.3.2.1 Méthode du potentiel

Cette méthode a été initialement introduite par "O.Khatib". Elle consiste à assimiler le robot à une particule contrainte à se déplacer dans un champ de potentiel fictif obtenu par la décomposition d'un premier champ attractif au but et d'un ensemble de champ répulsifs modélisant la présence d'obstacles dans l'espace du robot. Les déplacements de celui-ci (robot) sont calculés itérativement par un algorithme de "descente du gradient" du potentiel obtenu.

Le principe de cette méthode est le suivant : Le robot est plongé dans un potentiel qui résulte de la superposition d'un potentiel attractif lié à la configuration à atteindre, et d'une somme de potentiels répulsifs liés aux obstacles.

Pour que le robot soit attiré par le but et repoussé par les obstacles, il suffit de le faire évoluer, selon le gradient décroissant de la fonction potentiel. Cette méthode est utilisée initialement pour la commande et l'évitement d'obstacles, ensuite pour la planification.

L'avantage principal de cette méthode est sa rapidité de calcul permettant de l'appliquer facilement en temps réel, mais elle reste toujours sensible à l'occurrence de minimums locaux engendrant des configurations de blocage ou d'oscillation. Ces minimums sont généralement liés à la géométrie et à la répartition des obstacles dans l'espace de travail du robot et surtout aux coefficients de pénalités créées lors de la construction du champ potentiel.

Dans le cadre de la même approche par champs de potentiel, un autre type de méthode se base sur le calcul de variations pour la minimisation d'une fonctionnelle en considérant l'ensemble des chemins admissibles. Contrairement à la première méthode où le robot est réduit à un point, les méthodes basées sur le calcul de variations considèrent que la variation de base est donnée par une courbe déformable reliant les deux configurations extrêmes. L'évolution de la courbe est donnée par la résolution d'un système de n équations correspondant à la discrétisation spatiale de la courbe en n points.

La fonctionnelle à minimiser est généralement composée de divers termes dont un potentiel de non-collision aux obstacles et des critères de continuité et de différentiabilité de la courbe.

Enfin, la méthode peut fournir des solutions locales aux problèmes de minimisation de la fonctionnelle considérée, mais elle reste sensible à la position de la courbe choisie initialement avant le début de la recherche.

Chapitre II Commande optimale et planification de trajectoire

Ce problème présente une méthode combinant le calcul de variations et une technique de programmation dynamique pour la recherche de chemins maintenant une distance de sécurité par rapport aux obstacles.

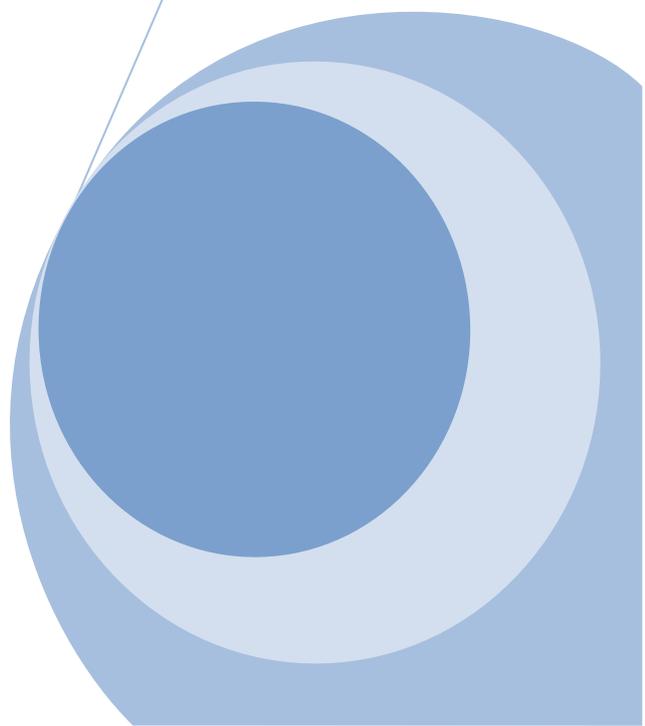
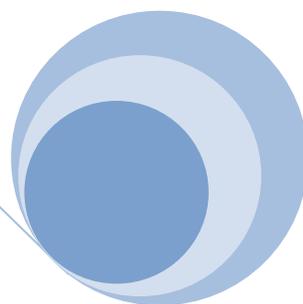
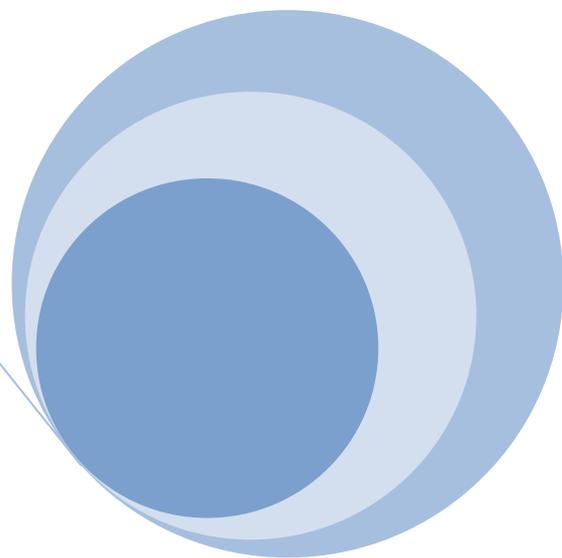
II.3.3.2 La méthode des contraintes

Cette méthode est proposée par "Faverjon et Tournassoud en 1987", elle consiste à aborder le problème de l'évitement d'obstacles, en modélisant localement chacun de ceux-ci (obstacles) par l'ensemble de ses plans tangents. La génération des mouvements du robot est obtenue par la minimisation d'un critère quadratique sur la tâche à effectuer en présence des contraintes linéaires associées aux équations des plans tangents aux obstacles. Toutefois, cette méthode reste sensible à la présence de minimums locaux. La solution de ce problème est la combinaison de cette méthode par une technique d'apprentissage [11].

II.4 Conclusion

Dans ce chapitre nous avons donné l'essentielle de la commande optimale en commençant par la définition, l'objectif et leurs problèmes (contraintes) et terminant par les différentes méthodes de la commande optimale, le deuxième point que nous avons traité est la définition, les classes et la technique de planification de trajectoire illustrée par quelques figures.

**Planification de trajectoire à temps optimal
pour le robot SCARA**



III.1 Introduction

Le problème de planification de trajectoire pour des robots est très complexe. Une approche de le résoudre consiste en son découplage en deux sous problèmes. Un problème de planification de chemin qui prend en considération seulement les contraintes géométriques telles que l'évitement d'obstacle et l'accessibilité. Le deuxième sous problème consiste à paramétrer le chemin obtenu dans la première étape avec le temps en prenant en considération les dynamiques du système [13,15].

Plusieurs méthodes existent qui appliquent le découplage précédent du problème de planification. Dans ce travail, nous utilisons la technique basée sur l'application du principe de maximum de Pontryagin aux problèmes de commande à temps optimal [12]. Ce principe dit que la commande à temps optimal est Bang-Bang, ce qui signifie qu'elle commute entre des courbes minimales et maximales. Ceci résume son calcul en la recherche des instants de commutations connaissant les courbes limites qui sont obtenues à partir de l'intégration des courbes maximale et minimale de l'accélération \ddot{s} .

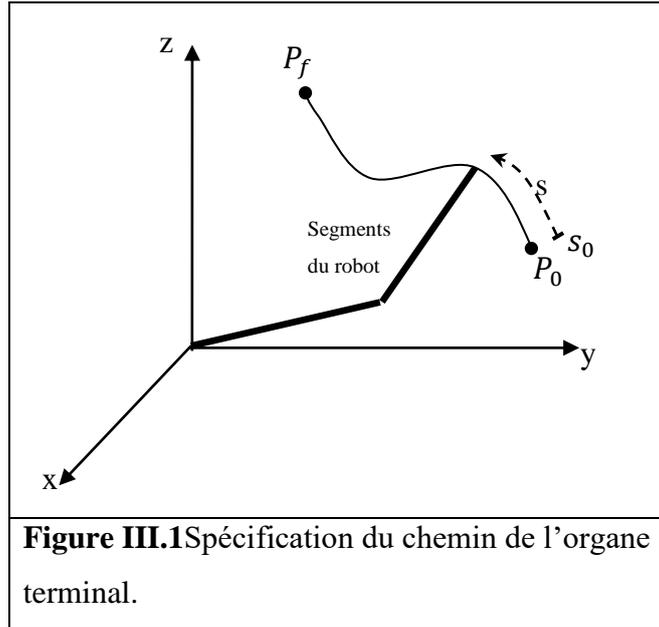
Les points de commutations peuvent être classés en trois groupes : Les singularités dynamiques (les plus communs), les points de discontinuités et les points de tangences (les moins fréquents) [14-16,18]. Ainsi pour disposer d'un algorithme de calcul de la solution optimale, on doit être en mesure de déterminer avec précision ces points de commutation.

Nous allons implémenter, sous MATLAB, un programme qui calcul la solution temps optimal pour la planification de trajectoire d'un robot SCARA. Le chemin de l'outil est spécifié au départ sous forme d'une courbe paramétrique à 3D utilisant un seul paramètre 's'.

III.2 Formulation du problème de commande à temps optimal

III.2.1 Le chemin défini

Le trajectoire définit pour que l'outil de bras manipulateurs de robot le suivre de point initial P_0 a un point final P_f de chemin pour but de déplacer un objet ou manipuler une tache prédéfini pour ce bras de robot la (Figure III.1) représente le chemin de l'outil de robot



Dans le cas générale on veut que le mouvement de robot doit faire déplacer l'outil de la position initial (ou position de repo) P_0 à la position final P_f comme dans la (figure III.1).

s : c'est le paramètre qui donne la distance le long de chemin de point P_0 .

On note que les dynamiques de l'organe terminal sont supposé négligeables par rapport à celle des segments.

Les équations dynamiques de tels robots manipulateurs peuvent être obtenues en utilisant le formalisme de Lagrange [13,19] et ont la forme suivante :

$$M(\theta)\ddot{\theta} + \dot{\theta}^T C(\theta)\dot{\theta} + g(\theta) = \tau \tag{III.1}$$

Les vecteurs θ et τ sont les vecteurs des variables articulaires est des couples articulaires respectivement tandis que $M(\theta)$ est la matrice de masse du robot et $C(\theta)$ est un tenseur ($n \times n \times n$) contenant les paramètres qui caractérisent les couples de Coriolis et centrifuges. $g(\theta)$ sont les forces généralisées (couples) dues à la gravité.

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

$$\tau_{imin}(\theta, \dot{\theta}) \leq \tau_i \leq \tau_{imax}(\theta, \dot{\theta}); \quad i = 1, \dots, n \tag{III.2}$$

Trouver le profil des couples $\tau(t)$ qui mènent le manipulateur de la position initiale P_0 à la position finale P_f en un temps minimal.

