

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Abderahmane Mira
Faculté de Technologie – Département génie électrique

Mémoire de fin d'études

En vue d'obtention du Diplôme MASTER en électronique option Automatique

Thème :

**Planification de trajectoires optimales d'un robot à 2
ddl avec minimisation de surcharge et d'effort**

Réalisé par :

- Benhacine Fares
- Bouamara Sifaques

Encadré par :

Mr. Haddar Hocine

■ Année 2013/2014

Remerciements

Nous tenons à remercier Dieu de nous avoir donné la force morale, physique et l'aide pour accomplir ce modeste travail.

*Nous tenons à remercier Monsieur **Haddar** pour nous avoir acceptés encadrés et dirigés durant l'élaboration de ce travail ainsi que pour leur assistance et tous leurs conseils.*

*Nous remercions chaleureusement les membres du jury Monsieur **Mendil** et Monsieur **Guermouz** pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre projet.*

Nous souhaitons aussi remercier en particulier, Nos professeurs d'Automatique qui nous ont encadrés auparavant et tous nos enseignants pour les connaissances qu'ils nous ont transmis, leur disponibilité et leurs efforts.

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail trouvent ici l'expression de notre sincère gratitude.

Dédicaces

A ma Mère Saliha,

A mon Père Boualem,

A qui je dois ce que je suis,

*Qu'ils trouvent dans ce site, le fruit de leurs sacrifices consentis pour mon
éducation, et l'expression de mon amour et de ma gratitude pour la bienveillance
avec laquelle ils m'ont toujours entouré.*

Que Dieu leur préserve longue vie et bonne santé.

A mes frères, Mamou, Boubkeur, Hamza, Salim, Abdou,

A ma Sœur,

A toute ma famille,

A tous mes amis, Yacine, Nafaa, Nabil, Locif et Ahmed.

fARES

Dédicaces

Je dédie ce travail à ma très chère mère Zohra et à mon cher père Youcef qui ont été toujours à mes côtés dans les moments difficiles, qui m'ont tout appris, tant donné sans rien demandé en retour. Sans eux je ne serais pas l'homme que je suis et rien n'aurait été possible. Que Dieu vous gardent pour nous chers parents. Je dédie aussi ce travail à mon chère grand père et mes grandes mère. Ainsi à mes chères frères, Jegou, Salim, Salas, et mes chère sœurs. Qui m'ont soutenu et aidé tout au long de mon parcours. A mes cousins, Fahim, Walid, et a tous mes amis Yacine, Khaled, Ali, Nabil, et à ma chère fiancée Warda.

Sifaqes

LISTE DES NOTATIONS

m (chap I)	Nombre de degrés de liberté
R	Articulation rotoïde
P	Articulation prismatique
(P_x, P_y, P_z)	Coordonnée cartésiennes
p_t	Facteur d'échelle
${}^i T_j$	Matrice de transformation
$Trans(a, b, c)$	Matrice de translation pure
$Rot(x, \theta)$	Matrice de rotation pure
(x, y)	Variation opérationnelles
J	Matrice jacobienne
F	Force
M	Masse
w	Vitesse angulaire
c_I	Tenseur d'inertie
L	Lagrange
T	Energie cinétique totale
V	Energie potentielle totale
Γ	Couple
$(q \text{ ou } \theta)$	Position articulaire
$(\dot{q} \text{ ou } \dot{\theta})$	Vitesse articulaire
$(\ddot{q} \text{ ou } \ddot{\theta})$	Accélération articulaire
θ_0	Position initial
θ_f	Position final
t	Temps
P_j	Vecteur de points de contrôle
$B_{j,K}$	La fonction de base de B-spline
J_c	Le coût
∂J_c	Le gradient de coût
p_w	Poids
t_f	Temps final
r_c	Position du centre de masse
\dot{r}_c	Vitesse du centre de masse

LISTE DES ABRÉVIATIONS

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

Liste des Figures

Fig.I.01 : Représentation d'une articulation rotoïde.	03
Fig.I.02 : Représentation d'une articulation prismatique	03
Fig.I.03 : Représentation d'un point dans l'espace	07
Fig.I.04 : Changements de repères	08
Fig.I.05 : Translation pure	10
Fig.I.06 : Rotation autour de θ l'axe x	11
Fig.I.07 : Robot a structure ouverte simple	13
Fig.I.08 : Le passage du repère R_{j-1} au repère R_j	14
Fig.I.09 : Modèle géométrique direct et inverse	15
Fig.II.01 : La commande optimale	23
Fig.III.01 représentation d'un bras manipulateur à 2 ddl	43
Fig.III.02 : Génération de la trajectoire initiale par les points de contrôle	52
Fig.III.03 : Détermination des points de contrôle à partir du modèle	53
Fig.III.04 : Explication de la fonction F_{mincon}	55
Fig.III.05 : Les positions articulaires θ_1 et θ_2 pour la trajectoire initiale	59
Fig.III.04 : Les vitesses articulaires $\dot{\theta}_1$ et $\dot{\theta}_2$ pour la trajectoire initiale	60
Fig.III.07 : Les accélérations articulaires $\ddot{\theta}_1$ et $\ddot{\theta}_2$ pour la trajectoire initiale	60
Fig.III.08 : Les couples initiaux	61
Fig.III.09 : Génération de la trajectoire initiale avec deux méthodes	62

Table des matières

Liste de notations

Liste d'abréviations

Liste de figures

Introduction Générale

Chapitre I Généralités sur la robotique

<i>I.1. Introduction</i>	1
<i>I.2. Définitions</i>	1
I.2.1. Le Robot	1
I.2.2. Les articulation	1
I.2.3. Espace articulaire	2
I.2.4. Espace opérationnel	2
I.2.5. Base, structure et organe terminal	2
I.2.6. Redondance d'un robot	3
I.2.7. Singularité d'un robot	3
I.2.8. La commande d'un robot	3
I.2.9. Composition d'un robot	3
1.2 .10. Architecture des robots	4
I.2.11. Caractérisation des performances des robots	4
I.2.11.1. Versatilité du robot	5
I.2.11.2. Auto adaptabilité à l'environnement	5
I.2.11.3. Critère relatif à la sécurité	5
I.2.11.4. Exactitude de trajectoire	5
I.2.11.5. Répétitivité de trajectoire	5
I.2.11.6. Reproductibilité de pose	6
I.2.11.7. Critère d'ordre économique	6
<i>I.3. Matrices de transformations homogènes</i>	6
I.3.1. Coordonnées homogènes	6
I.3.2. Transformation homogène	7
I.3.2.1. Changement de repère	7
I.3.2.2. Transformation de vecteurs	8
I.3.2.3 Matrice de translation pure homogène	8
I.3.2.4. Matrice de rotation homogène	9
<i>I.4. Modélisation</i>	10

I.4.1. MODELISATION GEOMETRIQUE	10
I.4.1.1. Modèle géométrique direct	10
I.4.1.2. Convention de Denavit –Hartenberg.....	11
I.4.1.3. Les paramètres de Denavit-Hartenberg.....	12
I.4.1.4. Modèle géométrique inverse.....	13
I.4.2. MODELISATION CINEMATIQUE	14
I.4.2.1. Modèle cinématique direct	14
I.4.2.2. La matrice jacobienne	14
I.4.2.3. L'intérêt de la matrice jacobienne.....	15
I.4.2.4. Le modèle cinématique inverse	15
1.4.3. Modélisation dynamique	15
1.4.3.1. Formalisme de Lagrange	16
<i>I.5. Génération de la trajectoire dans l'espace articulaire</i>	16
I.5.1 La fonction polynomiale cubique.....	17
I.5.3. mouvement continu entre points via	18
<i>I.6. Conclusion</i>	19
Chapitre II Commande optimale et optimisation	
<i>II.1 - Introduction.....</i>	20
<i>II .2 - La commande optimale</i>	20
<i>II.3- Problème de la commande optimale</i>	21
<i>II.4- Programmation dynamique</i>	22
<i>II.5 Conditions Nécessaires d'Optimalité</i>	23
II.5.1- Equations d'Euler-Lagrange.....	23
II.5.2 - Principe Du Minimum De Pontryagin.....	25
<i>II.6- Méthodes Indirectes</i>	26
II.6.1- Méthodes En Boucle Fermée.....	26
II.6.2- Méthodes De Commande En Boucle Ouverte	27
<i>II.7 - Méthodes Directes</i>	27
II.7.1 Méthodes De Discrétisation	27
II.8.2 - Méthodes de Paramétrisation	28
<i>II.8 Méthodes générales d'optimisation</i>	30
II.8.1 Définition de l'optimisation.....	31
II.8.2 Optimisation locale	31
II.8.2.1 Méthode de descente.....	31
II.8.2.2 Algorithme génétique.....	32
II.8.3 Optimisation globale.....	32

II.8.3.1 Méthode des variations locales.....	32
II.8.3.2 Algorithme par séparation et Evaluation	32
II.8.3.3 Branch and prune	33
II.8.4 Optimisation stochastique.....	33
II.8.4.1 Méthode adaptative.....	34
II.8.5 Importance du gradient	34
<i>II.9 Interpolation et approximation de courbes</i>	<i>36</i>
II.9.1 Hystorique	36
II.9.2- Principe de l'interpolation	37
II.9.3 Interpolation quadratique.....	37
II.9.3.1 Inconvénients.....	37
II.9.4 L'interpolation cubique	37
II.9.4.1 Avantages et inconvénients.....	37
II.9.5. L'interpolation de lagrange	38
II.9.5.1 Avantages et inconvénients.....	38
II.9.6. Interpolation par splines cubiques	38
II.9.6.1 Avantages et inconvénients.....	38
II.9.7. Les B-Splines (principe)	38
II.9.7.1 Fonctions de Base	38
<i>II.10 Description de la fonction Fmincon de MATLAB</i>	<i>39</i>
<i>II.11 Conclusion</i>	<i>40</i>
Chapitre III Trajectoire optimale d'un robot à deux ddl	
<i>III.1 Introduction</i>	<i>42</i>
<i>III.3 Génération de la trajectoire par B-spline</i>	<i>47</i>
<i>III.4 Planification de trajectoire optimale avec chemin contraint et effort minimal</i>	<i>49</i>
III.5. La Trajectoire initiale	52
III.5.1 Détermination des points de contrôle :	52
III.5.2 Optimisation de la trajectoire :.....	53
III.5.2.1 Approximation de la trajectoire par B-splines :.....	53
III.5.2.2 Optimisation.....	54
<i>III.6 Résultats de simulation</i>	<i>59</i>
<i>Conclusion :</i>	<i>65</i>
Conclusion Générale	
Références Bibliographiques	

Introduction Générale :

La robotique est un domaine connu par sa multitude de branches et d'applications dont on cite l'automatique, l'électronique, les mathématiques et la mécanique, la médecine aussi a bénéficié de la robotique, le domaine militaire ainsi que le domaine spatial ne peuvent plus aujourd'hui s'en passer du robot.

La planification des trajectoires optimales est l'objet de plusieurs recherches depuis les années 80. Certaines d'entre elles affirment que le chemin de l'organe terminal est prédéfini, elles cheminent vers une solution optimale en prenant en considération des limites constantes sur les forces des actionneurs. D'autres travaux prennent en considération des limitations constantes sur les couples des actionneurs et sur la dérivée de l'accélération (Jerk). Dans sa forme la plus récente, l'optimisation est le procédé par lequel est générée une trajectoire à temps minimal en prenant en considération les contraintes introduites par la dynamique du robot et les limitations sur les vitesses et les couples au niveau des articulation qui sont dues aux actionneurs. L'optimisation prend également en charge l'évitement d'obstacles. La trajectoire est représentée dans un espace de dimension fini. Cet espace peut être couvert par un ensemble fini d'éléments qui représentent sa base. [10]

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

CHAPITRE I : GÉNÉRALITÉS SUR LA ROBOTIQUE

I.1.Introduction

Pour concevoir, simuler ou commander un robot, il est nécessaire, entre autres, de disposer de modèles du mécanisme. Plusieurs niveaux de modélisation sont possibles. Ils dépendent des spécifications du cahier des charges de l'application envisagée : il en découle des modèles géométriques, cinématiques et dynamiques à partir desquels peuvent être engendrés les mouvements du robot, ou bien des modèles statiques qui décrivent les interactions du mécanisme avec son environnement.

L'obtention de ces différents modèles n'est pas aisée, la difficulté variant selon la complexité de la cinématique de la chaîne articulée. Entrent en ligne de compte le nombre de degrés de liberté, le type des articulations mais aussi le fait que la chaîne peut être ouverte simple, arborescente ou fermée.

I.2. Définitions

I.2.1. Le Robot

C'est un système mécanique poly-articulé mû par des actionneurs et Commandé par un ordinateur qui est destiné à effectuer une grande variété de tâches .L'Association Française de Normalisation (A.F.N.O.R.) définit un robot comme étant un système mécanique de type manipulateur commandé en position, reprogrammable, polyvalent (i.e., à usages multiples), à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un, ou plusieurs, bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel [1].

I.2.2. Les articulation

- La liaison au sens mécanique lie deux corps successifs en autorisant un certain nombre de degré de liberté, en robotique on utilise la notion d'articulation pour décrire le mouvement relatif entre deux corps, ces articulations qui ne possèdent qu'un seul d.d.l (rotation ou translation) sont communément appelées :
- la rotoïdes qui représente la rotation (pivot).
- la prismatique qui représente la translation (glissière).

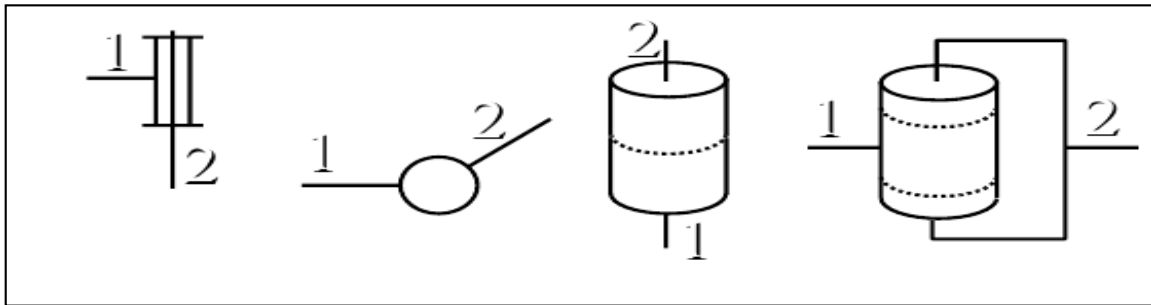


Fig.I.01 : Représentation d'une articulation rotoïde

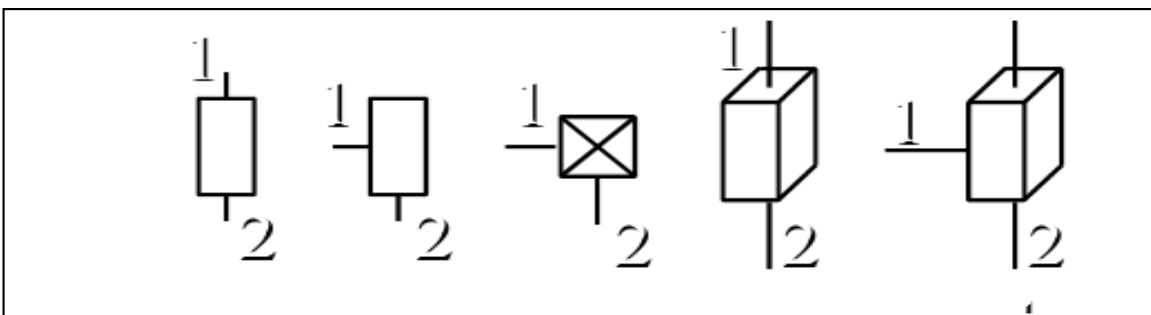


Fig.I.02 : Représentation d'une articulation prismatique

I.2.3. Espace articulaire

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

I.2.4. Espace opérationnel

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

I.2.5. Base, structure et organe terminal

On peut schématiser un robot à partir :

- ✓ d'une base fixe.
- ✓ d'un organe terminal (extrémité mobile du robot appelé aussi effecteur).
- ✓ d'une structure mécanique articulée qui amène l'organe terminal dans une situation (position et orientation) donnée.

I.2.6. Redondance d'un robot

Un robot est dit redondant lorsque le nombre d'articulations motorisées (n) est supérieur au nombre de paramètres nécessaires pour d'écrire une situation dans l'espace (m). Cette propriété permet essentiellement d'augmenter le volume atteignable et de contourner un obstacle lors du déplacement de l'organe terminal.