Les deux inégalités impliquées dans l'équation (III.2) sont combinées en une seule en procédant de la façon suivante. L'équation (III.2) implique :

$$\begin{aligned} \tau &= M(\theta)\ddot{\theta} + \dot{\theta}^T C(\theta)\dot{\theta} + g(\theta) \leq \tau_{max}(\theta, \dot{\theta}) =; \quad i = 1, \dots, n \\ -\tau &= -M(\theta)\ddot{\theta} - \dot{\theta}^T C(\theta)\dot{\theta} - g(\theta) \leq -\tau_{min}(\theta, \dot{\theta}); \quad i = 1, \dots, n \end{aligned} \quad (III.3)$$

La concaténation de ces deux inégalités en une seule peut se faire en considérant les matrices suivantes [15-18] :

$$\begin{cases} A(\theta) = \begin{bmatrix} M(\theta) \\ -M(\theta) \end{bmatrix}; \quad B(\theta) = \begin{bmatrix} C(\theta) \\ -C(\theta) \end{bmatrix}; \quad f(\theta) = \begin{bmatrix} g(\theta) - \tau_{max}(\theta, \dot{\theta}) \\ -g(\theta) + \tau_{min}(\theta, \dot{\theta}) \end{bmatrix} \\ A(\theta)\ddot{\theta} + \dot{\theta}^T B(\theta)\dot{\theta} + f(\theta) \leq 0 \end{cases} \quad (III.4)$$

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 du point de la position initiale.

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} et \ddot{s} , c'est-à-dire :

$$\begin{aligned} \theta &= \theta(s) \\ \dot{\theta} &= \dot{\theta}(s, \dot{s}) \\ \ddot{\theta} &= \ddot{\theta}(s, \dot{s}, \ddot{s}) \end{aligned} \quad (III.5)$$

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.

III.3 Etapes de calcul des vitesses ($\dot{\theta}(s, \dot{s})$) et accélérations ($\ddot{\theta}(s, \dot{s}, \ddot{s})$) articulaires

III.3.1 Calcul des vitesses ($\dot{\theta}(s, \dot{s})$) articulaires

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

$$r = r(\theta) = r(s) \quad (\text{III.6})$$

La différentiation de cette équation par rapport au temps donne :

$$\dot{r} = \left[\frac{\partial r}{\partial \theta} \right] \dot{\theta} = \frac{dr}{ds} \dot{s} \quad (\text{III.7})$$

Ici $\left[\frac{\partial r}{\partial \theta} \right]$ est le jacobéen des dérivées partiels des composantes de vecteur position de l'outil du robot par rapport aux variables articulaires et $\frac{dr}{ds}$ est un vecteur tangent au chemin de l'outil.

Ainsi quand le manipulateur n'est pas dans une configuration singulière le jacobéen $\left[\frac{\partial r}{\partial \theta} \right]$ est inversible et on peut résoudre l'équation (III.5) pour $\dot{\theta}$:

$$\dot{\theta}(s, \dot{s}) = \left[\frac{\partial r}{\partial \theta} \right]^{-1} \frac{dr}{ds} \dot{s} \quad (\text{III.8})$$

En pratique, les singularités sont évitées.

III.3.2 Calcul de d'accélération ($\ddot{\theta}(s, \dot{s}, \ddot{s})$) articulaires

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

$$\left[\frac{\partial r}{\partial \theta} \right] \ddot{\theta} + \left[\frac{\partial \dot{r}}{\partial \theta} \right] \dot{\theta} = \frac{dr}{ds} \ddot{s} + \frac{d^2r}{ds^2} \dot{s}^2 \quad (\text{III.9})$$

Avec $\left[\frac{\partial \dot{r}}{\partial \theta} \right]$ le dérivé de la matrice jacobéenne par rapport au temps, $\frac{d^2r}{ds^2}$ est le deuxième dérivé de r par rapport à s . Le premier élément du deuxième terme de l'équation (III.7) est l'accélération tangentielle de l'outil \ddot{s} et le deuxième élément est l'accélération normale.

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

$$\ddot{\theta} = \left[\frac{\partial r}{\partial \theta} \right]^{-1} \left(\frac{dr}{ds} \ddot{s} + \frac{d^2r}{ds^2} \dot{s}^2 - \left[\frac{\partial \dot{r}}{\partial \theta} \right] \dot{\theta} \right) \quad (\text{III.10})$$

III.3.3 Les expressions de l'accélération minimale et maximale

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

La substitution de l'équation (III.10) dans (III.4-b) conduit à :

$$A(\theta) \left[\frac{\partial r}{\partial \theta} \right]^{-1} \left(\frac{dr}{ds} \dot{s} + \frac{d^2r}{ds^2} \dot{s}^2 - \left[\frac{\partial \dot{r}}{\partial \theta} \right] \dot{\theta} \right) + \dot{\theta}^T B(\theta) \dot{\theta} + f(\theta) \leq 0 \quad (\text{III.11})$$

Ce qui donne directement :

$$A(\theta) \left[\frac{\partial r}{\partial \theta} \right]^{-1} \frac{dr}{ds} \dot{s} + A(\theta) \left[\frac{\partial r}{\partial \theta} \right]^{-1} \left(r_{ss} \dot{s}^2 - \left[\frac{\partial \dot{r}}{\partial \theta} \right] \dot{\theta} \right) + \dot{\theta}^T B(\theta) \dot{\theta} + f(\theta) \leq 0$$

En prenant en considération :

$$\dot{\theta}(s, \dot{s}) = \frac{\partial \theta}{\partial s} \frac{\partial s}{\partial t} = \theta_s \dot{s}$$

$$\theta_s = \frac{\partial \theta}{\partial s} = \left[\frac{\partial r}{\partial \theta} \right]^{-1} \frac{dr}{ds}$$

$$\begin{aligned} \theta_{ss} \dot{s}^2 &= \left(\frac{\partial}{\partial s} (\theta_s) \right) \dot{s}^2 = \left(\frac{\partial}{\partial s} ([r_\theta]^{-1} r_s) \right) \dot{s}^2 = \left(\frac{\partial}{\partial s} ([r_\theta]^{-1}) r_s + [r_\theta]^{-1} r_{ss} \right) \dot{s}^2 \\ &= \left(-[r_\theta]^{-1} \left(\frac{\partial}{\partial s} [r_\theta] \right) [r_\theta]^{-1} r_\theta \theta_s + [r_\theta]^{-1} r_{ss} \right) \dot{s}^2 \\ &= \left(-[r_\theta]^{-1} \left(\frac{\partial}{\partial \theta} [r_\theta] \theta_s \right) \theta_s + [r_\theta]^{-1} r_{ss} \right) \dot{s}^2 \\ &= [r_\theta]^{-1} \left(- \left(\sum_{i=1}^n r_{\theta\theta}(i, :, :) \dot{\theta}_{si} \right) \dot{\theta} + r_{ss} \dot{s}^2 \right) = [r_\theta]^{-1} \left(- \left[\frac{\partial r}{\partial \theta} \right] \dot{\theta} + r_{ss} \dot{s}^2 \right) \end{aligned}$$

Nous obtenons :

$$\underbrace{A(\theta) \theta_s}_{a(s)} \dot{s} + \underbrace{\{A(\theta) \theta_{ss} + \theta_s^T B(\theta) \theta_s\}}_{b(s)} \dot{s}^2 + \underbrace{f(\theta)}_{c(s)} \leq 0 \quad (\text{III.12})$$

Avec:

$$\begin{cases} a(s) = A(\theta)\theta_s = \begin{bmatrix} M(\theta) \\ -M(\theta) \end{bmatrix} \theta_s = [a_1 \quad \dots \quad a_n \quad -a_1 \quad \dots \quad -a_n]^T \\ b(s) = A(\theta)\theta_{ss} + \theta_s^T B(\theta)\theta_s = \begin{bmatrix} M(\theta) \\ -M(\theta) \end{bmatrix} \theta_s + \theta_s^T \begin{bmatrix} C(\theta) \\ -C(\theta) \end{bmatrix} \theta_s = [b_1 \quad \dots \quad b_n \quad -b_1 \quad \dots \quad -b_n]^T \\ c(s) = f(\theta) = \begin{bmatrix} g(\theta) - \tau_{max}(\theta, \dot{\theta}) \\ -g(\theta) + \tau_{min}(\theta, \dot{\theta}) \end{bmatrix} \end{cases}$$

Vue que pour tout s et \dot{s} sur le chemin de l'outil, $\theta(s)$ et $\dot{\theta}(s, \dot{s})$ peuvent être calculées par le modèle géométrique inverse et l'équation (III.6), τ_{imin} et τ_{imax} qui sont fonctions de θ et $\dot{\theta}$ (équation (III.2)) peuvent être écrits comme fonctions de (s, \dot{s}) .

Alors si $c_{1i}(s) \neq 0$ l'équation (III.12) peut s'écrire comme suit :

$$\alpha_i(s, \dot{s}) \leq \ddot{s} \leq \beta_i(s, \dot{s}) ; i = 1, \dots, n \quad (III.13)$$

Où :

$$\alpha_i(s, \dot{s}) = \begin{cases} \frac{(-c_{i+n}(s) - b_{i+n}(s)\dot{s}^2)}{a_{i+n}(s)} & \text{si } a_i(s) > 0 \\ \frac{(-c_i(s) - b_i(s)\dot{s}^2)}{a_i(s)} & \text{si } a_i(s) < 0 \end{cases} \quad (III.14)$$

$$\beta_i(s, \dot{s}) = \begin{cases} \frac{(-c_i(s) - b_i(s)\dot{s}^2)}{a_i(s)} & \text{si } a_i(s) > 0 \\ \frac{(-c_{i+n}(s) - b_{i+n}(s)\dot{s}^2)}{a_{i+n}(s)} & \text{si } a_i(s) < 0 \end{cases} \quad (III.15)$$

Pour le cas où $a_i(s) = 0$ pour une valeur donnée de s , le point est dit un point singulier. Ceci découle de l'expression de $a(s) = \begin{bmatrix} M(\theta) \\ -M(\theta) \end{bmatrix} \theta_s = \begin{bmatrix} M(\theta) \\ -M(\theta) \end{bmatrix} \left[\frac{\partial r}{\partial \theta} \right]^{-1} \frac{dr}{ds}$ vue que $\frac{dr}{ds}$ est un vecteur unitaire.

Quand $a_i(s) \neq 0$, les équations (III.14) et (III.15) donne 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 $[\alpha_i(s, \dot{s}), \beta_i(s, \dot{s})]$ pour les quels $a_i(s) \neq 0$. Il peut arriver que les intervalles $[\alpha_i(s, \dot{s}), \beta_i(s, \dot{s})]$ 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 $a_i(s) \neq 0$ et que l'intersection des intervalles $[\alpha_i(s, \dot{s}), \beta_i(s, \dot{s})]$ 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 $a_i(s) = 0$ 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 dans une pose singulière, au moins un $a_i(s) \neq 0$, ceci est la conséquence de l'expression de $a_i(s)$ dans le terme de gauche de l'équation (III.10) vu que la matrice de masse est positive et que $\frac{dr}{ds}$ est un vecteur unitaire.

On peut, alors, pour chaque point (s, \dot{s}) définir :

$$\alpha(s, \dot{s}) = \max_p \left(\alpha_p(s, \dot{s}) \right); \quad \beta(s, \dot{s}) = \min_p \left(\beta_p(s, \dot{s}) \right) \quad (\text{III.16})$$

A partir des transformations précédentes, on peut conclure que $\theta(s(t))_{t \in [0, T']}$ satisfait la contrainte (III.2) si et seulement si :

$$\forall t \in [0, T'], \quad \alpha(s(t), \dot{s}(t)) \leq \ddot{s}(t) \leq \beta(s(t), \dot{s}(t)) \quad (\text{III.17})$$

Notons que les deux fonctions $\begin{bmatrix} \dot{s} \\ \alpha(s, \dot{s}) \end{bmatrix}$ et $\begin{bmatrix} \dot{s} \\ \beta(s, \dot{s}) \end{bmatrix}$ peuvent être regardées comme deux champs de vecteurs (ils définissent deux systèmes d'équations différentielles d'ordre 1). Ces deux champs peuvent être intégrés (numériquement) dans le plan (s, \dot{s}) pour donner les profils d'accélération minimale (les profils α) et les profils d'accélération maximale (profils β).

En plus, il faut observer que si $\alpha(s, \dot{s}) \geq \beta(s, \dot{s})$, alors à partir de l'équation (III.17), il n'y a aucune valeur possible de \ddot{s} . Ainsi pour être valable chaque profil de vitesse doit rester en dessous de la courbe de vitesse maximale [15] (MVC) définie par :

$$MVC(s) = \begin{cases} \min\{\dot{s} \geq 0 : \alpha(s, \dot{s}) = \beta(s, \dot{s})\}, & \text{si } \alpha(s, 0) \leq \beta(s, 0) \\ 0, & \text{si } \alpha(s, 0) > \beta(s, 0) \end{cases} \quad (\text{III.18})$$

Il a été montré dans [12] que le profil de vitesse à temps minimal est obtenu par un contrôle Bang-Bang dans lequel le profil optimal suit de façon alternée les champs α, β en restant toujours en dessous de la MVC.

Enfin, la version mathématique du problème de commande à temps optimal peut être énoncée comme suit :

Etant donné $s(0)$ et $\dot{s}(0)$; choisir $\ddot{s}(t), t \in [0, t_f]$: $\alpha(s(t), \dot{s}(t)) \leq \ddot{s}(t) \leq \beta(s(t), \dot{s}(t))$ et que $s(t_f) = s_f$ et $\dot{s}(t_f) = \dot{s}_f$ et que $t_f = \int_{s_0}^{s_f} \frac{ds}{\dot{s}}$ soit minimisé.

III.4 Solution du problème de paramétrisation à temps optimal du chemin

Il a été montré dans [15] que pour minimiser t_f la commande doit être Bang-Bang, c'est-à-dire que l'accélération doit toujours prendre soit sa valeur maximale $\ddot{s} = \beta(s, \dot{s})$ soit sa valeur minimale $\ddot{s} = \alpha(s, \dot{s})$. Ainsi, trouver la loi de commande optimale revient à trouver les instants ou positions auxquels \ddot{s} commute entre la courbe minimale et la courbe maximale. Depuis l'introduction du problème de paramétrisation à temps optimal, beaucoup d'algorithmes ont été développés pour le résoudre. Dans ce travail, nous allons adopter celui de [12,15]. Dans les paragraphes qui suivent nous allons exposer les étapes de la procédure.

Etape 1 :

Dans le plan (s, \dot{s}) commencer à partir de $(s = s_0, \dot{s} = \dot{s}_0)$ et intégrer l'accélération maximale $\beta(s, \dot{s})$ vers l'avant jusqu'à atteindre :

- La courbe de vitesse maximale MVC dans ce cas allé à l'étape 2.
- La ligne horizontale $\dot{s} = 0$, dans ce cas le chemin n'est pas dynamiquement traversable.
- La ligne verticale $s = s_f$, dans ce cas allé à l'étape 3.

Etape 2 :

A partir du point d'intersection de la courbe obtenue par l'intégration vers l'avant de l'accélération maximale $\beta(s, \dot{s})$ et de la MVC, rechercher vers l'avant le long de la MVC, le prochain point de commutation $\alpha \rightarrow \beta$. A partir de ce point :

- Intégrer vers l'arrière le long de l'accélération minimale $\alpha(s, \dot{s})$ jusqu'à ce que la courbe obtenue coupe le profil β (de l'étape 1 ou de façon récursive de l'étape 2 courante). Le point d'intersection constitue un point de commutation $\beta \rightarrow \alpha$.
- Intégrer vers l'avant le long de l'accélération maximale $\beta(s, \dot{s})$, ensuite continuer comme dans l'étape 1.
- Si à partir d'un point de commutation donné, on ne peut pas intégrer α vers l'avant et/ou β vers l'arrière sans instantanément violer les contraintes alors ce point sera ignoré. Il faut noter que le nombre de point de commutation doit être impair.
- Si la condition de tangence est vérifiée sur un intervalle fini, alors la trajectoire dans le plan de phase coïncide avec la MVC le long d'un arc fini.

Le profil de vitesse optimale obtenu jusqu'ici est constitué par la concaténation des courbes obtenues par l'intégration vers l'avant β_{min} les obtenues par l'intégration vers l'arrière de α_{max} .

Etape 3 :

Démarrer à partir de $(s = s_f, \dot{s} = \dot{s}_f)$ et intégrer vers l'arrière le long de α_{max} jusqu'à intersection avec un profil direct β obtenu dans les étapes 1 ou 2. Le point d'intersection constitue un point de commutation $\beta \rightarrow \alpha$. Le profil optimal final sera obtenu par la concaténation du profil direct coupé et du profil α qui vient d'être calculé.

Si on examine les trois étapes de l'algorithme précédent, on peut conclure que la recherche des points de commutation $\alpha \rightarrow \beta$, qui sont situés sur la MVC, est cruciale pour l'approche par intégration numérique.

Dans [12], il a été montré que les points de commutation situés sur la MVC (commutation $\alpha \rightarrow \beta$) doivent vérifier une certaine condition de tangence pour que le profil optimal de vitesse passe par la MVC sans la couper pour entrer dans la zone inadmissible. Cette condition de tangence peut servir pour la détection de ces points de commutation. Mais elle n'est pas assez stable numériquement. Dans [14,15], les points de commutation situés sur la MVC (commutation $\alpha \rightarrow \beta$) sont groupés en trois classes:

1. Points de commutation discontinus : C'est les points s auxquels la MVC est discontinue. Il a été montré dans [14] que la MVC est discontinue à un point s si et seulement si r_{ss} ou θ_{ss} sont discontinues à ce point s . Ceci permet d'obtenir les points de discontinuité sans calculer la MVC mais il faut calculer r_{ss} ou θ_{ss} .
2. Points de singularités dynamiques : C'est les points auxquels la MVC est continue mais non différentiable. Dans ces points au moins une des composantes du vecteur $a(s)$ est nulle. Ici, il faut noter qu'il peut y avoir des points ayant un $a_i(s) = 0$ mais qui ne constituent pas des points de singularité dynamique. Une condition nécessaire pour qu'un point soit un point de singularité dynamique est que l'une des composantes du vecteur $\theta_s(s) \left[\frac{\partial \theta}{\partial s} \right]_i (s) = \theta_{si}(s) = 0$. Les points qui ont une composante $a_i(s) = 0$ sont appelés des points critiques.
3. Points de commutation tangents : C'est des points auxquels la MVC est continue et différentiable et le vecteur tangent au MVC à $(s, MVC(s))$ est colinéaire avec le

vecteur $(MVC(s), \alpha(s, MVC(s)))$ (ou bien ce qui est le même puisqu'on est sur la MVC à $(MVC(s), \beta(s, MVC(s)))$). En d'autre terme la pente de la tangente à ces points de commutation est $\frac{d\dot{s}}{ds} = \frac{\alpha}{\dot{s}} = \frac{\beta}{\dot{s}}$.

Le calcul des points de commutation discontinus est facile car ça revient à déterminer les points auxquels la MVC est discontinue, ceci est réalisé en examinant tous les échantillons entre s_0 et s_f .

Pour les points de commutation tangents à la MVC qui sont les moins fréquents parmi les trois ensembles, une procédure a été établie dans [14] pour les obtenir. Dans ce travail nous avons utilisé la relation introduite dans [14] et présenté aussi dans [15] pour les calculer. Elle consiste à chercher les points sur la MVC solutions de l'équation :

$$\kappa(s) = \frac{d\dot{s}}{ds} - \frac{\alpha(s, \dot{s})}{\dot{s}} = \frac{d\dot{s}}{ds} - \frac{\beta(s, \dot{s})}{\dot{s}} = 0 \quad (\text{III.19})$$

Dans cette relation $\frac{d\dot{s}}{ds}$ se calcul numériquement.

L'étape la plus difficile est la détermination et le traitement des points de singularité dynamique. Une procédure spéciale a été introduite dans [14,16 ,17] pour leur traitement. Nous allons dans ce qui suit exposer les points nécessaires.

Au début il faut rappeler que les points de singularité dynamique sont parmi les points qui ont une des composantes $a_k(s)$ qui est nulle. Cependant, la recherche des valeurs de s , qui ont cette propriété, n'est pas suffisante car seuls les points auxquels la MVC n'est pas différentiable constituent des points singuliers.

Considérons un point s^* qui vérifie $a_k(s^*) = 0$. Sans perte de généralité nous supposons que $a_k(s)$ passe par zéro à s^* en croissant. C'est-à-dire que $a_k(s) < 0$ à gauche de s^* et $a_k(s) > 0$ à droite de s^* (le cas où $a_k(s^*)$ est décroissante peut être traité similairement).

Il a été montré dans [15] que si le chemin est traversable, alors $c_k(s^*) < 0$. Supposons que cette condition est vérifiée, alors on distingue deux cas :

1. Si $b_k(s^*) > 0$. Sur la MVC, on a $a_k(s^*)\dot{s}^* + b_k(s^*)\dot{s}^{*2} + c_k(s^*) = 0$ et puisque $a_k(s^*) = 0$, on aura : $\dot{s}^* = \sqrt{-\frac{c_k(s^*)}{b_k(s^*)}}$. Ensuite, supposons que \dot{s}^+ est la valeur de la MVC si on ignore la contrainte k. Il a été montré dans [15] que :
 - a. Si $\dot{s}^+ < \dot{s}^*$, alors la contrainte k ne déclenche pas une singularité dynamique à s^* .
 - b. Si $\dot{s}^+ > \dot{s}^*$, alors la contrainte k déclenche une singularité dynamique à s^* .
2. Si $b_k(s^*) < 0$, alors la contrainte k ne déclenche pas une singularité dynamique à s^* .

La prochaine difficulté consiste en la sélection de l'accélération pour lancer l'intégration en directe et vers l'arrière à partir de la singularité dynamique. En effet, les fonctions α et β ne sont pas définies en ces points à cause de la division par $a_k(s^*) = 0$. Il a été montré dans [15] qu'un profil α peut atteindre le point (s^*, \dot{s}^*) seulement si son vecteur tangent au point d'intersection pointe vers (s^*, \dot{s}^*) . Ceci peut être accomplie à gauche de s^* , s'il existe un nombre λ tel que :

$$\lambda = \frac{\alpha_k(s^* - \varepsilon, \dot{s}^* - \lambda\varepsilon)}{\dot{s}^* - \lambda\varepsilon} = \frac{-b_k(s^* - \varepsilon)[\dot{s}^* - \lambda\varepsilon]^2 - c_k(s^* - \varepsilon)}{a_k(s^* - \varepsilon)[\dot{s}^* - \lambda\varepsilon]} \quad (\text{III.20})$$

L'utilisation du développement en série de Taylor des termes de l'expression précédente permet d'obtenir :

$$\lambda = -\frac{b'_k(s^*)\dot{s}^{*2} + c'_k(s^*)}{[2b_k(s^*) + a'_k(s^*)]\dot{s}^*} \quad (\text{III.21})$$

La même formule peut être obtenue à droite de s^* en considérant le point $(s^* + \varepsilon, \dot{s}^* + \lambda\varepsilon)$ et en résolvant l'équation suivante en utilisant le développement en série de Taylor :

$$-\lambda = \frac{\beta_k(s^* + \varepsilon, \dot{s}^* + \lambda\varepsilon)}{\dot{s}^* + \lambda\varepsilon} = \frac{-b_k(s^* + \varepsilon)[\dot{s}^* + \lambda\varepsilon]^2 - c_k(s^* + \varepsilon)}{a_k(s^* + \varepsilon)[\dot{s}^* + \lambda\varepsilon]} \quad (\text{III.22})$$

Maintenant nous donnons l'algorithme du traitement qu'il faut faire chaque fois qu'on rencontre un point avec $a_k(s^*) = 0$ avec $a_k(s) < 0$ à gauche de s^* et $a_k(s) > 0$ à droite de s^* .