Pour la chaîne ouverte simple, les combinaisons suivantes donnent une structure redondante :

- ✓ +6 articulations motorisées
- ✓ + de 3 rotoïdes d'axes concourants
- ✓ +de 3 articulations rotoïdes d'axes parallèles
- ✓ +de 3 articulations prismatiques
- ✓ 2 axes d'articulations prismatiques parallèles
- ✓ 2 axes d'articulations rotoïdes confondus

I.2.7. Singularité d'un robot

Pour tous les robots qu'ils soient redondants ou non, il se peut que dans certaines configurations dite singulière le nombre de d.d.l du robot soit inférieur à la dimension de l'espace opérationnel, ce cas se présente lorsque par exemple :

- ✓ 2 axes d'articulations prismatiques se retrouvent parallèles
- ✓ 2 axes d'articulations rotoïdes se retrouve confondus

I.2.8. La commande d'un robot

La commande d'un robot exige un peu plus que la résolution d'équation géométrique, en effet pour générer un mouvement et atteindre une situation donnée, il faut intégrer dans la boucle de commande les paramètres suivant :

- ✓ les valeurs de tension de commande de différents actionneurs
- ✓ les couples ou forces en sortie d'actionneurs
- ✓ les valeurs des couples s'exerçant sur les articulations
- ✓ les variations articulaires
- ✓ les variations opérationnelles (tâches à réaliser)

Commander un robot c'est d'être capable de décrire sous forme d'équations les comportements liés aux composantes citées précédemment et d'établir les liens entre :

- ✓ la valeur des tensions et les variations opérationnelles : modèle d'exécution de la tâche (modèle direct)
- ✓ les variations opérationnelles et les tensions de commande : modèle de commande (modèle indirect)

I.2.9. Composition d'un robot

a. Le mécanisme

Structure plus au moins proche de celle du bras humain, on dit aussi manipulateur quand il ne s'agit pas d'un robot mobile.

Sa motorisation est réalisée par de actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leur mouvement aux articulations par des systèmes appropriés.

b. La perception

Permet de gérer les relations entre le robot et son environnement.

Les organes de perception sont des capteurs dits « proprioceptifs » lorsqu'ils mesurent l'état interne du robot (position et vitesses des articulations) ou « extéroceptifs » lorsqu'ils recueillent des informations sur l'environnement (détection de présence, mesure de distance, vision artificielle).

c. La commande

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

d. L'interface homme-machine

À travers laquelle l'utilisateur programme les tâches que le robot doit exécuter.

e. Le poste de travail et les dispositifs perirobotique

Ils constituent l'environnement dans lequel évolue le robot.

1.2.10. Architecture des robots

Un robot comporte 2 parties essentielles :

a. Le porteur

Structure mécanique articulée constituée des 3 premiers degrés de liberté à partir du bâti. Si P est un point de l'extrémité et R0 un repère lié au bâti, le rôle du porteur est de fixer la position de P dans R0.

Les liaisons utilisées sont des liaisons pivot notées R ou prismatique notées P.

b. Le poignet

Il est destiné à l'orientation de la pince ou de l'outil porté par le robot.

1.2.11. Caractérisation des performances des robots

Contrairement aux autres machines, les robots présentent une grande flexibilité et une grande souplesse dans leurs utilisations. Ils peuvent même s'adapter aux modifications de l'environnement car ils intègrent différentes technologies.

Le robot est une machine complexe qui intègre de multiples technologies : mécanique, asservissement, électromécanique, électronique, informatique [11]. Les interactions entre ces

différents éléments et leurs contributions aux performances du robot sont difficiles à établir. Actuellement, bien qu'étant conscients du rôle essentiel de l'armoire de commande, nous ne la dissociions pas de la structure mécanique du robot qui est une partie très importante dans la conception de ce dernier. Par conséquent le robot est évalué comme un ensemble complet.

I.2.11.1. Versatilité du robot

Les différents types de structures mécaniques du robot permettent l'exécution des tâches diverses ou parfois les même tâches de différentes façons, d'une manière générale, les performances dépendent : de la posture du robot, de la position du robot au sein de l'espace de travail, de la charge, de l'inertie de l'objet manipulé, et de la vitesse avec laquelle la tâche sera exécutée [9].

I.2.11.2. Auto adaptabilité à l'environnement

Elle caractérise la capacité d'initiative du robot pour mener à bien des tâches incomplètement spécifiées, et ceci malgré des modifications imprévues de l'environnement [7].

I.2.11.3. Critère relatif à la sécurité

Un système mécanique en mouvement présente, en cas de choc un danger d'autant plus grand que son énergie cinétique est élevé [8]. En conséquence, le concepteur devra (tout en respectant les contraintes technologiques visant à assurer à la structure du robot une solidité suffisante) minimiser l'inertie du bras manipulateur en utilisant des matériaux appropriés et en allégeant les différents éléments constituant le manipulateur. De plus il devra définir une vitesse maximale du déplacement de l'organe terminal afin de limiter l'énergie cinétique du manipulateur.

I.2.11.4. Exactitude de trajectoire

Elle caractérise l'aptitude d'un robot à faire suivre à l'interface mécanique une trajectoire commandée « n » fois dans la même direction, et « n » fois dans la direction opposée [1].

I.2.11.5. Répétitivité de trajectoire

C'est l'étroitesse de l'accord entre les trajectoires atteintes pour la même trajectoire commandée « n » fois [1].

I.2.11.6. Reproductibilité de pose

Elle permet de quantifier les écarts entre les atteintes avant et après un arrêt du robot, cet arrêt du robot peut être un système, un arrêt d'urgence ou tout simplement l'arrêt en fin de journée et la reprise le lendemain matin. La valeur de cette caractéristique permettra de se rendre compte si le robot est capable d'accomplir une tâche sans programmation de certains points [35].

I.2.11.7. Critère d'ordre économique

Le prix de robot étant dû en grande partie à la configuration de sa structure (élément rigide, lourds et encombrant), ainsi qu'au soin apporté à la réalisation de celle-ci (qualité précision d'usinage), la solution d'avenir est probablement de remplacer les structures actuelles (rigides, précises) par des structures plus légères, semi-rigide dont la construction plus simple et moins précise entraînerait des coûts de fabrication réduits [6].

I.3. Matrices de transformations homogènes [3][2][1]

I.3.1. Coordonnées homogènes

- Un point est représenté par P_x , P_y et P_z coordonnées cartésiennes

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

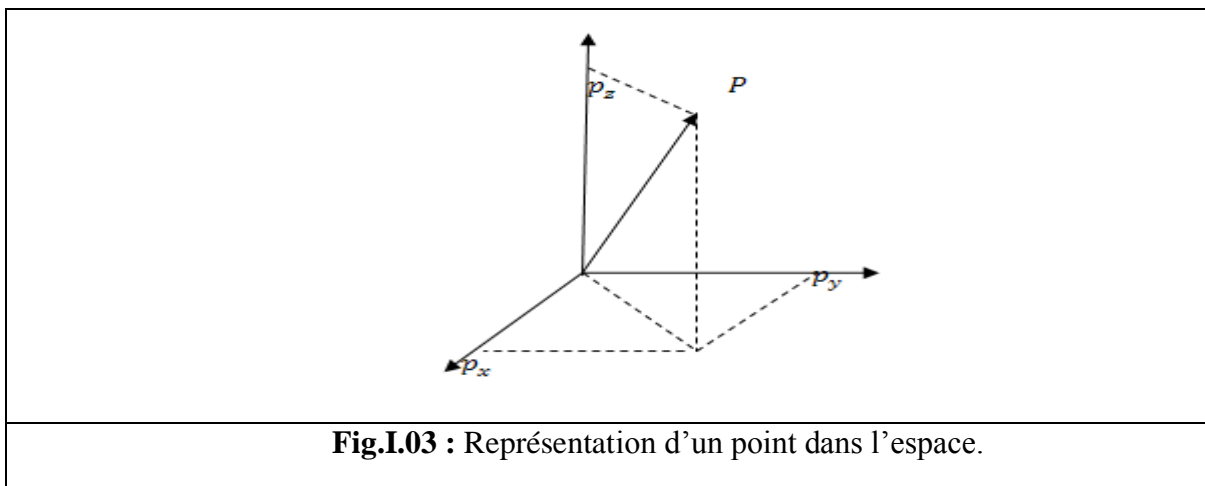


Fig.I.03 : Représentation d'un point dans l'espace.

- Représentation d'une direction (vecteurs libre)

$$u = \begin{bmatrix} u_x \\ u_y \\ u_z \\ 0 \end{bmatrix}$$

- Représentation d'un plan

On représente le plan par l'équation $\alpha x + \beta y + \gamma z + \delta = 0$ par la vectrice ligne q :

$$q = \begin{bmatrix} \alpha x \\ \beta y \\ \gamma z \\ \delta \end{bmatrix}$$

Pour tout p situé dans le plan q , le produit matriciel $q p$ Est nul.

I.3.2. Transformation homogène

I.3.2.1. Changement de repère

On définit la matrice de transformation homogène par :

$$T_j^i = [s_j^i \ n_j^i \ a_j^i \ P_{j,i}^i] = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

s_j^i, n_j^i et a_j^i sont les vecteurs unitaires des axes X_j, Y_j et Z_j du repère R_j exprimé dans R_i .

P_j vecteur exprimant l'origine du repère R_j dans le repère R_i

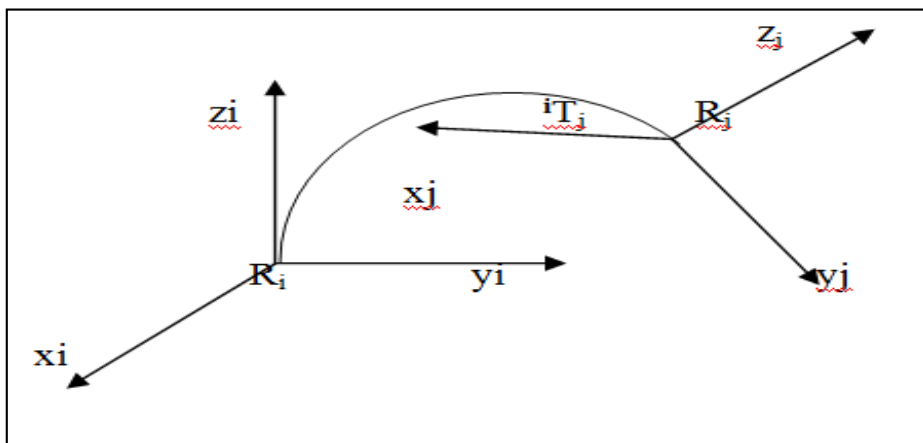


Fig.I.04 : Changements de repères

On écrit aussi :

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

La matrice A représente la matrice de rotation ou d'orientation du repère R_i par rapport à R_j .

La colonne P représente la translation du repère R_i par rapport au repère R_j .

Dans le cas d'une translation pure $A = I_3$, tel que I est la matrice unité.

Propriétés

La matrice A est orthogonale : $A^{-1} = A^T$

$$T_j^{i-1} = T_i^j$$

$$Rot(u, \vartheta)^{-1} = Rot(u, -\vartheta) = Rot(-u, \vartheta)$$

$$Trans(u, d)^{-1} = Trans(-u, d) = Trans(u, -d)$$

I.3.2.2. Transformation de vecteurs

Soit un vecteur P_j^i définissant le point P1 dans le repère R_j

On calcule les coordonnées homogènes du point P1 dans le repère R_i par l'équation suivante :

$$P_i^j = (O_i P_1)^i = s_j^i P_{1x}^j + n_j^i P_{1y}^j + a_j^i P_{1z}^j = T_i^j P_1^j$$

La matrice T_i^j permet donc d'exprimer dans le repère R_i les coordonnées d'un point dans le repère R_j

I.3.2.3 Matrice de translation pure homogène

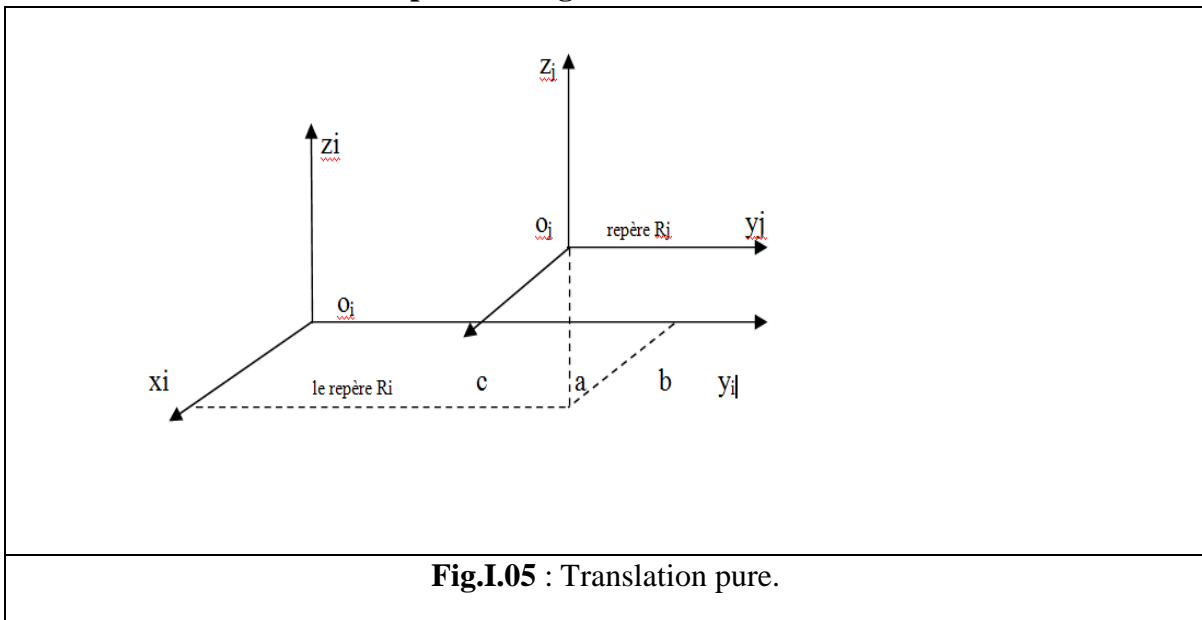


Fig.I.05 : Translation pure.

Soit $Trans(a, b, c)$ une transformation qui désigne la translation a , b , et c le long des axes x , y , et z respectivement.

La transformation dans ce cas s'exprime par :

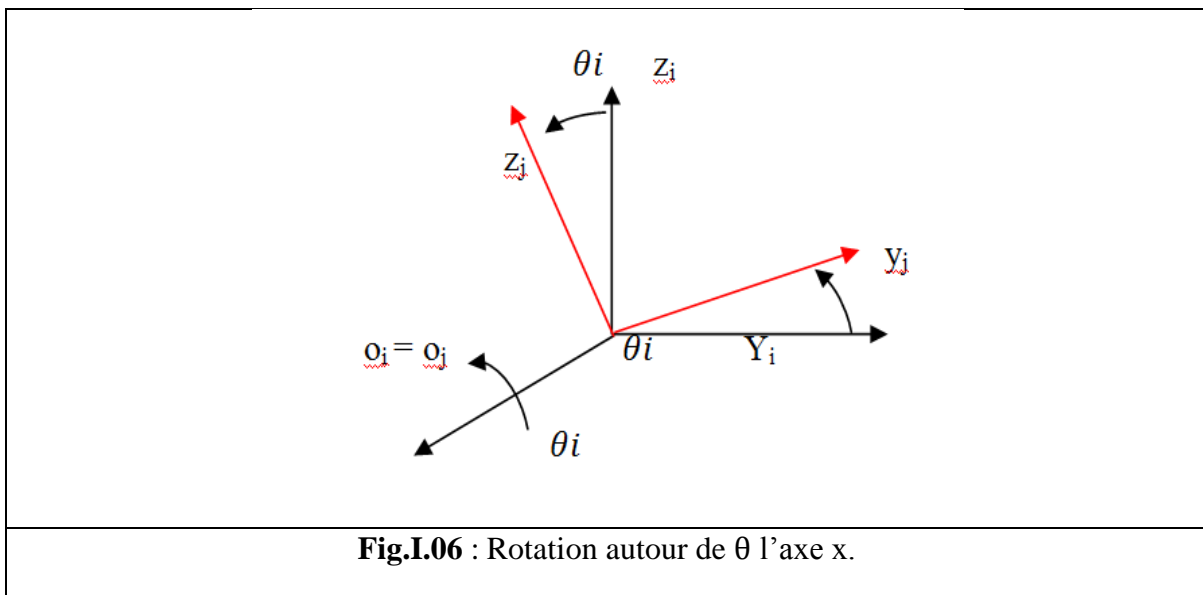
$$A = Trans(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

On utilise par la suite la notation $Trans(u, d)$ pour désigner une translation d'une valeur d le long de l'axe u .