1. Si $b_k(s^*) < 0$, alors la contrainte k ne déclenche pas une singularité dynamique.
2. Si $b_k(s^*) > 0$, alors calculer \dot{s}^* à partir de l'expression $\dot{s}^* = \sqrt{-\frac{c_k(s^*)}{b_k(s^*)}}$ et \dot{s}^+ en calculant la MVC à s^* après avoir ignoré la contrainte k.
 - a. Si $\dot{s}^* > \dot{s}^+$, alors la contrainte k ne déclenche pas une singularité dynamique.

- b. Si $\dot{s}^* < \dot{s}^+$, alors s^* est une singularité dynamique. Alors, il faut calculer la valeur de λ en utilisant l'équation (III.21) et :
 - i. Intégrer la fonction constante $(\dot{s}^*, \lambda \dot{s}^*)$ vers l'arrière pour un petit nombre de période d'échantillonnage. Ensuite continuer en suivant α comme avec l'algorithme précédent.
 - ii. Intégrer le champ constant $(\dot{s}^*, \lambda \dot{s}^*)$ vers l'avant (en directe) pour un petit nombre de périodes d'échantillonnage de s .

Notons qu'après être éloigné du point s^* avec un nombre limité de périodes d'échantillonnage, les champs α, β deviennent lissent de telle sorte qu'il ne reste pas de problème pour l'intégration.

III.5 Résultats de Simulation

Nous allons, maintenant, appliquer la méthode de planification de trajectoire à temps optimal à un robot SCARA dont l'outil doit suivre un chemin sous forme d'une ellipse en gardant la direction de l'organe terminale constante. Nous donnons ici les relations principales concernant la géométrie et les dynamiques du robot et nous renvoyons le lecteur à [20] pour plus de détails.

Le robot SCARA est un robot à 4 dls dont les articulations sont ordonnées comme RRRP. La figure suivante représente ce robot dans sa position de repos avec les repères attachés aux différents segments. On suppose que les centres de masses des trois premiers segments se trouvent sur le même niveau r_0 .

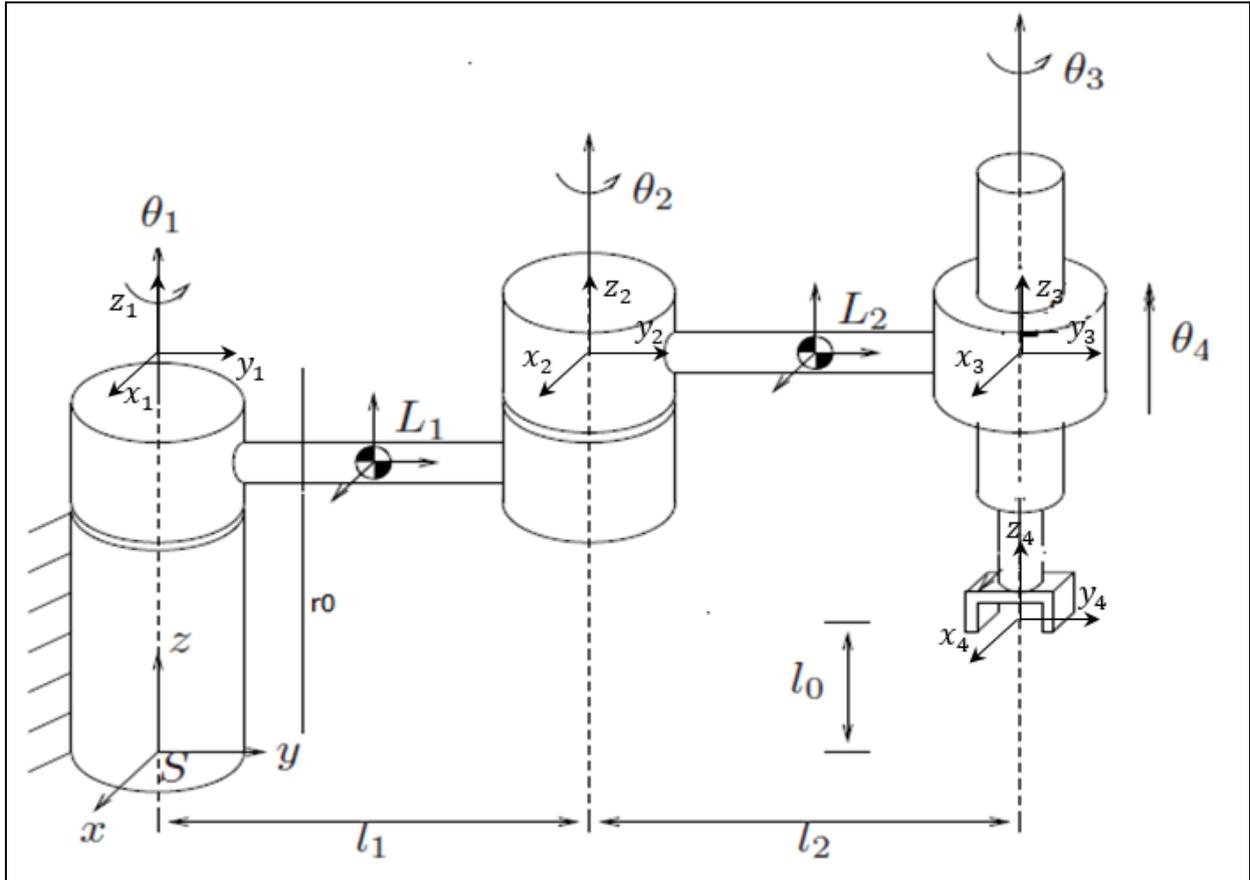


Figure III.2 Robot SCARA [20]

Les matrices de transformations homogènes entre les différentes repères sont données par :

$${}^B_1T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & r_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^2_1T = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.23)$$

$${}^3_2T = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^4_3T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -(r_0 - l_0) + \theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.24)$$

La transformation homogène qui permet le passage entre le repère de l'outil et celui de la base est donnée par (Transforme les coordonnées du repère de l'outil vers le repère de la base :

$$\begin{aligned}
 {}^E T = {}^B T_1 T_2 T_3 T_4 T = \\
 \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ 0 & 0 & 1 & l_0 + \theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & x \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & 0 & H_{14} \\ H_{21} & H_{22} & 0 & H_{24} \\ 0 & 0 & 1 & H_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.25)
 \end{aligned}$$

Les équations des dynamique du robot sont données par :

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \quad (III.26)$$

Avec $\tau = [\tau_1 \ \tau_2 \ \tau_3 \ \tau_4]^T$, $\theta = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4]^T$. La matrice $M(\theta)$ est la matrice de masse du robot. C'est une matrice qui caractérise les inerties du robot elle est symétrique et définie strictement positive.

$M(\theta) =$

$$\begin{bmatrix} \left\{ \begin{array}{l} \left(\frac{1}{4} m_1 + m_2 + m_3 + m_4 \right) l_1^2 \\ + \left(\frac{1}{4} m_2 + m_3 + m_4 \right) l_2^2 \\ + I_{1zz} + I_{2zz} + I_{3zz} + I_{4zz} \\ + 2l_1 l_2 \left(\frac{1}{2} m_2 + m_3 \right) \cos(\theta_2) \end{array} \right\} & \left\{ \begin{array}{l} + \left(\frac{1}{4} m_2 + m_3 \right) l_2^2 \\ + m_4 \\ + I_{2zz} + I_{3zz} + I_{4zz} \\ + l_1 l_2 \left(\frac{1}{2} m_2 + m_3 \right) \cos(\theta_2) \end{array} \right\} & \{ I_{3zz} + I_{4zz} \} & 0 \\ \left\{ \begin{array}{l} \left(\frac{1}{4} m_2 + m_3 + m_4 \right) l_2^2 \\ + \left(\frac{1}{2} m_2 + m_3 \right) l_1 l_2 \cos(\theta_2) \\ + I_{2zz} + I_{3zz} + I_{4zz} \\ + (I_{3zz} + I_{4zz}) \end{array} \right\} & \left\{ \begin{array}{l} \left(\frac{1}{4} m_2 + m_3 \right) l_2^2 \\ + m_4 \\ + I_{2zz} + I_{3zz} + I_{4zz} \end{array} \right\} & (I_{3zz} + I_{4zz}) & 0 \\ 0 & (I_{3zz} + I_{4zz}) & (I_{3zz} + I_{4zz}) & 0 \\ & 0 & 0 & m_4 \end{bmatrix}$$

(III.27)

La matrice $C(\theta, \dot{\theta})$ est linéaire en $\dot{\theta}$ et elle caractérise les force centrifuges et les effets Coriolis. Elle est donnée par :

$$C(\theta, \dot{\theta}) = \begin{bmatrix} -l_1 l_2 \left(\frac{1}{2} m_2 + m_3 + m_4\right) \sin(\theta_2) \dot{\theta}_2 & -l_1 l_2 \left(\frac{1}{2} m_2 + m_3 + m_4\right) \sin(\theta_2) (\dot{\theta}_1 + \dot{\theta}_2) & 0 & 0 \\ \left(\frac{1}{2} m_2 + m_3 + m_4\right) l_1 l_2 \sin(\theta_2) \dot{\theta}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(III.27)

La matrice $G(\theta)$ caractérise les forces de gravité

$$G(\theta) = [0 \quad 0 \quad 0 \quad m_4 g]^T \quad (III.28)$$

Cette matrice est remplacée par une matrice $N(\theta, \dot{\theta})$ qui dépend aussi de $\dot{\theta}$ pour le cas où on considère les forces de frottement. Dans ce cas le modèle dynamique sera décrit par :

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) \quad (III.29)$$

Le chemin de l'outil est paramétré par une variable réelle $s \in [0,1]$ comme suit :