Propriétés : $Trans(a, b, c) = Trans(x, a)Trans(y, b), Trans(z, c)$

L'ordre des multiplications étant quelconque.

I.3.2.4. Matrice de rotation homogène



On définit $Rot(x, \theta_1)$ la transformation homogène qui s'exprime par :

$$Rot(x, \theta_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_1 & -\sin \theta_1 & 0 \\ 0 & \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$Rot(x, \theta_1)$ désigne la rotation ou l'orientation de repère R_i d'un angle θ_1 autour de l'axe x du repère R_j

De la même façon on définit la rotation d'un angle θ_2 autour de y par :

$$Rot(y, \theta_2) = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Et la rotation d'un angle θ_3 autour de z par :

$$Rot(z, \theta_3) = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

I.4. Modélisation [1][2]

En robotique, on associe à tout élément du poste de travail un ou plusieurs repères. Ces repères sont généralement définis de telle sorte que leurs origines correspondent à des directions et à des points privilégiés ayant un rôle fonctionnel lors de l'exécution d'une tâche, telle que la direction d'insertion, le centre de gravité, l'orientation de l'articulation ou extrémité de l'outil [5].

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

- D'exprimer la situation des différents corps du robot les uns par rapport aux autres ;
- De spécifier les situations que doit prendre le repère associé à l'organe terminal du robot pour réaliser une tâche donnée ainsi que les vitesses correspondantes.

Nous présentons dans ce chapitre une notation qui permet de décrire de façons homogène les différents systèmes de coordonnées.

I.4.1. MODELISATION GEOMETRIQUE

I.4.1.1. Modèle géométrique direct

Le Modèle géométrique direct consiste à exprimer la situation de l'organe terminal en fonction des variables articulaires du robot. On doit trouver une façon de définir la position de l'objet de manière à ce que le robot sache où elle se trouve par rapport à un système de référence connu. La façon de faire est de fixer un système de coordonnées sur l'objet ainsi que sur l'outil et d'essayer de trouver les transformations nécessaires que doit subir le système de coordonnées de l'outil pour coïncider avec celui de l'objet.[4]

L'étude de la géométrie directe des robots manipulateurs comprend plusieurs étapes pour arriver aux objectifs visés par cette modélisation. Dans un premier temps on fixe les repères aux différentes parties du mécanisme et on décrit les relations entre ces repères, et aussi on localise ces repères lorsque le manipulateur s'articule. Le modèle géométrique direct (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminale, c'est-à-dire les coordonnées opérationnelles du robot, en fonction de ses coordonnées articulaire dans le cas d'une chaîne ouvert simple, Il peut être représenté par la matrice de passage 0T_n .

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) \dots \dots \dots {}^{n-1}T_n(q_n)$$

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

$$X = f(q) \quad (I.01)$$

Avec : X les coordonnées cartésiennes,

Et q les coordonnées articulaires.

On va admettre quelques hypothèses dans le but de simplifier la modélisation des robots. Ces hypothèses sont les suivantes :

- Les liaisons du manipulateur sont rigides.
- Les jeux dans les articulations sont négligeables.
- Les capteurs ont un gain unitaire et de dynamique négligeable.

I.4.1.2. Convention de Denavit –Hartenberg

Méthodologie à suivre pour décrire les robots à structure ouverte simples.

Une structure ouverte simple est composée de $n+1$ corps notés C_0, \dots, C_n et de n articulations.

Le corps C_0 désigne la base du robot et le corps C_n le corps qui porte l'organe terminal.

L'articulation j connecte le corps C_j au corps C_{j+1} (figI.07)

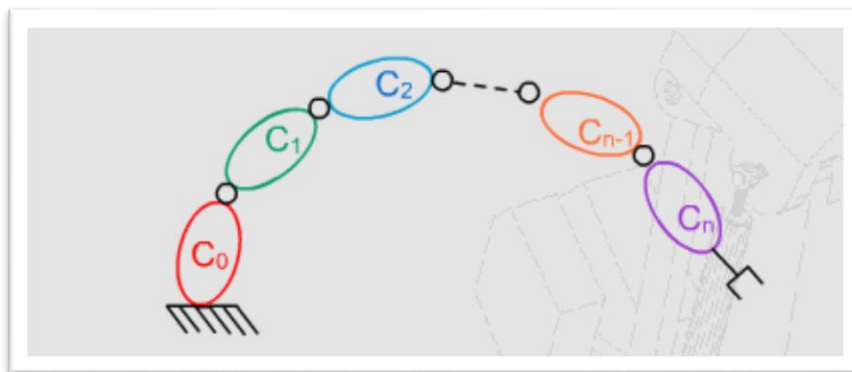


Fig.I.07 : Robot à structure ouverte simple.

La méthode de description est basée sur le principe suivant :

Principe

- Fixer un repère à chaque corps du robot.
- Calculer les matrices homogènes entre chaque deux segments successifs.
- Calculer la matrice homogène entre base et organe terminal.

Hypothèses

On suppose que le robot est constitué d'un chaînage de $n+1$ corps liés entre eux par n articulations rotoïdes ou prismatiques. A chaque corps, on associe un repère R_i .

Les repères sont numérotés de 0 à n . La i ème articulation, dont la position est notée q_i , est le point qui relie les corps $i-1$ et i .

Le repère R_j fixé au corps C_j , est défini de sorte que :

- L'axe z_j est porté par l'axe de l'articulation j .
- L'axe x_j est porté par la perpendiculaire commune aux axes z_j et z_{j+1} . Si les axes z_j et z_{j+1} sont parallèles ou colinéaires, le choix de x_j n'est pas unique.

I.4.1.3. Les paramètres de Denavit-Hartenberg

Le passage du repère R_{j-1} au repère R_j s'exprime en fonction des quatre paramètres géométriques suivants : (Fig.07)

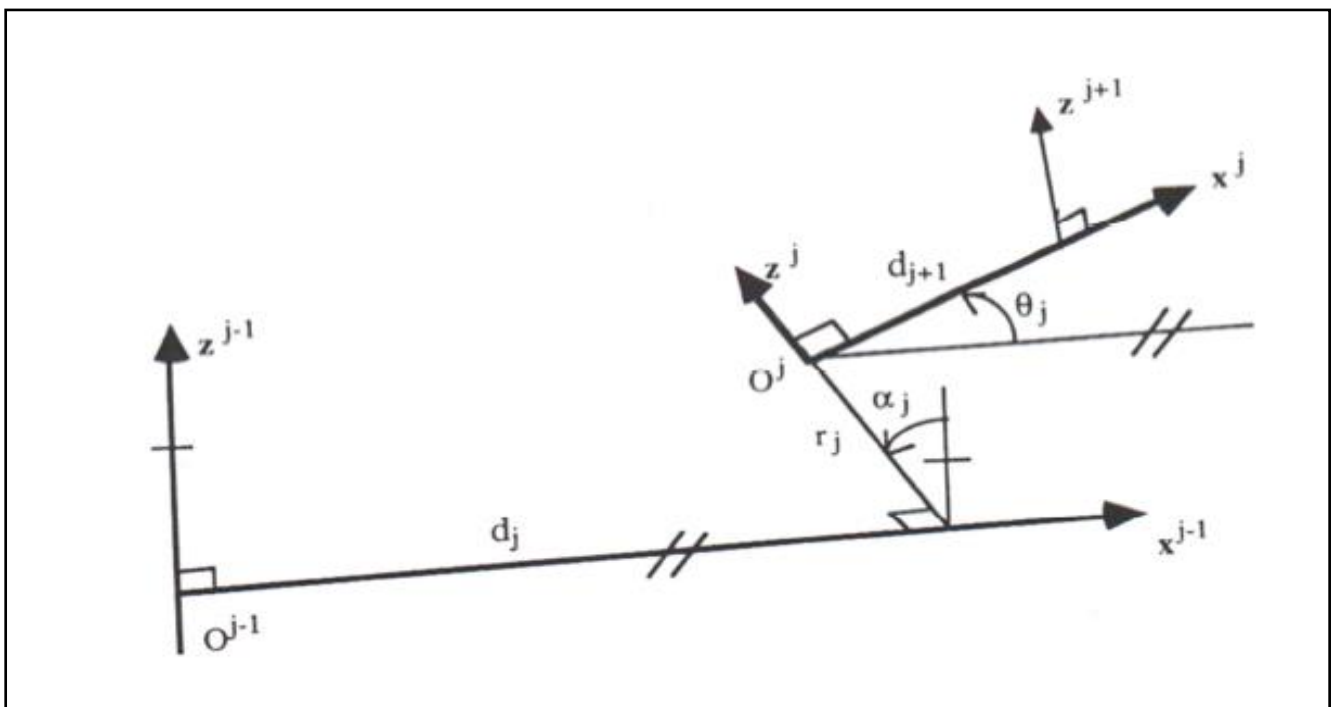


Fig.I.08 : Le passage du repère R_{j-1} au repère R_j

1/ α_j : angle entre les axes z_{j-1} et z_j correspondant à une rotation autour de x_{j-1}

2/ d_j : distance entre z_{j-1} et z_j le long de x_{j-1}

3/ θ_j : angle entre l'axe x_{j-1} et x_j correspondant à une rotation autour de z_j .

4/ r_j : distance entre x_{j-1} et x_j le long de z_j .

- Si l'articulation i est de type prismatique, alors d_i est variable
- Si l'articulation i est de type rotoïde, alors θ_i est variable.

La variable articulaire q_j associée à la j ieme articulation est définie par :

$$q_j = \delta_j \theta_j + \delta_j r_j \quad (I.02)$$

Avec :

$\delta_j = 0$ si l'articulation j est rotoïde.

$\delta_j = 1$ si l'articulation est prismatique

A partir de cette description on peut définir la matrice de transformation homogène définissant le repère R_j dans le repère R_{j-1} .

On a :

$${}^{j-1}T_j = Rot(x, \alpha_j) \times Trans(x, d_j) \times Rot(z, \theta_j) \times Trans(z, r_j) \quad (I.03)$$

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

Où $Rot(u, \alpha)$ et $Trans(u, d)$ sont des matrices homogènes (4x4) représentant.

I.4.1.4. Modèle géométrique inverse

Le problème dans cette section consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Ce qui est montré par la figure :

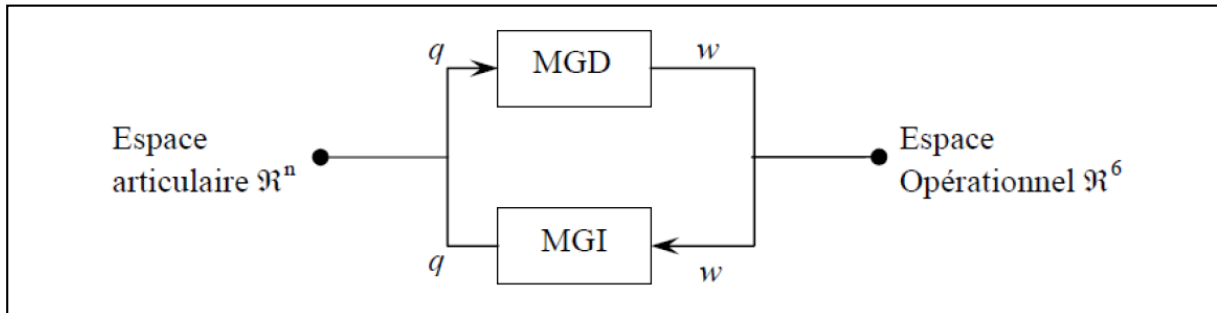


Fig.I.09 : Modèle géométrique direct et inverse.

Le problème posé est le suivant : étant donné la position et l'orientation de l'outil par rapport à la station de travail, comment calculer l'ensemble des angles articulaires qui accomplissent cet objectif ? La réponse à cette question constitue le modèle géométrique inverse du robot manipulateur.

La solution du problème, concernant la recherche des angles articulaires nécessaires pour positionner le repère de l'outil, par rapport au repère de la station de travail, est décomposée en deux parties. En premier lieu, sont déterminées les transformations nécessaires pour trouver le repère du poignet, par rapport au repère de la base et après, le modèle géométrique inverse est utilisé pour trouver les angles des articulations. Pour cela nous nous intéressons à une seule méthode celle de Paul qui traite séparément chaque cas particulier et convient à la plupart des robots industriels [1].

I.4.2. MODELISATION CINEMATIQUE

I.4.2.1. Modèle cinématique direct

Dans ce qui suit, nous abordons le problème du calcul du modèle cinématique direct, ce modèle décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires. Il est noté :

$$\dot{x}=J(q)\dot{q} \quad (I.04)$$

Où $J(q)$ désigne la matrice jacobienne de dimension $(m \times n)$ du mécanisme, égale à $\frac{\delta X}{\delta q}$ et en fonction de la configuration articulaire q .

La même matrice jacobienne intervient dans le calcul du modèle différentiel directe qui donne les variations élémentaires dX des coordonnées opérationnelles en fonction des variations dq , soit : $dX= J(q)$

Dans ce qui suit nous allons expliquer la méthode qui nous permet de déterminer la jacobienne de base d'un mécanisme à chaîne ouverte simple.

On peut obtenir la matrice jacobienne par la méthode de calcul directe, fondée sur la relation entre les vecteurs des vitesses de translation et de rotation V_n et w_n du repère R_n , représentant les éléments de réduction du torseur cinématique du repère R_n , et les vitesses articulaires \dot{q} :

$$V_n = \begin{bmatrix} V_n \\ w_n \end{bmatrix} = J_n \dot{q}$$

Notons que V_n est la dérivée par rapport au temps du vecteur P_n . En revanche w_n , ne représente pas la dérivée d'une représentation quelconque de l'orientation.

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

$$J_{ij} = \frac{\delta f_i(a)}{\delta q_j} \quad \text{Avec } i = 1, \dots, n, j = 1, \dots, n$$

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

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

I.4.2.2. La matrice jacobienne

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

I.4.2.3. L'intérêt de la matrice jacobienne

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

I.4.2.4. Le modèle cinématique inverse

L'objectif du modèle cinématique inverse (MCI) est de calculer, à partir d'une configuration q donnée, les vitesses articulaires \dot{q} qui assurent au repère terminal une vitesse opérationnelle \dot{X} imposée. On peut déterminer la différentielle articulaire dq correspondante à une différentielle des coordonnées opérationnelles spécifiées dX . Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires. La mise en œuvre peut être faite de façon analytique ou numérique :

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

$$\dot{q} = J^{-1} \dot{X} \quad (\text{I.05})$$

1.4.3. Modélisation dynamique

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

$$\Gamma = f(q, \dot{q}, \ddot{q}, f_e) \quad (\text{I.06})$$

Avec :

Γ : Vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique ;

q : Vecteur des positions articulaires ;

\dot{q} : Vecteur des vitesses articulaires ;

\ddot{q} : Vecteur des accélérations articulaires ;

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

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

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

$$\ddot{q} = g(q, \dot{q}, \Gamma, f_e) \quad (\text{I.07})$$

Parmi les applications du modèle dynamique, on peut citer :

- ✓ la simulation, qui utilise le modèle dynamique direct ;
- ✓ le dimensionnement des actionneurs ;
- ✓ l'identification des paramètres inertiels et des paramètres de frottement du robot ;
- ✓ la commande, qui utilise le modèle dynamique inverse.

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

- le formalisme de Lagrange
- le formalisme de Newton-Euler

On présente dans ce paragraphe ces deux formalismes pour les robots à chaîne ouverte simple (pour les robots à chaîne complexe, on y aborde également le problème de la détermination des paramètres inertiels minimaux.

1.4.3.1. Formalisme de Lagrange [3], [2]

Le but de cette partie est d'étudier la forme générale du modèle dynamique, de mettre en évidence les différents termes qui y interviennent et de déduire leurs propriétés caractéristiques. La méthode présentée n'est pas celle qui donne le modèle le plus performant du point de vue du nombre d'opérations, mais c'est la méthode la plus simple compte tenu de ces objectifs. Nous considérerons un robot idéal sans frottement, sans élasticité et ne subissant ou n'exerçant aucun effort extérieur.

Le formalisme de Lagrange décrit les équations du mouvement, lorsque l'effort extérieur sur l'organe terminal est supposé nul, par l'équation suivante :

$$\Gamma = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \quad (\text{I.08})$$

Avec :

- L : lagrangien du système égal à E-U ;
- E : énergie cinétique totale du système ;
- U : énergie potentielle totale du système.

I.5. Génération de la trajectoire dans l'espace articulaire [4]

La génération de la trajectoire dans l'espace articulaire donne en résultat un ensemble de données : position, vitesse et accélération articulaires $(\theta, \dot{\theta}, \ddot{\theta})$ qui sont utilisées comme un signal de référence .Pour une position initiale et finale données dans l'espace de travail on utilise la géométrie inverse pour déterminer les angles articulaires correspondant à cette

position et même pour l'orientation, la position initiale du manipulateur devient un ensemble d'angles articulaires de départ et la position finale correspond à un autre ensemble d'angles articulaires d'arrivée, ce qui est demandé pour la planification de la trajectoire est de trouver une fonction lisse pour chaque articulation dont la valeur à l'instant t_0 est la position initiale de l'articulation et dont la valeur à l'instant t_f est la position désirée de cette même articulation. Il y a beaucoup de fonctions lisses $\theta(t)$ qui pourraient être utilisées pour interpoler les valeurs d'angles articulaires. On cite :

I.5.1 La fonction polynomiale cubique [35], [37]

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

$$\theta(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 \quad (\text{I.09})$$

Ainsi on obtient la vitesse articulaire :

$$\dot{\theta}(t) = a_1 + 2 \cdot a_2 \cdot t + 3 \cdot a_3 \cdot t^2 \quad (\text{I.10})$$

Et aussi l'accélération articulaire :

$$\ddot{\theta}(t) = 2 \cdot a_2 + 6 \cdot a_3 \cdot t \quad (\text{I.11})$$

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

$$\theta(0) = \theta_0, \theta(t_f) = \theta_f \quad (\text{I.12})$$

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

$$\dot{\theta}(0) = 0, \dot{\theta}(t_f) = 0 \quad (\text{I.13})$$

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

$$\left\{ \begin{array}{l} \theta_0 = a_0 \\ \theta_f = a_0 + a_1 \cdot t_f + a_2 \cdot t_f^2 + a_3 \cdot t_f^3 \\ a_1 = 0 \\ a_1 + 2 \cdot a_2 \cdot t_f + 3 \cdot a_3 \cdot t_f^2 = 0 \end{array} \right. \quad (\text{I.14})$$

En résolvant le système d'équations on obtient :

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

Où, t_f la durée de mouvement ;

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

I.5.3. mouvement continu entre points via

Dans le cas de mouvement continu aux points via, les fonctions linéaires connectent les points via et les segments paraboliques sont ajoutés autour de ces derniers. Considérons 3 points correspondant aux angles $(\theta_j, \theta_k, \theta_l)$ avec les notations suivantes :

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

$$\begin{cases} \dot{\theta}_{jk} = \theta_k - \theta_j / t_{djk} \\ \ddot{\theta}_k = \text{sgn}(\dot{\theta}_{kl} - \dot{\theta}_{jk}) * |\ddot{\theta}_k| \\ t_k = \dot{\theta}_{kl} - \dot{\theta}_{jk} / \ddot{\theta}_k \\ t_{jk} = t_{djk} - \frac{1}{2} * t_j - \frac{1}{2} * t_k \end{cases} \quad (\text{I.15})$$

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

$$\begin{cases} \dot{\theta}(t_1) = \ddot{\theta}_1 * t_1 = (\theta_2 - \theta_1) / (t_{d12} - \frac{1}{2} * t_1) \\ \ddot{\theta}_1 = \text{sgn}(\theta_2 - \theta_1) * |\ddot{\theta}_1| \end{cases} \quad (\text{I.16})$$

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

$$\begin{cases} t_1 = t_{d12} - \sqrt{t_{d12}^2 - 2 * (\theta_2 - \theta_1) / \ddot{\theta}_1} \\ \dot{\theta}_{12} = (\theta_2 - \theta_1) / (t_{d12} - \frac{1}{2} * t_2) \\ t_{12} = t_{d12} - t_1 - \frac{1}{2} * t_2 \end{cases} \quad (\text{I.17})$$

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

$$\left\{ \begin{array}{l} \ddot{\theta}_n * t_n = (\theta(n-1) - \theta_n) / (t_d(n-1)n - \frac{1}{2} * t_n) \\ \ddot{\theta}_n = \text{sgn}(\theta(n-1) - \theta_n) * |\ddot{\theta}_n| \end{array} \right. \quad (\text{I.18})$$

Ce qui conduit à la solution :

$$\left\{ \begin{array}{l} t_n = t_d(n-1)n - \sqrt{t_d(n-1)n^2 - 2 * (\theta_n - \theta(n-1)) / \ddot{\theta}_n} \\ \dot{\theta}(n-1)n = (\theta_n - \theta(n-1)) / (t_d(n-1)n - \frac{1}{2} * t_n) \\ t(n-1)n = t_d(n-1)n - t(n-1) - \frac{1}{2} * t_n \end{array} \right. \quad (\text{I.19})$$

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

I.6. Conclusion

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

Puis nous avons passé à la modélisation des robots manipulateurs qui est basée sur la modélisation de la structure mécanique, les actionneurs et les capteurs. Elle permet la détermination des relations directes et inverses entre les coordonnées généralisées et les coordonnées cartésiennes et leurs dérivées respectives.

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

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

CHAPITRE II : COMMANDE OPTIMALE & OPTIMISATION

II.1 - Introduction

La conception classique du système de commande est généralement un processus d'épreuve-et-erreur dans lequel diverses méthodes d'analyse sont employées itérativement afin de déterminer le paramètre de conception d'un système « acceptable ».

L'objectif de la commande optimale est de déterminer les signaux de contrôle d'un processus qui causeront la satisfaction des contraintes physiques et au même temps minimiser (ou maximiser) quelques critères de performance. Une performance acceptable est généralement définie en termes de temps et de critère de domaine de fréquence tel que le temps de montée, temps de stabilisation, crête de dépassement, gain et marge de phase et le débit.[12]

Généralement, il y a deux approches qui permettent de résoudre un problème de commande optimale :

- Les méthodes indirectes basées sur un modèle mathématique du système (par exemple en utilisant le principe du maximum de Pontryagin).
- Les méthodes directes consistant à résoudre le problème discrétisé par les techniques d'optimisation paramétrique.

Les méthodes indirectes sont connues pour être plus précises par rapport aux méthodes directes mais leur inconvénient est qu'elles nécessitent le calcul de l'Hamiltonien qui est souvent très difficile sauf dans les cas de systèmes simples. En pratique, les méthodes directes sont largement utilisées. Pour la plupart des problèmes, ces méthodes exigent moins d'interaction pour définir le problème et surtout elles ne demandent pas le calcul analytique des dérivées. [13]

II .2 - La commande optimale

En référence à la figure 1, nous sommes intéressé à trouver la commande optimale $\mathbf{u}^*(\mathbf{t})$ (indique l'état optimal) qui mènera l'usine P de l'état initial à l'état final avec quelques contraintes sur les commandes et les états et au même temps extrémiser l'index de performances données. La formulation du problème de la commande optimale requiert :

- Une description mathématique (ou modèle) du processus à commander (généralement dans l'état de forme variable).
- Les spécifications de l'index de performances.
- Un rapport des conditions de frontière et des contraintes physiques sur les états et/ou les commandes. [14]

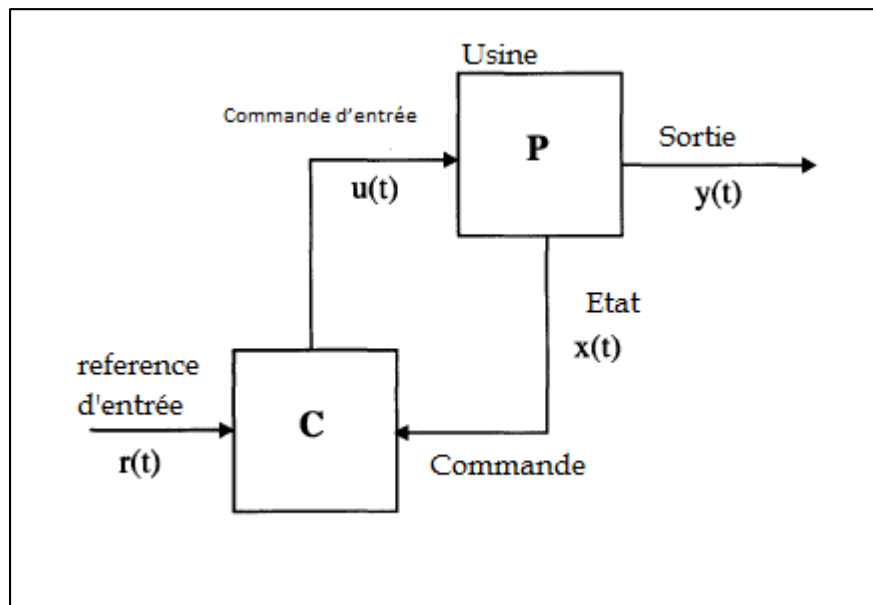


Figure II.1 : La commande optimale

II.3- Problème de la commande optimale

Le problème de la commande optimale sans contraintes peut être énoncé comme suit :
 Etant 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 :

$$J = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t) dt \quad (\text{II.1})$$

Sachant que :

$$\dot{x}(t) = f(x(t), u(t), t), x(t_0) = x_0 \quad (\text{II.2})$$

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

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

Ces méthodes seront discutées brièvement dans les sections suivantes.

En général, il n'est pas possible de résoudre analytiquement le problème (II.1), (II.2). Cependant une solution analytique existe pour un cas particulier de ce problème qui s'énonce comme suit : Trouver la commande optimale qui minimise la fonctionnelle :

$$J = x^T(t_f)Sx(t_f) + \int_{t_0}^{t_f} (x^T Qx + u^T Ru) dt \quad (II.3)$$

Sachant que :

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), x(t_0) = x_0 \quad (II.4)$$

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

$$u^*(x, t) = -R^{-1}B^T(t)P(t)x(t) \quad (II.5)$$

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

II.4- Programmation dynamique

L'utilisation du principe d'optimalité pour dériver une équation permettant de résoudre le problème de la commande optimale était proposée en premier lieu par Bellman [14]. L'application de ce principe au problème continu de la commande optimale a conduit à l'invention de la fameuse équation de HJB.

Pour le problème de la commande optimale (II.1), (II.2), l'équation de HJB est donnée par :

$$\frac{\partial J^*(x(t), t)}{\partial t} = -\min\{L(x(t), u(t), t) + \frac{\partial J^*(x(t), t)}{\partial t} f(x(t), u(t), t)\} \quad (II.6)$$

Et la condition finale est :

$$J^*(x(t_f), t_f) = \phi(x(t_f), t_f) \quad (II.7)$$

La solution du problème (II.6)-(II.7) permet de trouver la commande optimale u^* en fonction de $x(t)$ et t . La dérivation de l'équation de HJB peut être trouvée dans plusieurs manuels standards de la commande optimale. Cette équation est une condition suffisante d'optimalité, elle est satisfaite pour tout couple $(x(t), t)$. Un avantage de l'utilisation de l'approche HJB est de résoudre le problème de la commande optimale pour obtenir la loi de commande en boucle fermée.

En général, l'équation de HJB ne possède pas une solution analytique. Cependant, pour le problème de la commande optimale quadratique, (II.3)-(II.4), l'équation de HJB se réduit à l'équation différentielle de Riccati :

$$-P(t) = A^T P(t) + p(t)A(t) + Q - P(t)B(t)R^{-1}B^T(t)p(t) \quad (\text{II.8})$$

$$p(t_f) = S \quad (\text{II.9})$$

Ce résultat peut être obtenu si la valeur $J^* = x^T P(t)x$ est substituée dans l'équation de HJB. Citons qu'une nouvelle notion de solution appelée la solution de viscosité a été introduite récemment par [15].

II.5 Conditions Nécessaires d'Optimalité

II.5.1- Equations d'Euler-Lagrange

Les conditions nécessaires d'optimalité peuvent être déduites à partir des méthodes de calculs variationnels qui sont basées sur le fait qu'à chaque point stationnaire la variation de la fonction coût disparaît pour une variation arbitraire de la commande [15].

Pour résoudre le problème (II.1)-(II.2), nous utiliserons les multiplicateurs de Lagrange $\lambda(t) \in R^n$ Par conséquent l'indice de performance est donné par la formule suivante :

$$J_A = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} [L(x(t), u(t), t) + \lambda^T(t)(f(x(t), u(t), t) - \dot{x})] dt \quad (\text{II.10})$$

Introduisons le Hamiltonien H défini par :

$$H(x(t), u(t), \lambda(t), t) = L(x(t), u(t), t) + \lambda^T(t)f(x(t), u(t), t) \quad (\text{II.11})$$

Nous pouvons, ainsi, écrire (II.10) sous la forme :

$$J_A = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} H(x(t), u(t), \lambda(t), t) dt - \int_{t_0}^{t_f} \lambda^T(\dot{x}(t)) dt \quad (\text{II.12})$$

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

$$\int_{t_0}^{t_f} \lambda^T \dot{x}(t) dt = \lambda^T(t_f)x(t_f) - \lambda^T(t_0)x(t_0) - \int_{t_0}^{t_f} \dot{\lambda}^T(t)x(t) dt \quad (\text{II.13})$$

Cependant (II.12) devient :

$$J_A = \varphi(x(t_f), t_f) - \lambda^T(t_f)x(t_f) + \lambda^T(t_0)x(t_0) + \int_{t_0}^{t_f} [H(x(t), u(t), \lambda(t), t) + \dot{\lambda}^T(t)x(t)] dt \quad (\text{II.14})$$

Le problème original, (II.1)-(II.2), a été converti à un problème de minimisation, (II.14), sans contraintes. Pour achever la stationnarité, l'effet de la variation de la commande sur la

fonction coût doit être nul pour $t_0 \leq t \leq t_f$. En considérant t_0 et t_f comme étant fixes, alors la première variation de la commande due à la variation de J_A est :

$$\delta J_A = \left[\left(\frac{\partial \varphi}{\partial x} - \lambda^T \right) \delta x \right]_{t=t_f} + \left[\lambda^T \delta x_0 \right]_{t=t_0} + \int_{t_0}^{t_f} \left[\frac{\partial H}{\partial u} \delta u + \left(\frac{\partial H}{\partial x} + \dot{\lambda}^T \right) \delta x \right] dt \quad (\text{II.15})$$

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

$$\dot{\lambda}^T (t) = - \frac{\partial H}{\partial x} = - \lambda^T \frac{\partial f}{\partial x} - \frac{\partial L}{\partial x} \quad (\text{II.16})$$

Avec la condition finale :

$$\lambda^T (t_f) = \left[\frac{\partial \varphi}{\partial x} \right]_{t=t_f} \quad (\text{II.17})$$

Comme la condition initiale $x(t_0)$ est fixe, ceci implique que $\delta x(t_0)$ disparaît, et l'équation (II.15) se réduit à :

$$\delta J_A = \int_{t_0}^{t_f} \left[\frac{\partial H}{\partial u} \delta u \right] dt \quad (\text{II.18})$$

Pour un minimum local, il est nécessaire que δJ_A disparaisse pour δu quelconque, donc il est nécessaire que :

$$\frac{\partial H}{\partial u} = \left(\frac{\partial f}{\partial u} \right)^T \lambda + \left(\frac{\partial L}{\partial u} \right)^T = 0 \quad (\text{II.19})$$

Pour tout $t_0 \leq t \leq t_f$:

Les équations (II.2), (2.16), (2.17) et (II.19) sont des conditions nécessaires qui doivent être satisfaites par les solutions optimales du problème, quand l'instant final est fixe. Ces équations sont connues sous l'appellation Equations d'Euler-Lagrange.

En résumé, pour obtenir la commande optimale $u^*(t)$ qui minimise l'indice de performance (II.1) sous la condition (II.2) les équations suivantes doivent être résolues :

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

$$x(t_0) = x_0 \quad (\text{II.21})$$

$$\dot{\lambda}^T(t) = - \left(\frac{\partial f}{\partial x} \right)^T \lambda - \left(\frac{\partial L}{\partial x} \right)^T \quad (\text{II.22})$$

$$\lambda(t_f) = \left(\frac{\partial \varphi}{\partial x} \right)^T \quad (\text{II.23})$$

$$\frac{\partial H}{\partial u} = \left(\frac{\partial f}{\partial u} \right)^T \lambda + \left(\frac{\partial L}{\partial u} \right)^T = 0 \quad (\text{II.24})$$

Remarque

1-Dans le cas où le temps final n'est pas fixe, une autre condition nécessaire doit être donnée. Cette condition est dérivée en variant la fonction coût par rapport au temps. La condition nécessaire obtenue est :

$$\left(\frac{\partial \varphi}{\partial t} + H \right)_{t=t_f} = 0 \quad (\text{II.25})$$

2-Dans le cas où $x(t_f)$ est fixe :

$$x(t_f) = x_f \quad (\text{II.26})$$

Donc, seule la condition (II.20) doit être remplacée par (II.23).