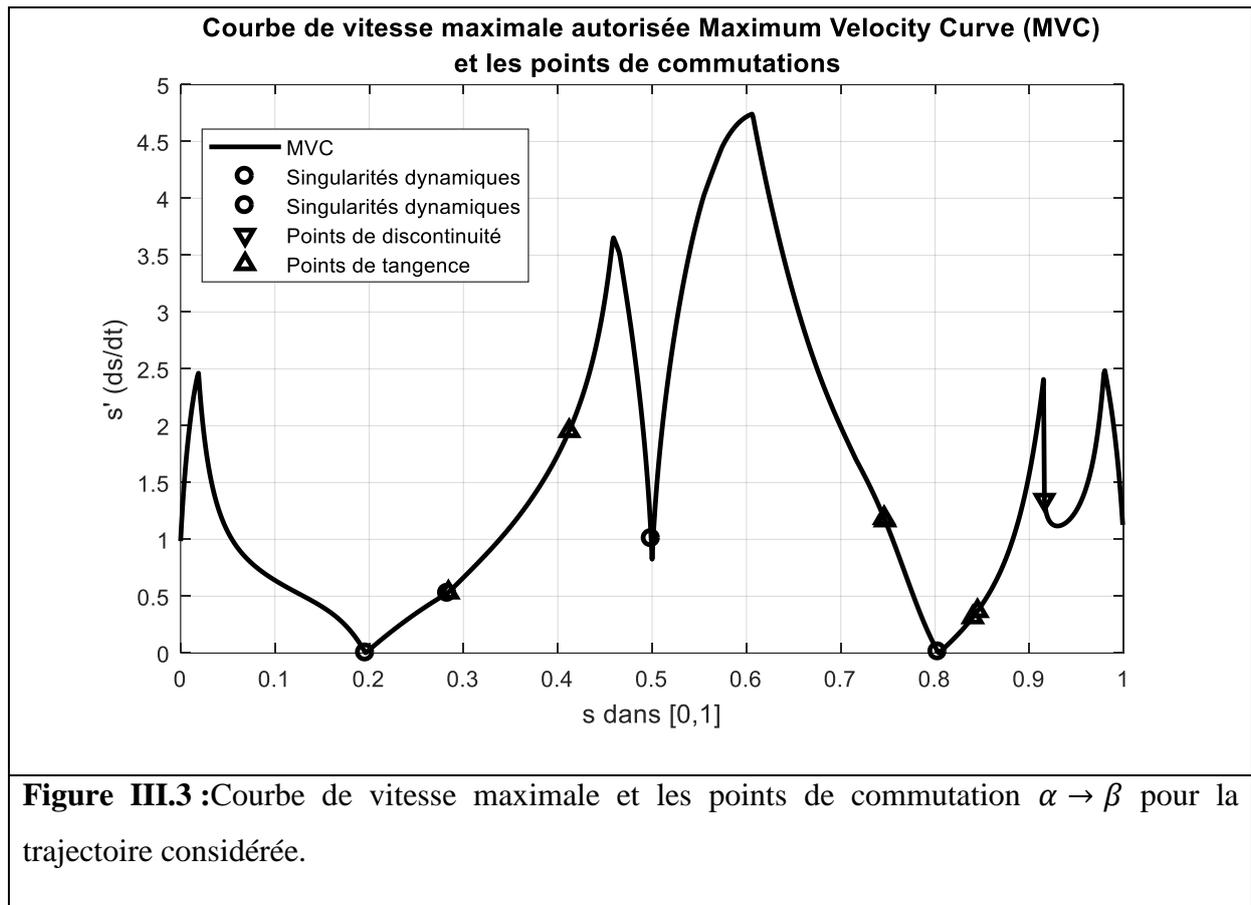
La première trajectoire est donnée par les équations suivantes :

$$\begin{cases} x = a_x \cos(2\pi s) \\ y = 0.3 \\ z = a_z \sin(2\pi s) + 0.2 \\ \theta_1 + \theta_2 + \theta_3 = 0 \text{ L'orientation de l'outil est constante} \end{cases} \quad (III.30)$$

Nous ne présentons pas ici les relations obtenues par dérivation pour l'implémentation des relations (III.4 à 18) dans le programme de simulation.

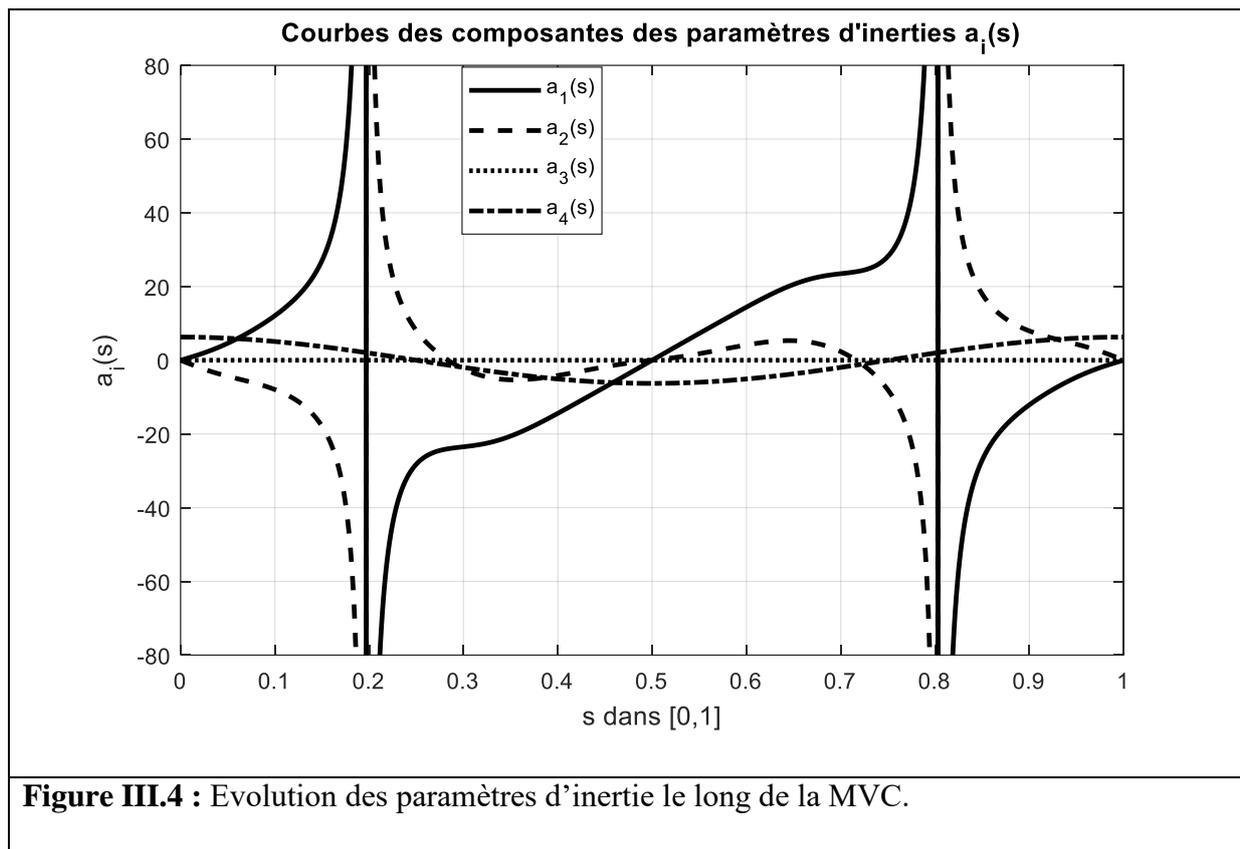
Les étapes de simulation:

La première tâche consiste à calculer la courbe de vitesse maximale permise (Maximum Velocity Curve MVC), ensuite, de déterminer les points de commutation le long de cette courbe. Ceci est représenté sur la figure suivante.

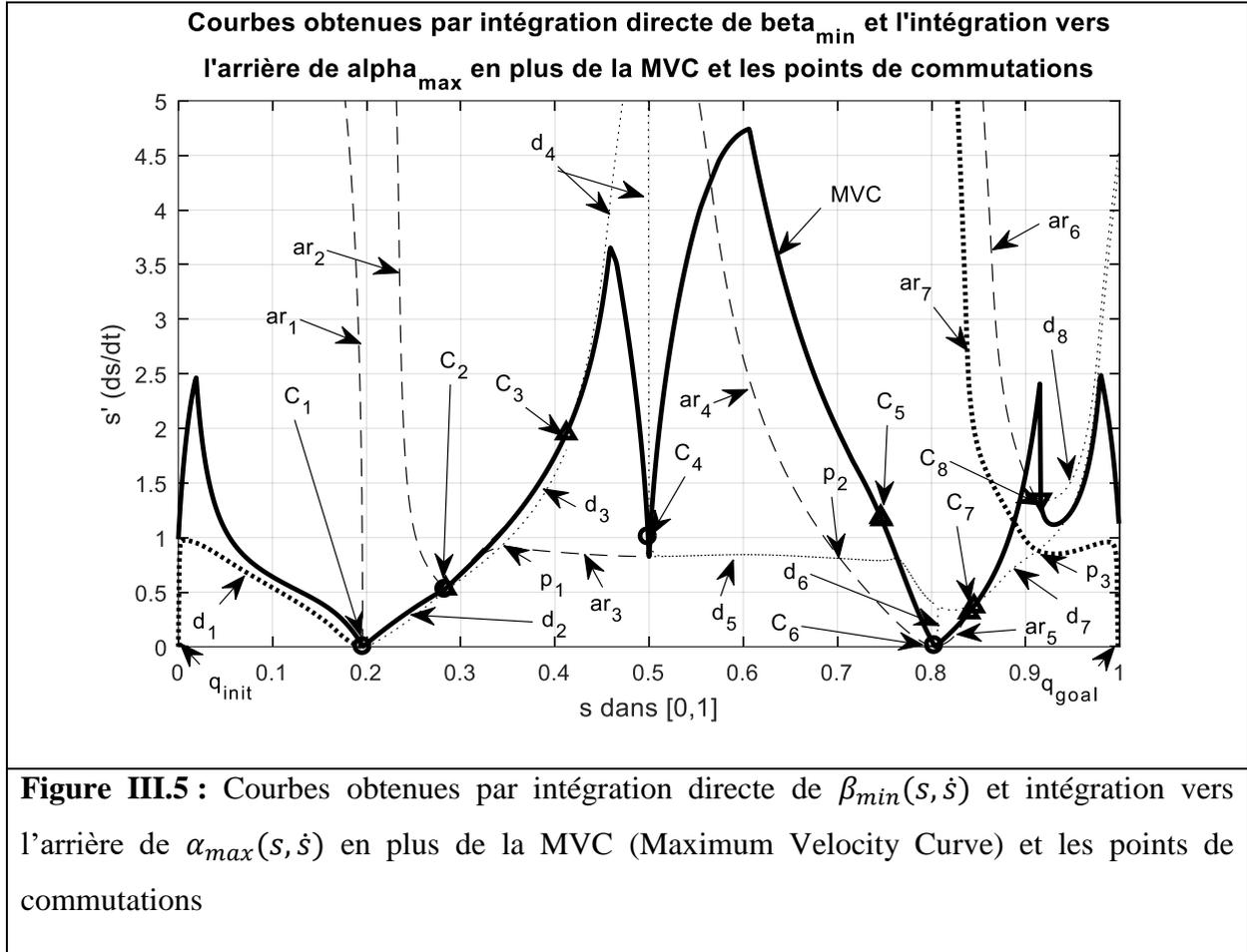


On remarque qu'il y a 4 points de singularités dynamiques dont un est aussi un point de tangence. Il y a un seul point de discontinuité qui comme nous allons le voir après ne jouera aucun rôle dans la détermination du profil optimal de vitesse pour cette trajectoire.

La figure suivante montre l'évolution des paramètres d'inertie du vecteur $a(s)$. On remarque que certains points parmi ceux où un paramètre $a_i(s)$ s'annule correspondent à des singularités dynamiques. Mais il y en a d'autres qui ne vérifient les conditions de tels points. On remarque aussi que la troisième composante est identiquement nulle et que la quatrième composante a des valeurs très petites comparées aux autres paramètres.



En appliquant le principe de maximum de Pontryagin, on sait que la commande à temps minimale est Bang-Bang. Ceci réduit le calcul de la trajectoire optimale à la recherche des points de commutations entre l'accélération minimale et l'accélération maximale ensuite l'intégration de ces dernières sur leurs intervalles. La figure suivante représente les résultats de ces intégrations. Cette figure est utilisée pour le calcul de la solution à temps optimale.