II.5.2 - Principe Du Minimum De Pontryagin

Dans les problèmes réels, les variables de commande sont, d'habitude bornées donc le Hamiltonien ne peut pas être différenciée par rapport à la commande. Dans ce cas les conditions nécessaires sont dérivées par le principe du minimum de Pontryagin [16] qui s'énonce ainsi :

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

$$\dot{\lambda} = - \frac{\partial H}{\partial x} \quad (\text{II.27})$$

$$\lambda(t_f) = \left(\frac{\partial \varphi}{\partial x} \right)^T \quad (\text{II.28})$$

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

Pour tout $t \in [t_0 t_f]$ et pour tout $u(t) \in U \subset R_m$. Donc la commande optimale doit minimiser le Hamiltonien défini par (II.11). L'inégalité (II.26) est très utilisée pour l'obtention de la commande optimale si la commande est bornée. Il est à noter que le principe du minimum est une généralisation de l'approche du calcul variationnel. Il faut aussi signaler que l'approche variationnel et le principe du minimum mènent à un problème connu comme « A non linear two-point boundary problem : NLTPBP » qui est difficile à résoudre de façon analytique. Les méthodes numériques proposées pour résoudre ces problèmes sont appelées méthodes indirectes par opposition aux méthodes directes basées sur la transformation du problème de la commande optimale à un problème de programmation non linéaire.

II.6- Méthodes Indirectes

On distingue deux types : les méthodes en boucle fermée et les méthodes en boucle ouverte :

II.6.1- Méthodes En Boucle Fermée

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

- La première approche pour obtenir une approximation de la commande optimale en boucle fermée est basée sur l'utilisation du développement en séries de puissance pour résoudre soit l'équation de HJB soit le NLTPBP. Cette approche est utilisée par Lukes [17] pour trouver une solution approximative de l'équation de HJB relatif à la commande optimale à horizon infini. La solution de l'équation de HJB est réduite à la résolution d'un système d'équations algébriques. Utilisant la même idée, Willemstein [18] a développé la méthode de Lukes pour résoudre le problème de la commande optimale à horizon fini. Le travail de Lukes a été appliqué par Garrard et Jordan à la commande de l'avion F8 [19]. La même technique est utilisée par Nishikawa et al. [20] pour obtenir la solution optimale approximative d'un indice de performance quadratique à horizon fini pour le système perturbé suivant :

$$\dot{x} = A(t)x + \varepsilon f(x, t) + B(t)u \quad (\text{II.30})$$

De la même manière, Yoshida et al. [21] ont pu résoudre le problème de la commande optimale à indice de performance quadratique à horizon fini et infini pour le système décrit par :

$$\dot{x} = f(x) + Bu \quad (\text{II.31})$$

- La seconde approche pour obtenir la commande optimale en boucle fermée est basée sur la linéarisation des conditions nécessaires d'optimalité autour de la solution optimale ou via le développement de l'indice de performance jusqu'à l'ordre 2 et les contraintes jusqu'à l'ordre 1 autour de la solution optimale.

- La troisième approche est basée sur l'écriture des équations d'état sous forme linéaire :

$$\dot{x} = f(x, u, t) + A(x, u, t)u \quad (\text{II.32})$$

Et, donc, de résoudre le problème de la commande optimale en résolvant l'équation de Riccati.

$$\dot{P}(x, u, t) = P(x, u, t)A(x, u, t) + A^T(x, u, t)P(x, u, t) - P(x, u, t)B(x, u, t)R^{-1}B^T(x, u, t)P(x, u, t) + Q \quad (\text{II.33})$$

Et la commande optimale est donnée par :

$$u^*(x, t) = -R^{-1}B^T(x, u, t)P(x, u, t)x(t) \quad (\text{II.34})$$

- La quatrième approche est basée sur la solution du problème inverse de la commande optimale.

II.6.2- Méthodes De Commande En Boucle Ouverte

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

II.7 - Méthodes Directes

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

Le problème de commande optimale peut être converti à un problème de programmation mathématiques soit en utilisant la technique de discrétisation, soit en se servant de la technique de paramétrisation. Notre thèse est basée sur la deuxième technique.

II.7.1 Méthodes De Discrétisation

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

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

Où les points t_i sont appelés des nœuds.

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

$$\mathbf{z} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{n-1})$$

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

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

$$\mathbf{z} = (\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{n-1})$$

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

II.8.2 - Méthodes de Paramétrisation

La technique de paramétrisation peut être appliquée de trois manières différentes :

a- *Paramétrisation de la Commande* :

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

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

Où les $C_i^{u_k}$ sont les paramètres inconnus et les 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, le problème de la commande optimale est ainsi converti à un problème de programmation non linéaire à paramètres statiques qu'il est facile à résoudre que le problème original.

Plusieurs fonctions ont été utilisées pour l'approximation des variables de la commande [22], parmi lesquelles on peut citer : les fonctions constantes par morceaux, les fonctions linéaires par morceaux, les fonctions splines d'ordre donné.

L'application de cette technique nécessite l'intégration des équations d'état qui est un processus coûteux en termes de temps de calcul [23].

b- Paramétrisation d'Etat-Commande

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

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

$$x_j(t) = \sum_{i=0}^N C_i^{x_j} \varphi_i(t) \quad j = 1, 2, \dots, n \quad (\text{II.37})$$

Où les $C_i^{u_k}, C_i^{x_j}$ sont des paramètres inconnus et les φ_i sont un ensemble de fonctions appropriées. En utilisant cette méthode, le problème de la commande optimale est converti à un problème de programmation mathématique non linéaire.

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

Dans cette thèse, nous avons opté pour cette technique mais comme les fonctions φ_i dans notre cas sont des fonctions à blocs d'impulsion (BPFs), nous avons pu surmonter les inconvénients suscités.

c- Paramétrisation d'Etat

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

$$x_j(t) = \sum_{i=0}^N C_i^{x_j} \varphi_i(t) \quad j = 1, 2, \dots, n \quad (\text{II.38})$$

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

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

Cependant, cette technique souffre de quelques inconvénients :

1. Il est difficile de traiter les systèmes non linéaires car il n'est pas toujours facile de trouver les variables de commande comme fonction des variables d'état.

2. Il n'existe pas une voie systématique pour appliquer cette technique aux problèmes de la commande optimale à aspect général quand le nombre de variable d'état et de commande n'est pas égal.

Cette technique a été appliquée à des cas spéciaux, par exemple au problème de la commande optimale linéaire avec le nombre de variables d'état égal au nombre de variables de commande ou à des systèmes non linéaires mono entrée pouvant être exprimés sous la forme canonique contrôlable. Récemment M.Jaddu a utilisé cette technique via les polynômes de Chebyshev associée à la méthode de quasi-linéarisation pour résoudre le problème de la commande optimale dans son aspect général, cependant sa méthode souffre de la taille du problème qui demeure exorbitant, et surtout de la violation des contraintes.

La technique de paramétrisation peut être employée en utilisant différentes fonctions de base. Les BPFs sont utilisées dans cette thèse pour paramétriser les variables d'état et les variables de contrôle. Les BPFs ont des propriétés exceptionnelles (à voir dans le prochain chapitre) qui permettent d'agir directement sur le problème de la commande optimale quelque soit sa nature.

L'utilisation des fonctions à blocs d'impulsion pour résoudre le problème de la commande optimale n'est pas nouvelle. V. Purnachandra et K. Ranganatha ont utilisé les BPFs pour déterminer la commande en boucle fermée d'un problème de commande optimale linéaire, P. Sannuti a utilisé les BPFs pour l'analyse et la synthèse des systèmes dynamiques linéaires, Shien-Yu Wang [13] a pu résoudre le problème de la commande optimale des systèmes non linéaires à retard en utilisant les BPFs, S. Lapin et N.D. Egupov ont analysé les systèmes linéaires et non linéaires à coefficients variables via les BPFs. A ce stade les BPFs sont utilisés avec succès dans l'étude du problème de la commande optimale sans contraintes. Récemment N. Boussiala et H. Chaabi ont utilisé les BPFs pour résoudre le problème de la commande optimale des systèmes non linéaires multidimensionnels soumises à des contraintes. Nous verrons dans cette thèse qu'il est possible de traiter le problème de la commande optimale sujet à divers types de contraintes.

II.8 Méthodes générales d'optimisation [25]

L'optimisation est un dispositif très souhaitable dans la vie quotidienne, on aime travailler et employer notre temps et nos ressources d'une manière optimale. Le sujet de l'optimisation est tout à fait général dans le sens qu'il peut être vu à partir de différentes perspectives selon l'approche (algébrique ou géométrique), l'intérêt (singulier ou multiple), la nature des signaux (déterministe ou stochastique) et l'étape (singulière ou multiple) utilisée en optimisation.

La recherche continue de l'homme pour cette optimisation a donné naissance à plusieurs méthodes et techniques. [12]

II.8.1 Définition de l'optimisation

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

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

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

Par contre, x est un minimum global de f sur S si son image par f est inférieure ou égale à celle de tous les autres points du domaine S , et pas seulement dans un certain voisinage. Bien évidemment, tout minimum global est aussi un minimum local.

II.8.2 Optimisation locale

II.8.2.1 Méthode de descente

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

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

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

II.8.2.2 Algorithme génétique

Leur but est d'obtenir une solution approchée à un problème d'optimisation, lorsqu'il n'existe pas de méthode exacte (ou que la solution est inconnue) pour le résoudre en un temps raisonnable. Les algorithmes génétiques utilisent la notion de sélection naturelle et l'appliquent à une population de solutions potentielles au problème donné. La solution est approchée par « bonds » successifs, comme dans une procédure de séparation et évaluation, à ceci près que ce sont des formules qui sont recherchées et non plus directement des valeurs. [26]

II.8.3 Optimisation globale

II.8.3.1 Méthode des variations locales

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

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

II.8.3.2 Algorithme par séparation et Evaluation

Cette méthode est généralement connue sous le nom de "branch and bound".

Soit " \mathcal{S} " un ensemble fini. On dispose d'une fonction " f " qui, pour toute solution réalisable " x " de " f ", renvoie un coût " $f(x)$ ". Le but du problème est de trouver la solution réalisable " x " de coût minimal. D'un point de vue purement existentiel, le problème est trivial : une telle solution " x " existe bien car l'ensemble " \mathcal{S} " est fini. En revanche, l'approche effective du problème se confronte à deux difficultés. La première est qu'il n'existe pas forcément un algorithme simple pour énumérer les éléments de " \mathcal{S} ". La seconde est que le nombre de solutions réalisables est très grand.

Comme son nom l'indique, la méthode se fait en deux phases essentielles qui sont :

La phase de séparation qui consiste à diviser le problème en un certain nombre de sous-problèmes qui ont chacun leur ensemble de solutions réalisables de telle sorte que tous ces ensembles forment un recouvrement (idéalement une partition) de l'ensemble. Ainsi, en résolvant

tous les sous-problèmes et en prenant la meilleure solution trouvée, et de borner les solutions dans chaque sous-espace.

La phase d'évaluation des nœuds de l'arbre de recherche Sur le plan pratique : le problème de départ constitue le problème racine de l'arbre de recherche. On calcule une évaluation de la solution recherchée en essayant de borner inférieurement et supérieurement les solutions du domaine entière.

L'aboutissement de la méthode à une solution optimale se fait quand les bornes inférieure et supérieure coïncident, Sinon, on divise le domaine de recherche en deux sous-problèmes, que l'on va étudier séparément.

Pour déterminer qu'un ensemble de solutions réalisables ne contient pas de solution optimale, la méthode la plus générale consiste à déterminer une borne inférieure pour le coût des solutions contenues dans l'ensemble (s'il s'agit d'un problème de minimisation). Si on arrive à trouver une borne inférieure de coût supérieur au coût de la meilleure solution trouvée jusqu'à présent, on a alors l'assurance que le sous-ensemble ne contient pas l'optimum (solution optimale).

II.8.3.3 Branch and prune

Cette méthode, est principalement similaire à la précédente dans la création de l'arbre de recherche et la division du problème en sous-problème.

Il est question suivre un domaine qui a les caractéristiques de contenir la solution optimale, mais il n'est pas toujours correct et précis pour être testé.

Dans le cas où un ensemble ne contient aucun point dont la dérivée est nulle. Mais si on sait qu'un ensemble contient au moins un point de dérivée nulle mais son identification est imprécise, alors on sépare les intervalles et on agit comme la méthode précédente.

II.8.4 Optimisation stochastique

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

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

Où f est une fonction continue sur $S \in R^n \rightarrow R$.

II.8.4.1 Méthode adaptative

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

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

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

II.8.5 Importance du gradient [25]

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

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

La méthode de plus grande pente consiste en une série de minimisation unidimensionnelle, chacune suivant la direction de la plus grande pente ou la minimisation commence. Bien sûr, la direction du gradient n'est pas constante, on s'attend à de multiples itérations pour trouver le minimum, mais on montre que la méthode est convergente pour une fonction quadratique.

Note : la direction de plus grande pente est évidemment une combinaison de variables. On y remarque une propriété des directions successives de recherche : si chaque minimisation par

recherche linéaire est exacte, les directions successives sont orthogonales, ce qui n'est pas optimale pour une recherche efficace. Dans cette méthode, on part de x_0 (vecteur initial) et on calcule le vecteur gradient :

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

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

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

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

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

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

Pour les formes quadratiques définies positives

Une forme quadratique générale est :

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

Où

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

et

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

Cette fonction a un minimum si et seulement si $H \geq 0$ (si $H = 0$, le minimum est à l'infini, car la dérivée seconde étant nulle, la fonction est plate. Si H est négatif, alors on se

trouve sur une parabole de courbure négative, donc sans minimum). Le minimum, s'il existe, est à $x = -H^{-1}g$. En effet, au minimum, la dérivée de F est nulle, donc :

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

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

Si $H < 0$, le remède serait de prendre un pas de $x = -g$, c'est-à-dire mettre arbitrairement H égale à 1, de manière à ce que le pas soit au moins dans la bonne direction (celui de l'inverse du gradient ou croit la fonction), même si ce pas a maintenant une longueur arbitraire.

II.9 Interpolation et approximation de courbes

II.9.1 Hystorique [28]

Les problèmes interpolations et approximation sont un vaste sujet qui s'étend de l'ajustement de mesures à la conception d'images de synthèse en passant par la création de polices de caractères. Le comte de Lagrange s'est penché sur les problèmes d'interpolation au XVIIIe siècle mais il faut attendre l'arrivée des ordinateurs pour que le domaine se développe réellement.

Au XVIIIe siècle, l'interpolation et l'approximation servaient surtout à relier ou approcher des mesures, l'intérêt était avant tout scientifique.

Avec la révolution industrielle, les machines sont apparues, il a fallu dessiner les pièces pour pouvoir les produire. Pour tracer des courbes, les dessinateurs utilisaient des méthodes manuelles qui reposaient sur la déformation de lames de métal, de ressorts et l'utilisation de pistolets. Pour les surfaces, des gabarits étaient construits en 3 dimensions.

Puis vers 1950 les machines à commandes numériques sont arrivées, il devenait obligatoire d'exprimer ces courbes mathématiquement. La première approche a été de numériser le travail des dessinateurs mais cela était long et coûteux, la nécessité de trouver des courbes capables d'être utilisées depuis la conception jusqu'à la réalisation s'est donc fait sentir.

Les splines cubiques ont été la première méthode mise au point, puis les courbes de Bézier sont arrivées, avec une conception différente, plus souple. L'évolution s'est poursuivie avec les B-splines.

II.9.2- Principe de l'interpolation

Lorsque nous voulons interpoler des points, la manière la plus simple consiste à les relier par des droites. Bien sûr, cette solution n'est pas du tout satisfaisante car elle n'est pas du tout harmonieuse. Pour avoir un résultat acceptable, il faudrait mettre les points très rapprochés les uns des autres. Il devient alors très difficile de manipuler l'ensemble. C'est pour ces raisons que nous cherchons des méthodes plus perfectionnées.

II.9.3 Interpolation quadratique [29]

L'interpolation quadratique est l'une des méthodes les plus simples. Le but est de relier les points par des courbes de degré deux. Mais si nous nous contentons de faire passer une parabole par les trois premiers points puis une autre par les trois suivants, la courbe comportera des cassures. En effet, à la jonction de deux paraboles, la pente ne sera pas la même. Ce procédé ne présente donc aucun avantage sur l'interpolation linéaire.

Pour supprimer ces cassures et obtenir une courbe fluide, nous modifions les conditions.

La courbe doit passer par deux points imposés, en un point, la pente doit être égale à la pente de la fin de la parabole précédente.

II.9.3.1 Inconvénients

Le choix de la pente initiale est très important. La courbe peut avoir une allure complètement différente, ce qui rend les courbes complètement imprévisibles.

II.9.4 L'interpolation cubique [28], [29]

L'interpolation cubique repose sur le même principe que l'interpolation quadratique.

Les conditions sont les suivantes :

- La courbe doit passer par deux points imposés.
- Nous imposons les tangentes en ces deux points de façon intelligente. Généralement, la tangente en un point doit être parallèle à la droite reliant le point précédent au point suivant.

Ce type d'interpolation est aussi appelé interpolation d'Hermite cubique.

II.9.4.1 Avantages et inconvénients

L'interpolation cubique ne présente pas les mêmes défauts que la quadratique. Ainsi des points d'inflexion sont possibles ailleurs que sur les points d'interpolation.

La courbe est très robuste.

L'inconvénient de cette interpolation cubique est que si la pente ne possède pas de "cassure", le rayon de courbure présente des "cassures". Cela nuit à la fluidité de la courbe.

II.9.5. L'interpolation de Lagrange [28][29]

Au lieu, d'interpoler des points par arcs de paraboles ou de cubiques, l'interpolation de Lagrange permet de relier tous les points à l'aide d'une seule fonction polynomiale.

Le problème est donc de trouver un polynôme qui vaut :

$$y_0 \text{ en } x_0, y_1 \text{ en } x_1, \dots, y_n \text{ en } x_n$$

Avec:

$$x_0 < x_1 < \dots < x_n$$

II.9.5.1 Avantages et inconvénients [30][31]

L'avantage de l'interpolation de Lagrange est qu'elle est relativement simple, et qu'il est possible de relier beaucoup de points avec une seule fonction polynomiale.

Le principale désavantage de l'interpolation de Lagrange est que plus nous prenons de points, plus la courbe oscille entre ces points. L'interpolation de Lagrange n'est donc pas une très bonne méthode.

II.9.6. Interpolation par splines cubiques [28]

L'interpolation par splines cubiques repose sur le même principe que l'interpolation cubique. La différence est que nous voulons non seulement la continuité de la pente mais aussi du rayon de courbure.

II.9.6.1 Avantages et inconvénients [29]

Les courbes tracées avec cette méthode sont très harmonieuses. Elle comportent pas de cassure du rayon de courbure.

Le désavantage est que la courbe est beaucoup moins robuste.

II.9.7. Les B-Splines (principe) [31], [29]

C'est la méthode utilisée dans notre mémoire et elle sera mise en application dans le chapitre pratique. Il s'agit de fabriquer une courbe qui présente tous les avantages des autres courbes mais sans ses inconvénients. Ainsi, la courbe devra approximer les points de contrôle, ce qui la rend simple à manipuler.

II.9.7.1 Fonctions de Base [32]

L'idée principale des B-Splines est de remplacer les polynômes par des fonctions. Ensuite, nous sommerons ces fonctions avec les points de contrôles pour obtenir la courbe.

Une B-Spline ne dépend pas uniquement des points de contrôle mais aussi d'un vecteur noeud.

Nous choisissons un vecteur noeud composé de $t_0, t_1, t_2, \dots, t_m$ que nous noterons

$$T = \{t_0, t_1, t_2, \dots, t_m\}$$

Nous choisissons $t_0 \leq t_1 \leq \dots \leq t_m$ Nous prendrons des nombres réels comme noeuds.

Nous choisissons également des points $p_0, p_1, p_2, \dots, p_m$

II.10 Description de la fonction *Fmincon* de MATLAB [33]

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

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

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

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

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

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

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

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

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

Alors on a un problème de la forme suivante :

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

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

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

II.11 Conclusion

Dans ce chapitre, les différentes méthodes de commande optimale ont été rappelées brièvement. La méthode adoptée dans le cadre de cette thèse s'inscrit dans la catégorie des méthodes directes avec paramétrisation commande – état via les fonctions à blocs d'impulsion (BPFs) dont le chapitre suivant expose de façon détaillée l'outil mathématique ainsi que les applications inhérentes nous avons aussi présenté en fonction de nos besoins ; dans le cas générale on utilise les actionneurs électriques quand il s'agit d'une articulation rotoïde et pour soulever des poids légers. Par contre on utilise les actionneurs pneumatiques ou hydrauliques dans le cas d'une articulation prismatique et aussi quand il est question de soulever un poids plus important.

Ensuite, les différentes méthodes de commande optimale ont été rappelées brièvement. Suivies d'un rappel sur la commande optimale avec les diverses méthodes de résolution du

problème de la commande optimale : méthodes directes, méthodes indirectes. A la fin du chapitre, nous avons cité les différentes méthodes d'interpolation de courbe y compris Le **B-spline** qui est utilisé dans la génération de notre trajectoire.

Le problème d'optimisation et les méthodes de sa résolution y sont expliqués. En particulier, la fonction *Fmincon* du logiciel MATLAB a été introduite.

CHAPITRE III : TRAJECTOIRE OPTIMALE D' UN ROBOT
À 2 DDL

III.1 Introduction

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

Dans ce chapitre nous allons faire une étude détaillée d'une méthode de planification d'une trajectoire optimale en gardant inchangée la position initiale et finale du robot, cette méthode a pour objectif la minimisation du coût (l'effort fournis par le robot) et la maximisation du poids que peut soulever le robot. [33][34]

Initialement, nous allons commencer par générer la trajectoire par la méthode B-spline. qui nous permettra d'avoir les points de contrôle à optimiser. L'expression analytique du gradient de la fonction de coût par rapport à ces points de contrôle permet à la procédure d'optimisation de converger facilement vers le minimum et la rend plus robuste. On aura ainsi transformé le problème de recherche d'une commande optimale qui minimise le critère J_c en un problème d'optimisation.

III.2. Modèle dynamique d'un bras manipulateur rigide à deux degrés de liberté

Pour le calcul du modèle dynamique d'une chaîne cinématique (**Fig III.1**), on utilise la formulation de lagrange décrit précédemment au chapitre I (I.8), et l'on définit les paramètres suivants :

m_i : masse du corps i de la chaîne cinématique.

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

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

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

$$\dot{r}_{c1} = w_1 \times r_{c1}$$

$\|\dot{r}_{c1}^2\| = \dot{\theta}_1^2 \frac{l_1^2}{4}$: vitesse du centre de masse du segment 1.

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

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

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

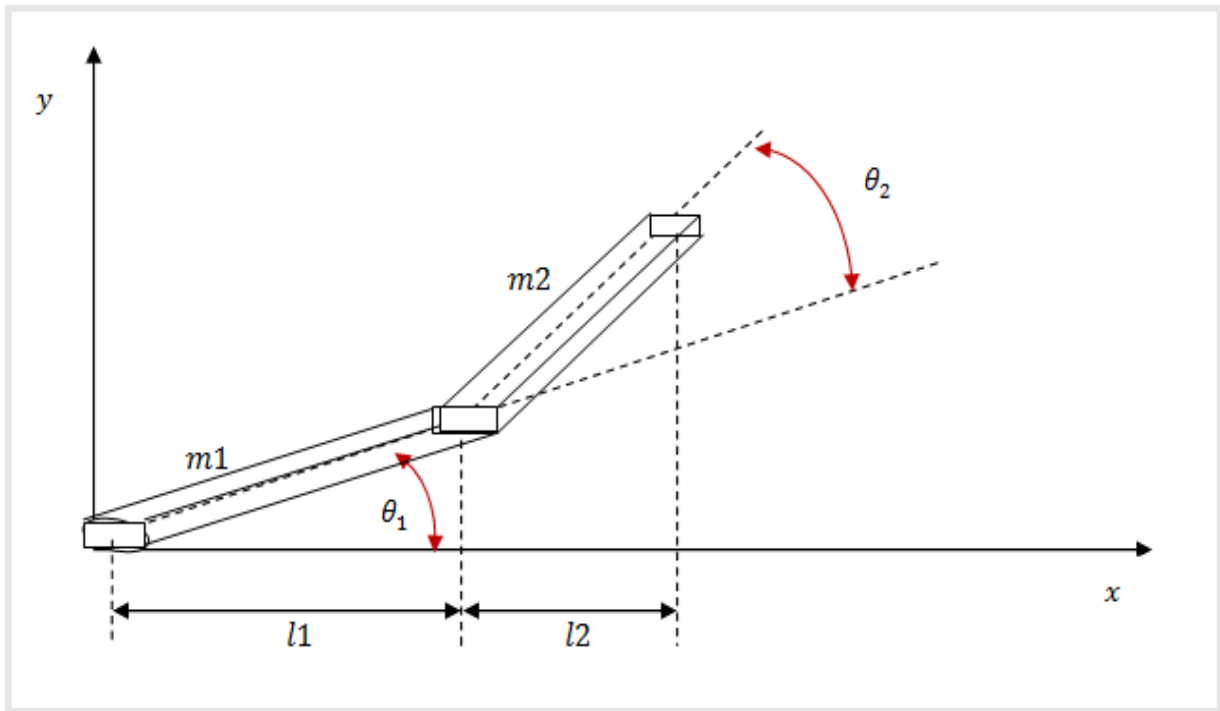


Fig.III.01 représentation d'un bras manipulateur à 2 ddl

III.2.1. Modèle dynamique d'un robot à 2 ddl sans charge sur l'effecteur [37]

➤ Calcul de l'énergie cinétique :

L'énergie cinétique du segment 1 :

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

$$E_{c1} = \frac{l_1^2}{8} m_1 \dot{\theta}_1^2 + \frac{(l_1^2 + b_1^2)}{24} m_1 \dot{\theta}_1^2 \tag{III.1}$$

L'énergie cinétique du segment 2 :

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

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

L'énergie cinétique totale :

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

➤ Calcul de l'énergie potentielle :

L'énergie potentielle du segment 1 :

$$E_{P1} = \frac{m_1}{2} l_1 g \sin \theta_1 \quad (\text{III.3})$$

L'énergie potentielle du segment 2 :

$$E_{P2} = m_2 g l_1 \sin \theta_1 + m_2 g \frac{l_2}{2} \sin(\theta_1 + \theta_2) \quad (\text{III.4})$$

L'énergie potentielle totale :

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

➤ Calcul du lagrangien :

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

➤ Les équations du mouvement

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

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

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

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

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

La dérivée du lagrangien par rapport à θ_1 est :

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

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

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

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

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

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

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

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

Donc le couple est :

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

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

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) \quad (\text{III.5})$$

avec : $M(\theta) =$

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

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

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

Où :

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

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

III.2.2. Modèle dynamique d'un robot à 2 ddl avec un poids (p_w) sur l'organe terminal :

➤ Calcul de l'énergie cinétique :

$$\dot{T} = E_{c1} + E_{c2} + E_{cpw} \quad (\text{III.6})$$

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

$$E_{cpw} = \frac{p_w}{2g} l_1^2 \dot{\theta}_1 + \frac{p_w}{2g} l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{p_w}{g} l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2 \quad (\text{III.7})$$

➤ L'énergie potentielle

$$V_{pw} = p_w [l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)] \quad (\text{III.8})$$

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

➤ Lagrangien :

$$\dot{L} = \dot{T} - \dot{V} \quad (\text{III.9})$$

➤ Les équations du mouvement :

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

$$\tau = \dot{M}(\theta)\ddot{\theta} + \dot{h}(\theta, \dot{\theta}) \quad (\text{III.10})$$

Avec :

$\dot{M} =$

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

$h' =$

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

III.3 Génération de la trajectoire par B-spline

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

La séquence de nœuds :

$$0 = t_0 = \dots = t_{K-1} \leq t_K \leq \dots \leq t_m \leq t_{m+1} = \dots = t_{m+K} = t_f \quad (\text{III.11})$$

Avec : $m \geq K - 1$

Les points de contrôle P_j :

$$P_j = \{p_0, \dots, p_m\} \quad (\text{III.12})$$

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

$$\theta(t, p) = \sum_{j=0}^m p^j B_{j,K}(t) \quad (\text{III.13})$$

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

$$B_{j,K}(t) = \begin{cases} g_{jl}(t), & \text{si } t_{j+l} \leq t < t_{j+l+1} \\ 0, & \text{si } t < t_{j+out} \geq t_{j+K} \end{cases} \quad (\text{III.14})$$

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

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

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

$$B_{j,K}(t) = \frac{t-t_j}{t_{j+K-1}-t_j} B_{j,K-1}(t) + \frac{t_{j+K}-t}{t_{j+K}-t_{j+1}} B_{j+1,K-1}(t) \quad (\text{III.15})$$

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

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

En commençant par :

$$B_{j,1}(t) = \begin{cases} 1, & \text{si } t_j \leq t < t_{j+1} \\ 0, & \text{sinon} \end{cases} \quad (\text{III.16})$$

➤ Détermination des vitesses et des accélérations :

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

$$\frac{\partial B_{j,K}(t)}{\partial t} = (K - 1) \left[\frac{B_{j,K-1}(t)}{t_{j+K-1}-t_j} - \frac{B_{j+1,K-1}(t)}{t_{j+K}-t_{j+1}} \right] \quad (\text{III.17})$$

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

$$\frac{d}{dt} \theta(t, P) = \sum_{j=0}^m p^j \frac{\partial}{\partial t} B_{j,K}(t) = (K - 1) \sum_{j=0}^{m+1} \frac{p^j - p^{j-1}}{t_{j+K-1}-t_j} B_{j,K-1}(t) \quad (\text{III.18})$$

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

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

$$\frac{d^2}{dt^2} \theta(t) = \sum_{j=0}^m p^j \frac{\partial^2}{\partial t^2} B_{j,K}(t) = (K - 1)(K - 2) \sum_{j=0}^{m+2} \left[\frac{p^j - p^{j-1}}{t_{j+K-1}-t_j} - \frac{p^{j-1} - p^{j-2}}{t_{j+K-2}-t_{j-1}} \right] * \frac{B_{j,K-2}(t)}{t_{j+K-2}-t_j} \quad (\text{III.10})$$

➤ Principales propriétés des fonctions B-Spline :

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

2. la fonction $B_{j,K}$ s'annule en d'hors de l'intervalle $[t_j t_{j+1}[$;
3. pour $t \in [t_j t_{j+1}[$, $0 < B_{j,K} \leq 1$;
4. sur l'intervalle $[t_K t_{m-K}[$, $\sum_{j=0}^{m-K-1} B_{j,K} = 1$;
5. la fonction $B_{j,K}$ est de classe c^∞ à droite de chaque point ;
6. au voisinage d'un nœud de multiplicité r , la fonction $B_{j,K}$ est seulement de classe c^{K-r} ;

III.4 Planification de trajectoire optimale avec chemin contraint et effort minimal

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

$$\left\{ \begin{array}{l} \text{minimiser } j_c = j_w + j_e \\ \text{tel que:} \\ M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) = \Gamma \\ \underline{\Gamma} \leq \Gamma(t) \leq \bar{\Gamma} \\ \underline{\theta} \leq \theta(t) \leq \bar{\theta} \text{ et } p_w > 0 \\ \underline{\dot{\theta}} \leq \dot{\theta}(t) \leq \bar{\dot{\theta}} \\ \theta(0) = \theta_0, \dot{\theta}(0) = 0 \\ \theta(t_f) = \theta_f, \dot{\theta}(t_f) = 0 \end{array} \right. \quad \begin{array}{l} \text{(III. 19)} \\ \text{(III. 20)} \\ \text{(III. 21)} \\ \text{(III. 22)} \\ \text{(III. 23)} \\ \text{(III. 24)} \\ \text{(III. 25)} \end{array}$$

Avec

$$j_w = -w p_w \quad \text{(III.26)}$$

$$j_e = \frac{1}{2} \int_0^{p t_f} \Gamma^T w_e \Gamma dt \quad \text{(III.27)}$$

L'équation(III.5) représente la dynamique inverse du manipulateur où θ est le vecteur de dimension n de coordonnées articulaires. $M(\theta)$ est la matrice d'inertie de dimension $(n \times n)$ et $h(\theta, \dot{\theta})$ contient les effets de coriolis, ceux de gravité et enfin ceux de frottement.