Les points C_1, C_2, C_4, C_6 sont des points de singularité dynamique à inertie nulle. Le point C_2 est aussi un point de tangence.

Les points C_2, C_3, C_5, C_7 sont des points de tangence. Ceci est confirmé pour C_2, C_3 et C_7 si on examine de près les courbes obtenues par intégration vers l'avant de β_{min} et vers l'arrière de α_{max} . Le point C_5 sera ignoré car ces courbes coupent la MVC seulement après ce point.

Le point C_8 est un point de discontinuité qui sera ignoré car la courbe émanant du point final par intégration vers l'arrière de $\alpha_{max}(s, \dot{s})$ coupe la MVC avant ce point.

Les courbes $d_1, d_2, d_3, d_4, d_5, d_6, d_7$ et d_8 sont obtenues par intégration directe de $\beta_{min}(s, \dot{s})$ à partir des points $(s = 0, \dot{s} = 0)$, $C_1, C_2, C_3, C_4, C_6, C_7$ et C_8 respectivement.

Les courbes $ar_1, ar_2, ar_3, ar_4, ar_5$ et ar_6 sont obtenues par intégration vers l'arrière de l'accélération minimale $\alpha_{min}(s, \dot{s})$ à partir des points $C_1, C_2, C_4, C_6, C_7, C_8$ respectivement.

Pour ar_7 elle est obtenue par intégration de $\alpha_{min}(s, \dot{s})$ à partir du point de destination q_{goal} . La courbe obtenue par intégration vers l'arrière à partir du point C_3 est confondue avec la MVC et n'apparaît pas sur la figure et vue qu'elle est située du côté supérieur à la MVC elle est ignorée ici.

Pour obtenir le profil de vitesse optimale (à temps minimal) à partir de ces courbes nous procédons comme suit :

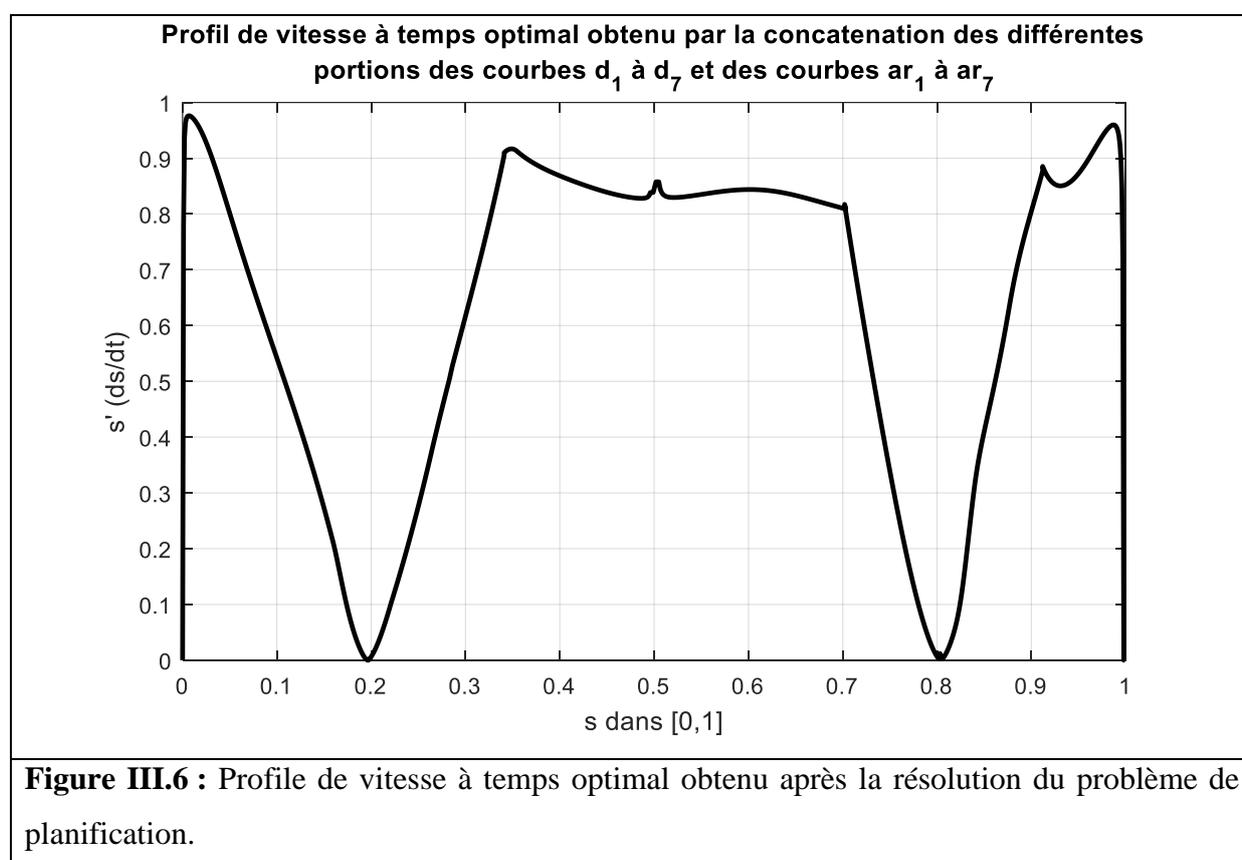
A partir du point initial $q_{init} = (s_0, \dot{s}_0)$, on prend la courbe d_1 qui est obtenue par intégration de l'accélération maximale autorisée $\beta_{min}(s, \dot{s})$ et ceci jusqu'au premier point de singularité dynamique C_1 (premier point de contact entre la MVC et la courbe d_1). Ensuite, on prend la courbe d_2 obtenue aussi par intégration de l'accélération maximale $\beta_{min}(s, \dot{s})$ ainsi le point C_1 n'est pas un point de commutation au sens propre du mot. Les courbes ar_1 et ar_2 n'ont aucun rôle ici car elles sont situées complètement au-dessus de la MVC.

Juste après le point de contact C_2 on suit aussi la courbe d_3 obtenue elle aussi par intégration de l'accélération maximale $\beta_{min}(s, \dot{s})$ à partir de C_2 . Là aussi, on remarque que C_2 n'est pas un point de commutation pour cette trajectoire. On continue sur la courbe d_3 jusqu'au point p_1 de son intersection avec la courbe ar_3 obtenue par intégration de l'accélération minimale $\alpha_{max}(s, \dot{s})$ à partir du point C_4 . Ce point d'intersection est effectivement un point de commutation. La trajectoire optimale continue sur la courbe ar_3 jusqu'au point C_4 . Après le point C_4 , on suit la courbe d_5 obtenue par intégration de l'accélération maximale $\beta_{min}(s, \dot{s})$ après le point C_4 et ceci tant qu'elle est située en dessous de la courbe ar_4 . Ainsi le point C_4 est un point de commutation.

Après le point d'intersection p_2 des courbes d_4 et ar_4 , on poursuivra la courbe ar_4 car elle est située en dessous de la courbe d_5 qui elle va couper la MVC et sortir du domaine permis par contre la courbe ar_4 est tangente à la MVC au point C_6 où l'on commutera vers la courbe ar_5 obtenue elle aussi par intégration de l'accélération minimale $\alpha_{max}(s, \dot{s})$. Ainsi le point p_2 d'intersection des courbes d_5 et ar_4 est un point de commutation par contre le point C_6 n'est pas un point de commutation pour cette trajectoire. Il faut aussi noter que le point C_5 qui est un point de tangence a été complètement ignoré car il donne des courbes qui couperont la MVC et conduiront en dehors de la zone permise. Au point C_7 on passe de la courbe ar_5 vers la courbe d_7 obtenue par intégration de l'accélération maximale $\beta_{min}(s, \dot{s})$ jusqu'à son intersection p_3 avec la courbe ar_7 . Ainsi le point C_7 et le point d'intersection p_3 entre d_7 et ar_7

sont les deux des points de commutation. Il faut noter que le point C_8 qui est un point de discontinuité est ignoré car il donnera une courbe qui dépasse la MVC ou en général qui dépasse le point terminal q_{goal} .

En procédant à la concaténation des portions de courbes comme discutées ci-avant nous avons obtenu le profil de la vitesse à temps optimal représenté par la figure suivante. Les pics qui apparaissent aux points de singularité ont été ignorés. Maintenant, à partir de ce profil, il faut procéder au calcul des variables articulaires et de leurs dérivées ainsi que des couples articulaires.



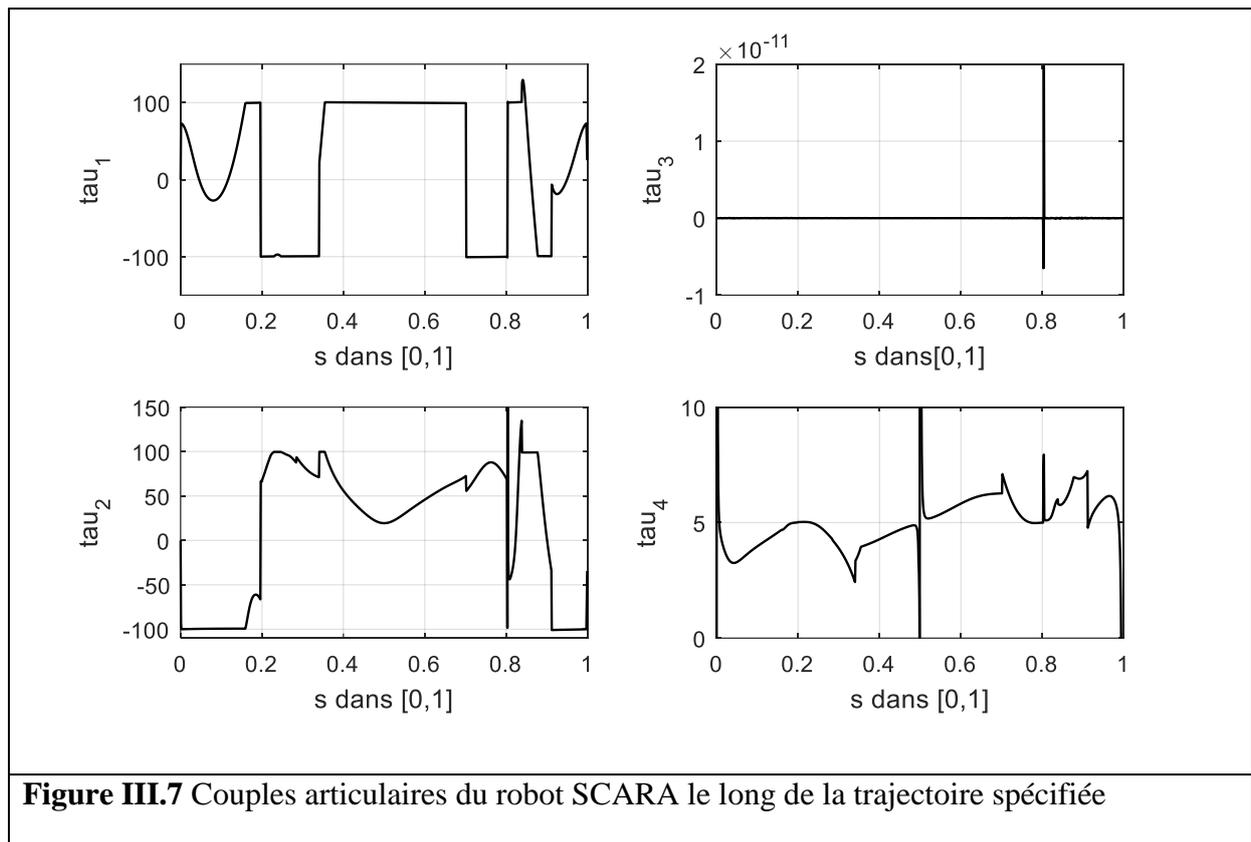
Jusqu'ici nous avons paramétré l'évolution par le paramètre s est le temps n'a pas été utilisé. Pour revenir vers la représentation en fonctions du temps, il faut tirer la relation entre le temps et le paramètre s . Cette relation est donnée par :

$$t = \int_0^t dt = \int_{s_0}^s \frac{ds}{\left(\frac{ds}{dt}\right)} = \int_{s_0}^s \frac{ds}{\dot{s}} \quad (\text{III.31})$$

Ceci signifie qu'il faut faire l'intégral de la fonction donnée dans la figure précédente (le profil de vitesse temps optimal). Mais en examinant cette courbe, on voit que \dot{s} passe par

zéro en un ensemble finie de points. Nous avons approximé cet intégral en utilisant la méthode trapézoïdale et en ignorant les points où \dot{s} s'annule et nous avons obtenu $t_f = 15.6128\text{sec}$. Cette valeur n'est que approximative vue que \dot{s} s'annule.

Dans la figure qui suit nous donnons les couples articulaires. Ces derniers ont été obtenus en utilisant l'équation des dynamiques du robot exprimées en utilisant la variable s .



On remarque que seules les articulations fournissent un effort significatif. On remarque aussi qu'à chaque instant l'une des deux articulations 1 ou 2 est saturée c'est-à-dire qu'elle fournit l'effort maximal. On voit aussi que l'articulation 3 est pratiquement inactive (c'est une articulation rotoïdes pure et son rôle est d'orienter l'organe terminale).

A titre indicatif, nous donnons ces mêmes courbe en fonction du temps cette fois-ci approximé en ignorant les points où $\dot{s} = 0$.

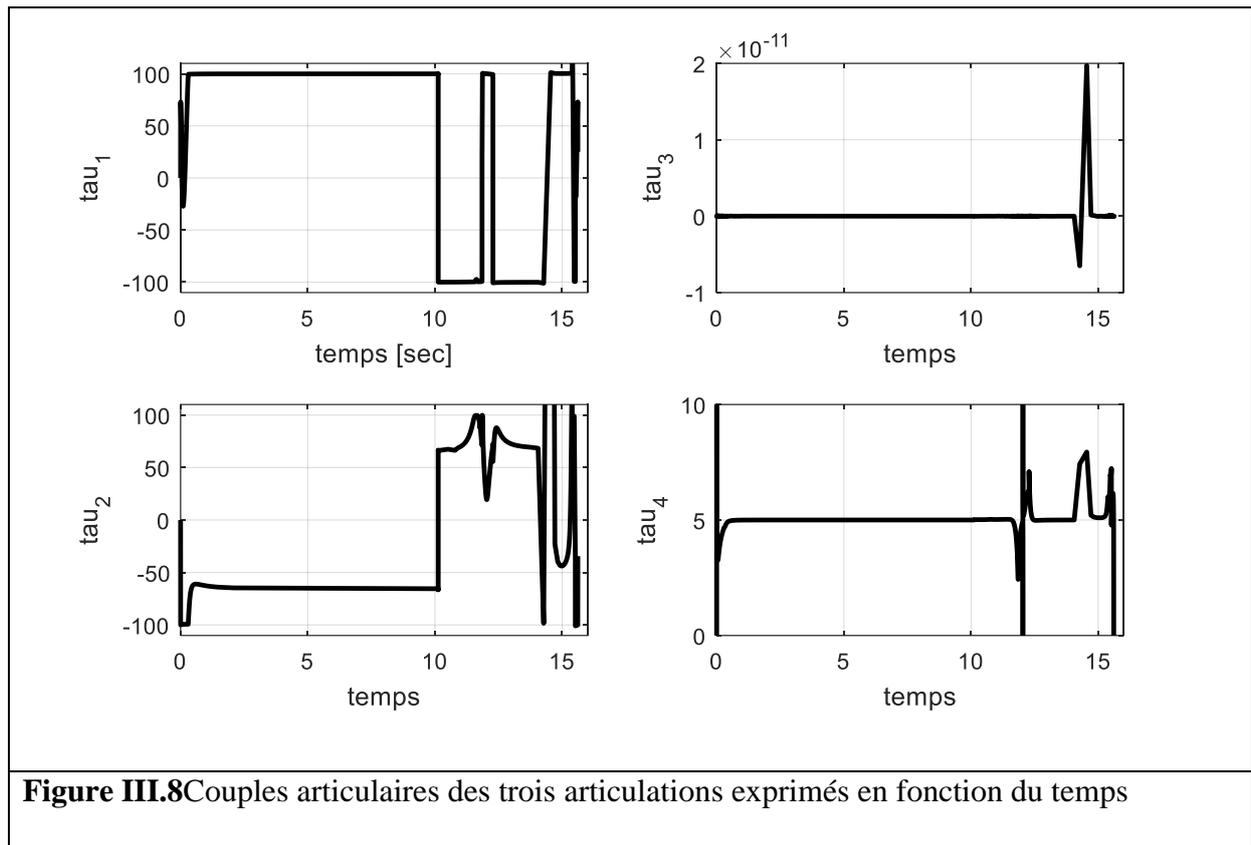


Figure III.8 Couples articulaires des trois articulations exprimés en fonction du temps

Dans la figure qui suit, nous donnons les coordonnées articulaires en fonction du temps le long de la trajectoire optimale.

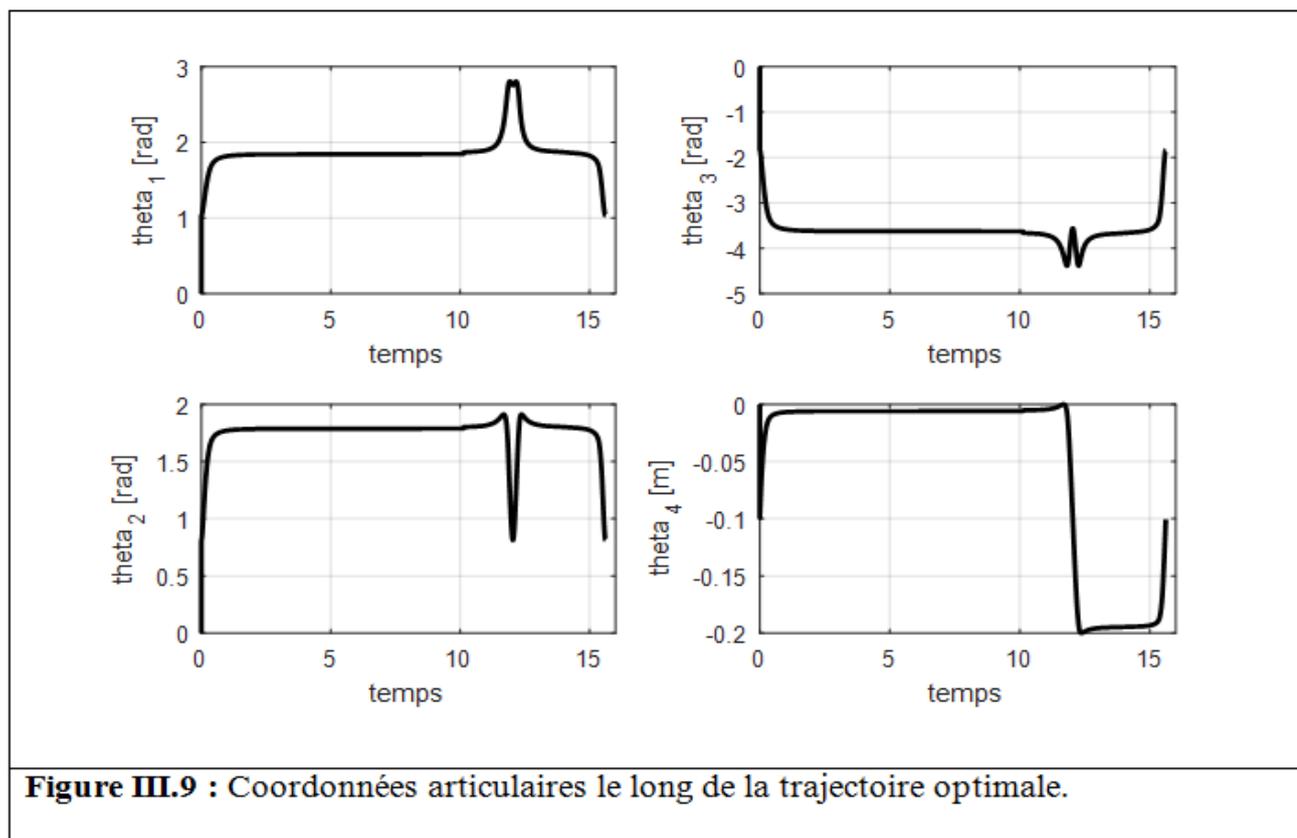
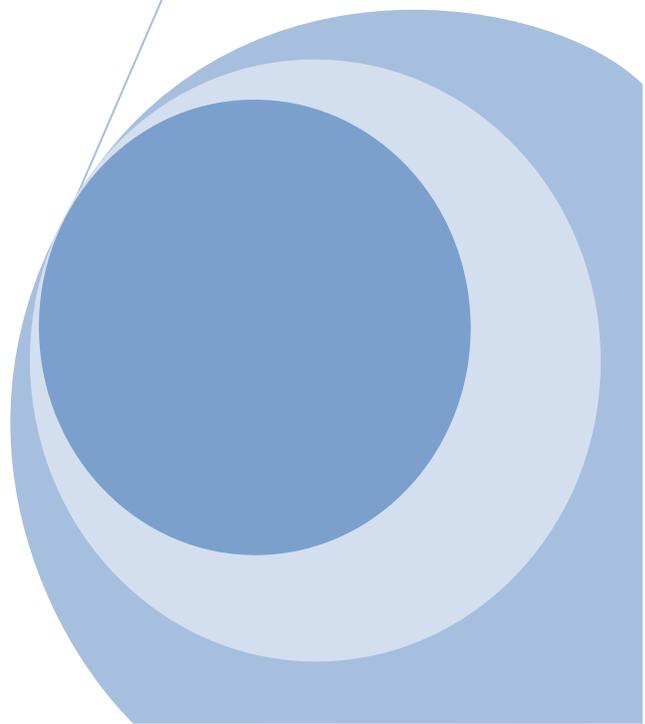
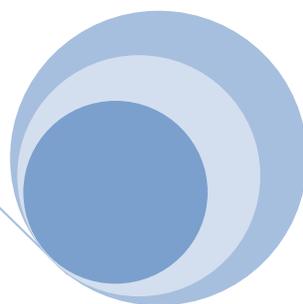
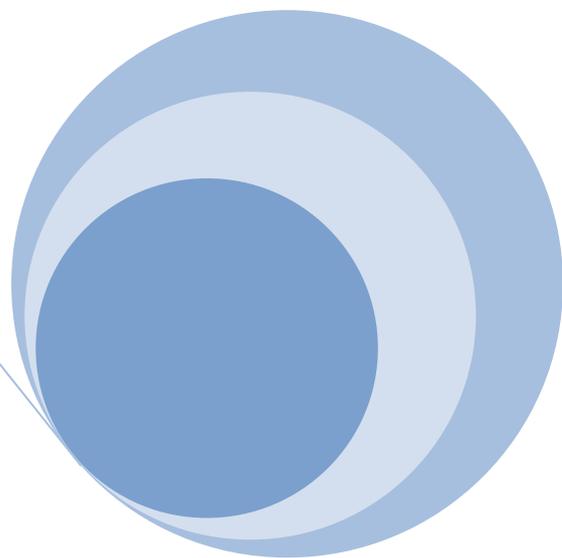


Figure III.9 : Coordonnées articulaires le long de la trajectoire optimale.