Les contraintes sur les couples articulaires, les coordonnées θ et les vitesses articulaires $\dot{\theta}$ sont données respectivement par les équations (III.13) et (III.18), avec les bornes supérieures et inférieures sur Γ , θ et $\dot{\theta}$ supposées données au départ. Le point soulevé par le manipulateur est p_w .

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

Le cas d'une surcharge des actionneurs

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

$$j_p = \frac{1}{2} \int_0^{p_t t_f} \varphi_+^T W_p \varphi_+ dt \quad (III.28)$$

Avec

$$\varphi_+ = \begin{cases} \varphi & \text{si } \varphi > 0 \\ 0 & \text{sinon} \end{cases} \quad (III.29)$$

$$\varphi = \begin{bmatrix} \tau - \tau_1 \\ \tau - \bar{\tau} \end{bmatrix} \quad (III.30)$$

Donc l'expression finale du coût j_c devient :

$$j_c = j_w + j_e + j_p \quad (III.31)$$

Utilisation du facteur d'échelle

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

Si l'on utilise un facteur d'échelle p_t pour le temps, qui normalise l'espace de temps (l'espace des nœuds dans notre cas), l'équation (III.11) devient :

$$\theta(t, p_t, P) = \theta(\eta, P) = \sum_{j=0}^m p^j B_{j,K}(\eta) \quad (III.32)$$

Avec $\eta = p_t t$.

$$\frac{\partial}{\partial t} B_{j,K}(\eta) = \frac{1}{p_t} \frac{\partial}{\partial \eta} B_{j,K}(\eta) \quad (III.33)$$

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

$$\frac{\partial}{\partial p_t} B_{j,K}(\eta) = -\frac{\eta}{p_t} \frac{\partial}{\partial \eta} B_{j,K}(\eta) \quad (III.34)$$

Ceci implique que :

$$\frac{\partial}{\partial p_t} \theta(\eta, P) = -\frac{\eta}{p_t} \frac{\partial}{\partial \eta} \theta(\eta, P) \quad (III.35)$$

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

$$\left\{ \begin{array}{l} \text{minimiser } j_c = j_w + j_e + j_p \quad \text{(III. 36)} \\ \text{telque :} \\ M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) = \Gamma \quad \text{(III. 37)} \\ \underline{\Gamma} \leq \Gamma(t) \leq \bar{\Gamma} \quad \text{(III. 38)} \\ \underline{\theta} \leq p^j \leq \bar{\theta} \text{ et } (p_t, p_w) > 0 \quad \text{(III. 39)} \\ \underline{\dot{\theta}} \leq \frac{(K-1)(p^j - p^{j-1})}{t_{j+K-1} - t_j} \leq \bar{\dot{\theta}} \quad \text{(III. 40)} \\ p^0 = p^1 = \theta_0, p^{m-1} = p^m = \theta_f \quad \text{(III. 41)} \\ \theta(p_t t_f) = \theta_f, \dot{\theta}(p_t t_f) = 0 \quad \text{(III. 42)} \end{array} \right.$$

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

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

III.5. La Trajectoire initiale

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

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

Pour notre choix initial, nous avons supposé une trajectoire quelconque avec des points bien définis afin de montrer l'efficacité de la méthode utilisée.

On prend :

x : varie de -0.75 jusqu'à 0.75 avec un pas de 0.125

$$y = [0.02 \ 0.1 \ 0.4 \ 0.4 \ 0.45 \ 0.5 \ 0.6 \ 0.5 \ 0.5 \ 0.4 \ 0.1 \ 0.05 \ 0.01] \quad \text{(III.43)}$$

Avec :

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

R : C'est le rayon de centre $R = 0.75$

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

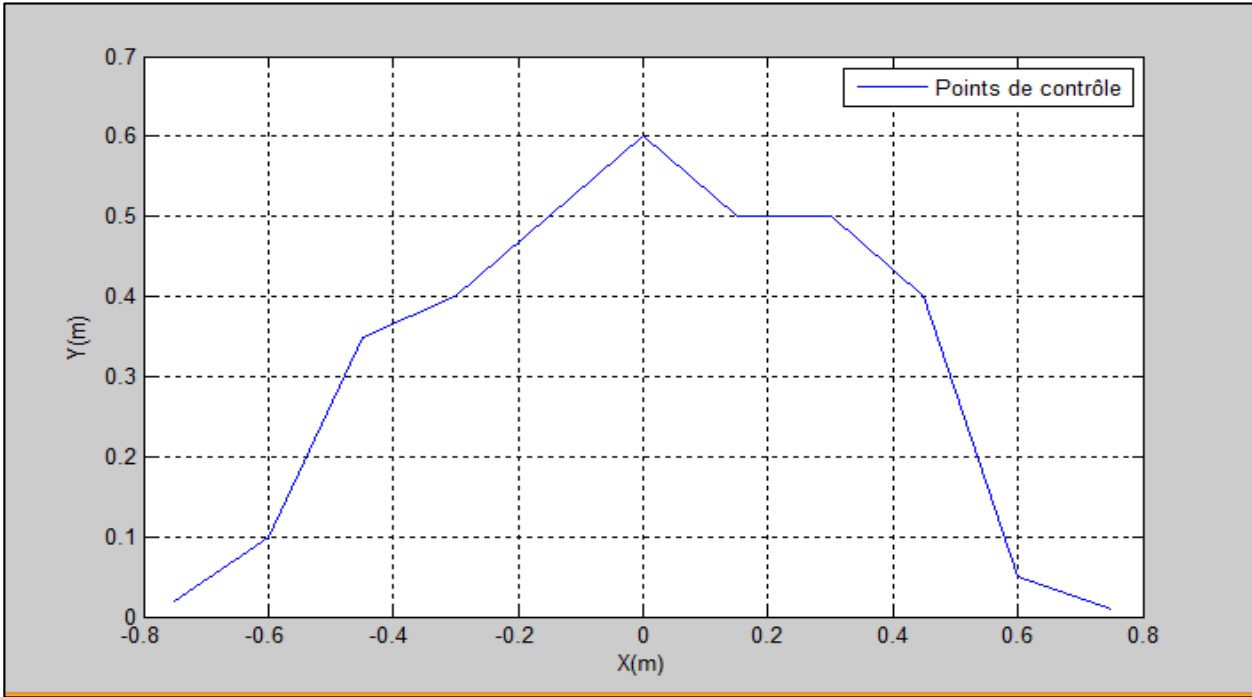


Fig.III.02 : Génération de la trajectoire initiale par les points de contrôle

III.5.1 Détermination des points de contrôle :

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

$$\begin{cases} x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{cases} \quad (III.44)$$

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

$$\begin{cases} \theta_1 = \text{Arctg} \left(\left(\frac{y(l_1+l_2 \cos(\theta_2)) - xl_2 \sin(\theta_2)}{x((l_1+l_2 \cos(\theta_2)) + yl_2 \sin(\theta_2))} \right) \right) \\ \theta_2 = \pm \text{Arcos} \left(\left(\frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2l_1 l_2} \right) \right) \end{cases} \quad (III.45)$$

Résolvant l'équation (III.45), on obtient les résultats rapportés dans le tableau ci-dessous.

Nombre de points	$x(m)$	$y(m)$	$\theta_1(grade)$	$\theta_2(grade)$
1	-0.7500	0.0200	-42.4273	109.4388
2	-0.6250	0.1000	-49.9838	123.3099
3	-0.5000	0.4000	-79.4583	122.4570
4	-0.3750	0.4000	-87.1939	133.0393
5	-0.2500	0.4500	78.0279	136.8864
6	-0.1250	0.5000	63.0719	136.8166
7	0	0.6000	49.0540	127.1090
8	0.1250	0.5000	35.0279	136.8166
9	0.2500	0.5000	22.5055	131.8103
10	0.3750	0.4000	5.9251	133.0393
11	0.5000	0.1000	-29.5975	137.4482
12	0.6250	0.0500	-36.3298	123.9982
13	0.7500	0.01	-40.1374	109.4631

Fig.III.03 : détermination des points de contrôle à partir du modèle inverse

III.5.2 Optimisation de la trajectoire :

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

III.5.2.1 Approximation de la trajectoire par B-splines :

➤ **Calcul de $B_{j,5}$** : En utilisant la formule récursive (III.15) donnée précédemment et on commençant par la formule (III.16) on prend :

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

D'après l'équation(III.11) , on obtient :

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

Le nombre de point est $(m=16)$, $m = \text{length}(tt) - k = 21$ (nombre de neuds) - 5 (l'ordre du B-spline) et on a les point de contrôle p^j qui sont déterminés dans le tableau (**Fig.III.03**) ($p_i^j = \theta_i^j$).

On remarque que le nombre de points ($m=16$) est supérieur au nombre de points du contrôle ($m \geq p_i^j$) ; donc il faut ajouter trois points de contrôle, un au début ($p_i^0 = p_i^1$) et les deux autres à la fin ($p_i^{13} = p_i^{14} = p_i^{15}$).

III.5.2.2 Optimisation

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

Avec :

La fonction objective (III. 36), les contraintes linéaires d'égalités (III.39) et inégalité (III.41) et les limites sur la position (III.42).

La figure (**Fig III.4**) représente un diagramme qui explique procéder pour appliquer la fonction **Fmincon** de MATLAB.

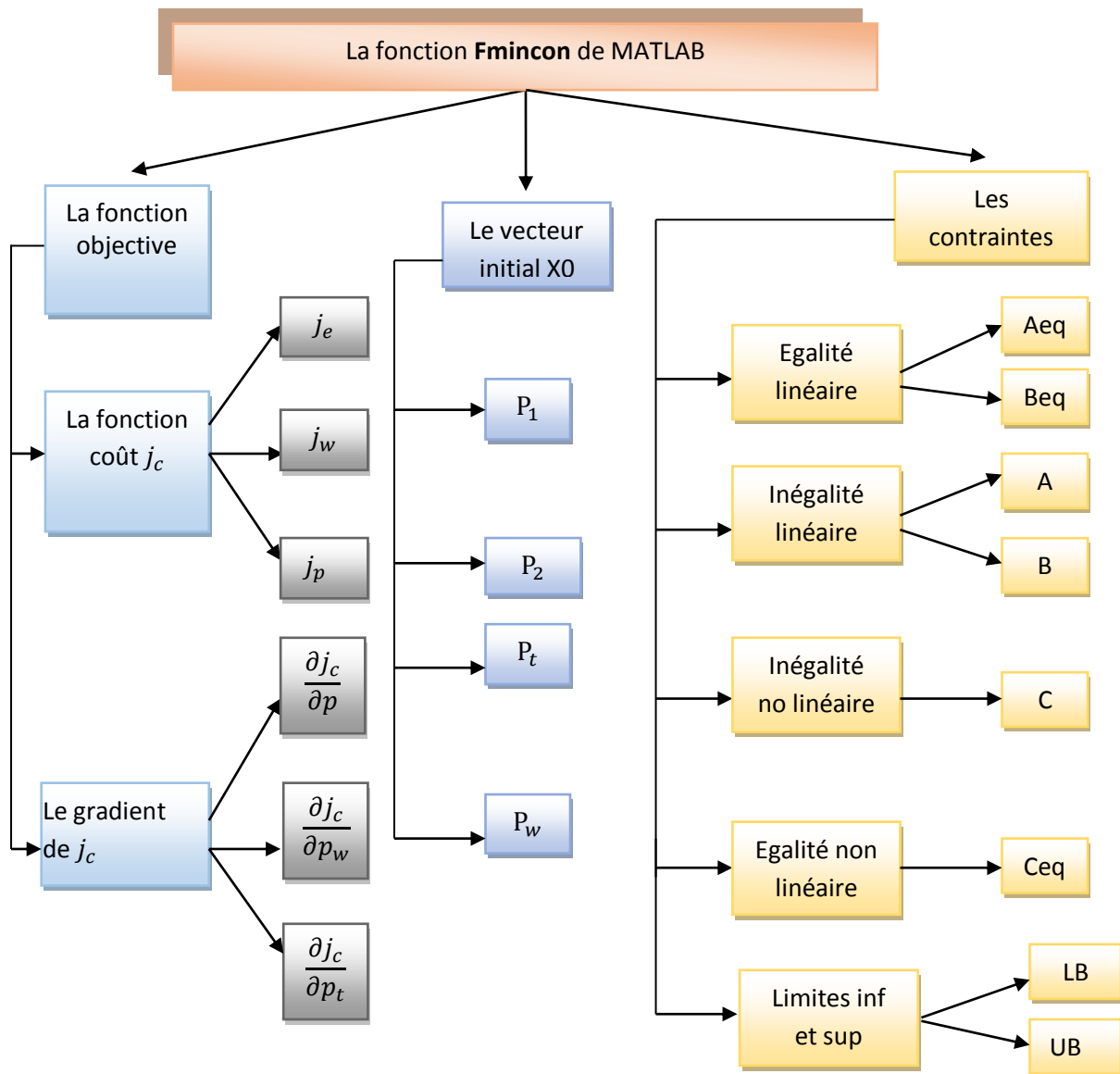


Fig.III.04 : Explication de la fonction fmincon

Détermination des contraintes utilisée dans la fonction d'optimisation FMINCON

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

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

- Les limites sur la position **LB** et **UB** sont déterminée comme suit : $\underline{\theta} \leq p^j \leq \bar{\theta}$ et $(p_t, p_w) >$
- **C** on le définit à partir de $\underline{\tau} \leq \tau(p_t t) \leq \bar{\tau}$;

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

Pour que la fonction d'optimisation prenne en considération le gradient, on utilise l'instruction du Matlab dénommée « **gradobj** », dans la fonction objective.

Calcul des gradients

- Le gradient de couple

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

$$\nabla \Gamma = \left[\frac{\partial \Gamma}{\partial P} \frac{\partial \Gamma}{\partial p_w} \right]^T$$

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

$$\frac{\partial \Gamma_i}{\partial P} = \frac{\partial \Gamma_i}{\partial \theta_i} \frac{\partial \theta_i}{\partial P} + \frac{\partial \Gamma_i}{\partial \dot{\theta}_i} \frac{\partial \dot{\theta}_i}{\partial P} + \frac{\partial \Gamma_i}{\partial \ddot{\theta}_i} \frac{\partial \ddot{\theta}_i}{\partial P} \quad (\text{III.46})$$

La dérivée du couple 1 par rapport à p^j

$$\bullet \quad \frac{\partial \Gamma_1}{\partial P} = \frac{\partial \Gamma_1}{\partial \theta_1} \frac{\partial \theta_1}{\partial P} + \frac{\partial \Gamma_1}{\partial \theta_2} \frac{\partial \theta_2}{\partial P} + \frac{\partial \Gamma_1}{\partial \dot{\theta}_1} \frac{\partial \dot{\theta}_1}{\partial P} + \frac{\partial \Gamma_1}{\partial \dot{\theta}_2} \frac{\partial \dot{\theta}_2}{\partial P} + \frac{\partial \Gamma_1}{\partial \ddot{\theta}_1} \frac{\partial \ddot{\theta}_1}{\partial P} + \frac{\partial \Gamma_1}{\partial \ddot{\theta}_2} \frac{\partial \ddot{\theta}_2}{\partial P} \quad (\text{III.47})$$

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

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

$$\bullet \quad \frac{\partial \Gamma_2}{\partial p^j} = \frac{\partial \Gamma_2}{\partial \theta_1} \frac{\partial \theta_1}{\partial p^j} + \frac{\partial \Gamma_2}{\partial \theta_2} \frac{\partial \theta_2}{\partial p^j} + \frac{\partial \Gamma_2}{\partial \dot{\theta}_1} \frac{\partial \dot{\theta}_1}{\partial p^j} + \frac{\partial \Gamma_2}{\partial \dot{\theta}_2} \frac{\partial \dot{\theta}_2}{\partial p^j} + \frac{\partial \Gamma_2}{\partial \ddot{\theta}_1} \frac{\partial \ddot{\theta}_1}{\partial p^j} + \frac{\partial \Gamma_2}{\partial \ddot{\theta}_2} \frac{\partial \ddot{\theta}_2}{\partial p^j} \quad (\text{III.48})$$

Les dérivées incluses dans ces expressions se calcul directement à partir du modèle géométrique inverse du robot.