III.6 Conclusion

Dans ce chapitre nous avons implémenté sous MATLAB un programme qui calcul la trajectoire à temps optimal pour un robot SCARA en partant d'un chemin spécifié pour l'organe terminal en termes d'un paramètre réel $0 \leq s \leq 1$. Le programme est basé sur le fait qu'une telle solution est Bang-Bang. Ainsi, les calculs sont concentrés sur l'obtention de la courbe de vitesse maximale MVC ensuite à détermination des points de commutation entre l'accélération minimale et l'accélération maximale le long de cette courbe. Ce principe donne l'un des algorithmes les plus rapides pour résoudre ce problème. Cependant, la méthode a un problème de robustesse car pour certains problèmes de planification certains points de singularité conduisent à l'échec de l'algorithme à trouver la solution et la méthode nécessite certaines adaptations pour des cas spéciaux.

Conclusion générale



Conclusion générale

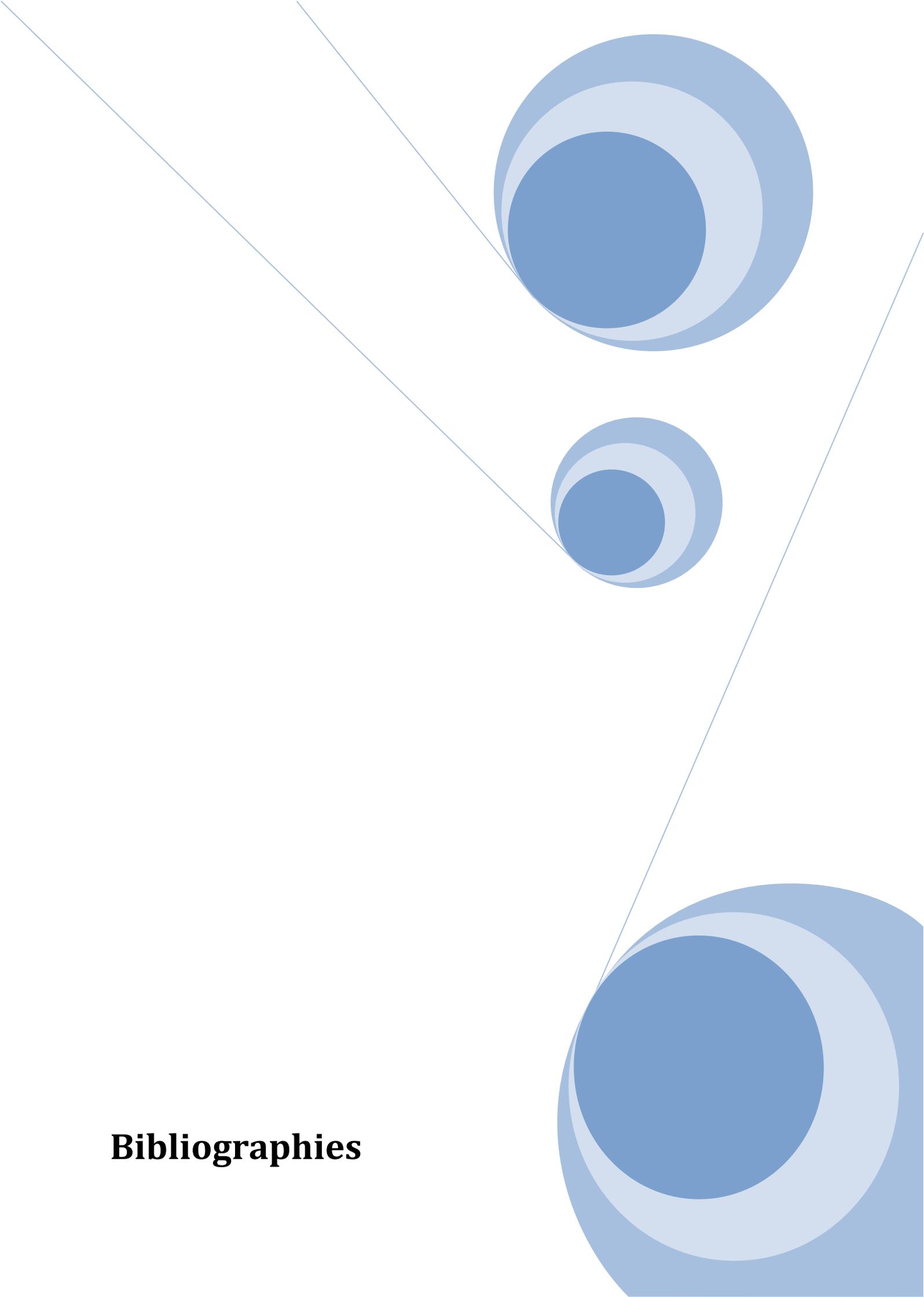
Dans ce travail, nous avons traité le problème de planification de trajectoire à temps optimal pour un robot SCARA. La méthode que nous avons utilisée donne un planificateur très rapide, ceci est dû à deux aspects :

D'un côté le problème de planification est découpé en deux sous problèmes, planification d'un chemin hors collision ensuite la planification de trajectoire. Le chemin planifié au départ prend en considération seulement les contraintes géométriques et il est paramétré en utilisant une seule variable $s \in [0,1]$.

De l'autre côté, la méthode utilise le fait que la commande pour la trajectoire à temps optimale doit être Bang-Bang. Ceci réduit les calculs en la recherche de point de commutation entre l'accélération minimale et maximale. Nous avons utilisé une procédure spéciale pour la recherche de ces points. Cette procédure commence par le calcul de la courbe de vitesse maximale (MVC) qui détermine la limite de la région dans laquelle les contraintes sur les couples articulaires sont vérifiées.

Dans les chapitres que nous avons traités on a pris pas mal d'information sur la robotique en générale et particulièrement sur les robots manipulateurs aussi comment utilisé la commande optimale pour la résolution de problème posé avec leur déférentes algorithmes finalisant par la résolution de problème on utilisant logiciel Matlab.

Les résultats de simulations montrent que la méthode est très efficace pour la planification de chemin à temps optimal qui est obtenu en quelques secondes. Néanmoins, le problème rencontré est celui de la robustesse de la méthode contre les cas de singularités dynamiques qui engendrent des points où $\dot{s} = 0$ ce qui impose la recherche de techniques pour le contourner.



Bibliographies

Bibliographie

- [1] S JLASSI, "Formulation et étude des problèmes de commande co-manipulation robotique", Thèse doctorat, Université Abderrahmane Mira de Bejaia, le 28/11/2013.
- [2] W Khalil et E Dombre, "Modélisation identification et commande des robots", revue et augmentée, 2^{ème} édition, janvier 1999.
- [3] M-L ZIANE, "Navigation flou d'un robot mobile", Mémoire master, Université Abderrahmane Mira de Bejaia, 2013.
- [4] YChikh et H-M Lounis, "Commande non linéaire d'un bras manipulateur", mémoire master, Université Abderrahmane Mira de Bejaia, 2016.
- [5] J-L Boimond, "Cours robotique", Cours, ISTIA, Université Angers.
- [6] F Makhloufi, "Modélisation et commande des robots manipulateurs par les outils de l'intelligence artificielle", Thèse doctorat, Université Badji-Mokhar-Annaba, Annaba 2015.
- [7] S Chabane, "planification et contrôle de trajectoire pour un bras de robot", mémoire master, Université Abderrahmane Mira de Bejaia, 2016.
- [8] M. Montigny, "Génération automatique de code pour la modélisation, simulation et commande de robot en temps réel", mémoire, L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES, mars 2000.
- [9] I Kairouan, "cours commande optimale", cours master, 2010.
- [10] B Bazem, "Planification de trajectoire a temps optimale d'un robot manipulateur à trois degrés de libertés", Mémoire master, Université Abderrahmane Mira Bejaia, 2012.
- [11] L Bouhalassa, "Planification de trajectoire d'un robot basé sur les réseaux de neurones et les algorithmes génétiques", Mémoire de magister, 2010.
- [12] Bobrow J. E. Dubowsky S. & Gibson J. S. "Time Optimal Control of Robotic Manipulators along Specified Path", Int. J. Robotic Reseach, Vol. 4. n° 3, 1985.
- [13] Howie Choset, Kevin Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia Kavraki, Sebastian Thrun, "Principles of Robot Motion, Theory, Algorithms, and Implementation" The MIT Press 2005.
- [14] J. J. Slotine, H. S. Yang, "Improving the Efficiency of Time Optimal Path Following Algorithms" IEEE Trans Robotics and Automation, vol. 5 n° 1, Feb 1989.
- [15] Quang-Cuong Pham, "A General, Fast and Robust Implementation of The Time-Optimal Path Parametrization Algorithms", IEEE Trans Robotics, vol. 30, n° 6, Dec. 2014.

- [16] Peiyao Shen, Xuebo Zhang and Yong chun Fang, ‘‘Essential Properties of Numerical Integration for Time-Optimal Trajectory Planning Along a Specified Path’’, Cs, RO, 1Dec 2016.
- [17] L. Zlajpah, ‘‘On Time Optimal Path Control of Manipulators with Bounded Joint Velocities and Torques’’ Proceeding of IEEE Conf. On Robotics and Automation, pp. 1572-1577, 1996.
- [18] P. T. Hung, Quang-Cuong Pham ‘‘On the Structure of the Time-Optimal Path Parametrization Problem with Third-Order Constraints’’ cs RO., Feb. 2017.
- [19] John J. Craig, ‘‘Introduction to Robotics, Mechanics and Control’’ 3rd edition, Prentice Hall 2005.
- [20] R. M. Murray, L. Zexiang, S. Sastry ‘‘Mathematical Introduction to Robotic Manipulation’’ CRC Press 1994.

Résumé

La robotique est l'ensemble des techniques permettant la conception et la réalisation de machine automatique ou des robots. Ce mémoire explique tout d'abord les différents concepts de base de la robotique en général et particulièrement les divers méthodes des calculs outil pour leur fonctionnement et modélisation développé par des chercheurs ensuite les approches utile pour la commande tel que la commande optimale et planification de trajectoire. Les robots c'est des appareils qui effectuant des taches précise grâce à un système de commande automatique, ce mémoire a pour objectif de mettre un programme on utilisant la méthode de commande optimale et le principe de maximum de pontryagin pour qu'un robot SCARA suivre un chemin prédéfini.

Mots clés: Planification de trajectoire, robotique, commande optimale, principe de maximum de pontryagin.

Abstract

Robotics is the set of techniques for the design and production of automatic machines or robots. This memory firstly explains the different basic concepts of robotics in general and especially the various methods of calculating tool for their operation and modeling developed by researchers then the useful approaches for the control such as optimal control and trajectory planning. Robots are devices that perform precise tasks thanks to an automatic control system; this memory aims to put a program using the optimal control method and the maximum principle of pontryagin for an SCARA robot to follow a predefined path.

Pass word: trajectory planning, Robotics, optimal control, maximum principle of pontryagin