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

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

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

Où : $j = [1 \dots m]$ et $K = 5$

D'après ces calculs, on obtient :

$$\frac{\partial \Gamma}{\partial P} = \begin{cases} \frac{\partial \Gamma_1}{\partial P} \\ \frac{\partial \Gamma_2}{\partial P} \end{cases} \quad (\text{III.49})$$

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

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

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

Donc :

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

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

(III.50)

➤ le gradient de J_c :

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

$$\nabla J_c = \left[\frac{\partial J_c}{\partial P} \frac{\partial J_c}{\partial p_w} \frac{\partial J_c}{\partial P_t} \right]^T \quad (\text{III.51})$$

Avec : $J_w = -wp_w$

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

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

Calcul de la dérivée partielle du coût J_c par rapport au poids p_w :

$$\frac{\partial J_c}{\partial p_w} = \frac{\partial J_w}{\partial p_w} + \frac{\partial J_e}{\partial p_w} \quad (\text{III.52})$$

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

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

Avec :

$$\begin{aligned} \frac{\partial J_e}{\partial p_w} &= \frac{\partial}{\partial p_w} \left(\frac{1}{2} \int_0^{t_f} \tau^T W_e \tau dt \right) = \frac{1}{2} \int_0^{t_f} \frac{\partial}{\partial p_w} (\tau^T W_e \tau) dt = \frac{1}{2} \int_0^{t_f} \frac{\partial}{\partial \tau} (\tau^T W_e \tau) \frac{\partial \tau}{\partial p_w} dt \\ &= \frac{1}{2} \int_0^{t_f} (\tau^T W_e) \frac{\partial \tau}{\partial p_w} dt \end{aligned}$$

Pour l'expression de $\frac{\partial \tau}{\partial p_w}$ voir l'équation (III.50)

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

$$\frac{\partial J_c}{\partial P} = \frac{\partial J_w}{\partial P} + \frac{\partial J_e}{\partial P} = \frac{\partial}{\partial P} \left(-wp_w + \frac{1}{2} \int_0^{t_f} \Gamma^T w_e \Gamma dt \right) \quad (\text{III.53})$$

$$\text{Ainsi :} \quad \frac{\partial J_c}{\partial P} = \frac{\partial}{\partial P} (-wp_w) + \frac{\partial}{\partial P} \left(\frac{1}{2} \int_0^{t_f} \Gamma^T w_e \Gamma dt \right) \quad (\text{III.54})$$

Etant donné que $\frac{\partial}{\partial P} (-wp_w) = 0$, alors la dérivée de J_e par rapport à P est comme suit :

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

L'expression de $\frac{\partial \Gamma}{\partial P}$ est donnée dans l'équation (III.49).

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

$$\frac{\partial J_c}{\partial p_t} = \frac{\partial J_w}{\partial p_t} + \frac{\partial J_e}{\partial p_t} = \frac{\partial}{\partial p_t} \left[\left(\frac{1}{2} \int_0^{t_f} \Gamma^T w_e \Gamma dt \right) + (-wp_w) \right] = \frac{1}{2} \Gamma^T w_e \Gamma |_{t=t_f} \frac{\partial t_f}{\partial p_t} + 0$$

$$\text{Ce qui donne : } \frac{\partial J_c}{\partial p_t} = \Gamma^T w_e \Gamma |_{t=t_f} \frac{\partial (\frac{\eta_f}{p_t})}{\partial p_t} + 0 = -\frac{t_f}{p_t} \Gamma^T w_e \Gamma |_{t=t_f} \quad (\text{III.55})$$

III.6 Résultats de simulation

Pour les paramètres du robot, nous avons pris :

$$m_1=14\text{kg}; m_2=7\text{kg}; l_1=0.75\text{m}; l_2=0.5\text{m}; b_1=0.01\text{m}; b_2=0.01\text{m}; g=9.98\text{m/s}^2.$$

Au départ, nous nous sommes appuyés sur les paramètres initiaux suivants calculés précédemment.

$$p_1 = [-42.3966, -42.3966, -50.3238, -78.6374, 85.6095, 65.7490, 49.0775, 32.3917, 18.1351, 0.7405, -36.1101, -40.1070, -40.1070, -40.1070].$$

$$p_2 = [109.4388, 109.4388, 126.1570, 130.5416, 138.5904, 136.0545, 127.1090, 136.0545, 129.0501, 126.8699, 126.8699, 109.4631, 109.4631, 109.4631].$$

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

$$p_t = 10;$$

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

Le cout initial est égal à :

$$J_c \text{initial} = 3.4526 \cdot 10^4$$

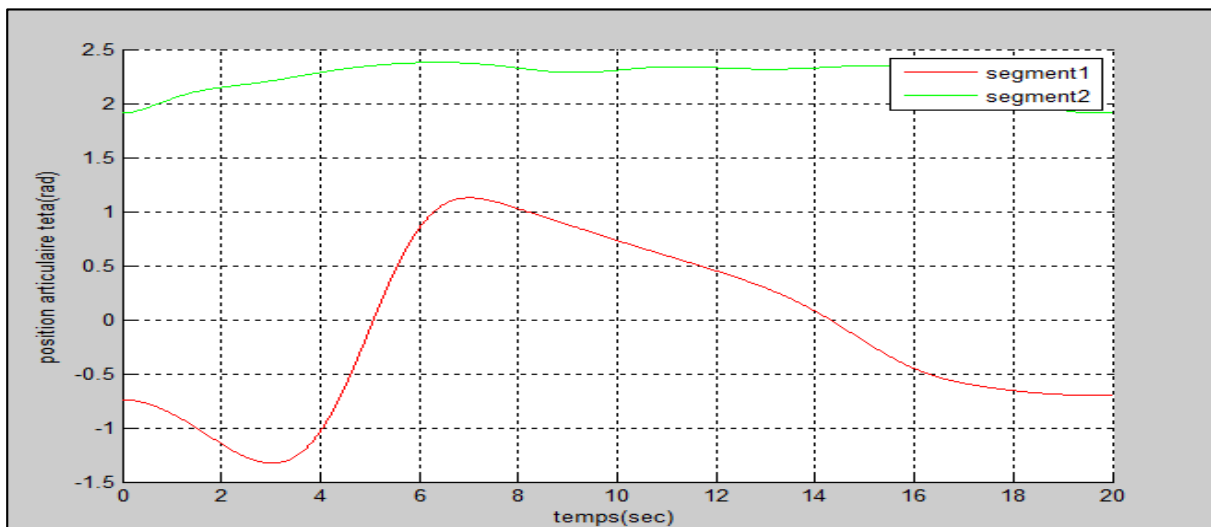


Fig.III.05 : Les positions articulaires θ_1 et θ_2 pour la trajectoire initiale

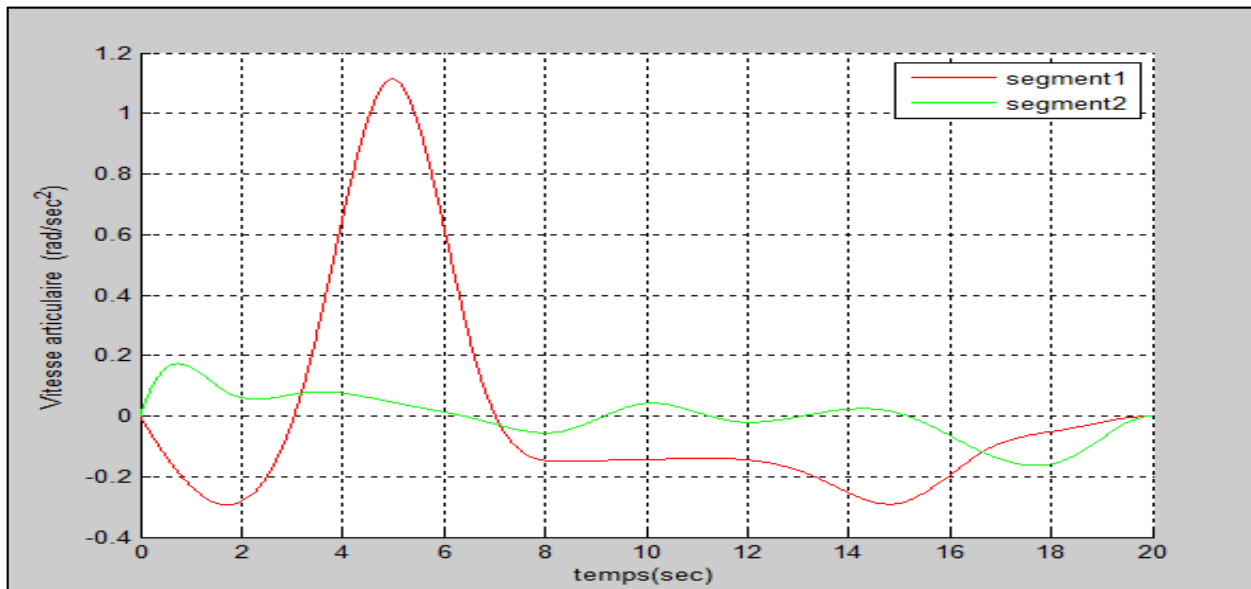


Fig.III.06 : Les vitesses articulaires $\dot{\theta}_1$ et $\dot{\theta}_2$ pour la trajectoire initiale

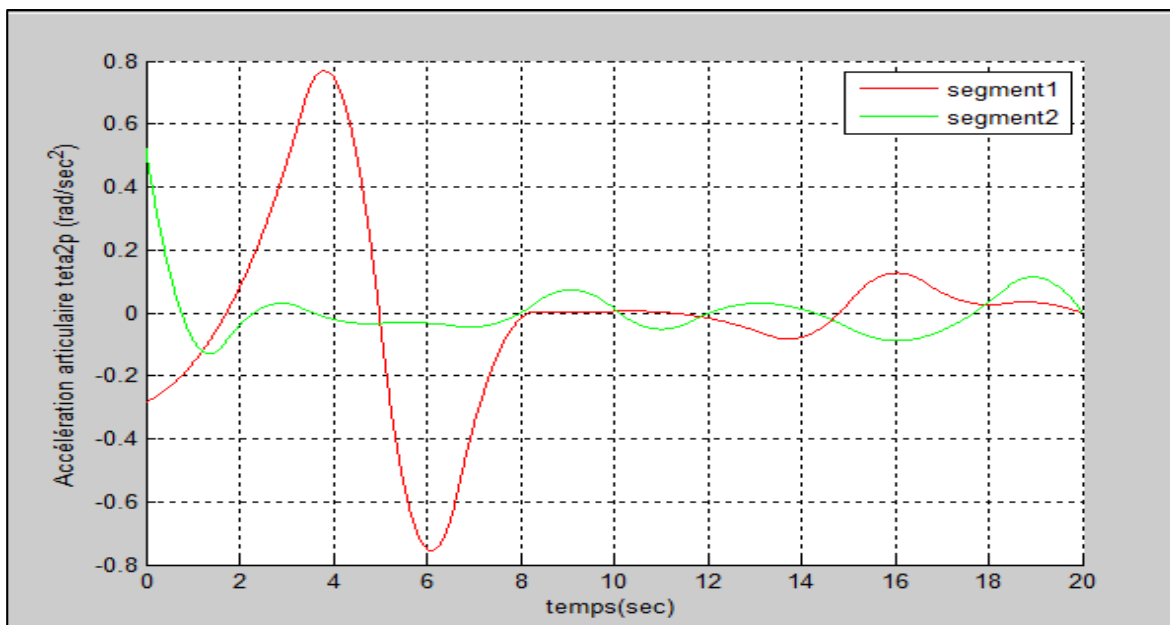


Fig.III.07 : Les accélérations articulaires $\ddot{\theta}_1$ et $\ddot{\theta}_2$ pour la trajectoire initiale

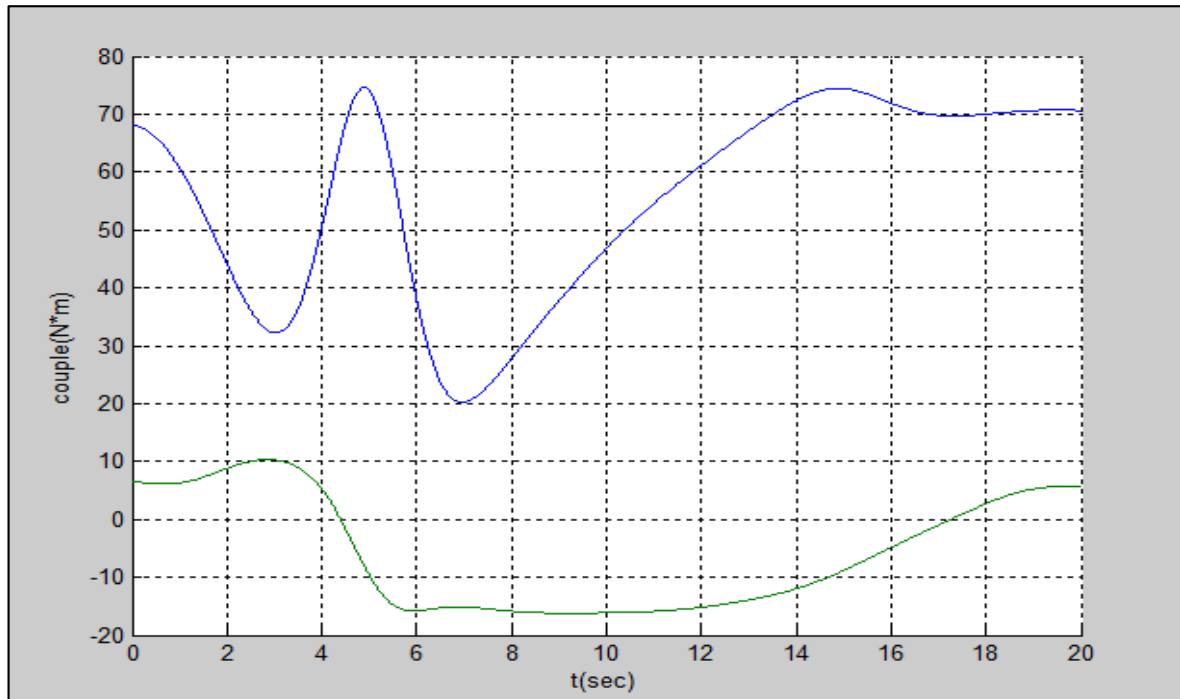


Fig.III.08 : Les couples initiaux

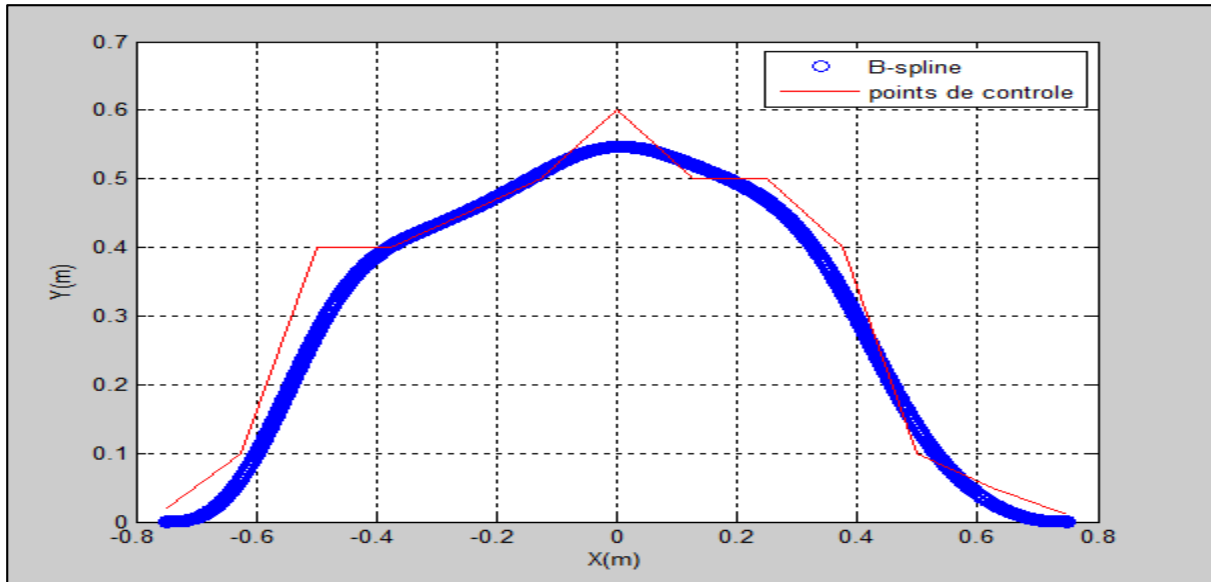


Fig.III.09 Génération de la trajectoire initiale avec deux méthodes

Nous allons exécuter une série d'essais afin de voir de près les résultats

Premier essai

- On prend les contraintes sur les vitesses articulaires :

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

- Les contraintes sur les positions articulaires :

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

- Les contraintes sur les couples :

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

Les résultats de premier essai

On obtient les points de contrôle optimaux suivants :

$$p1 = [-42.4267 \ -42.4267 \ 87.5733 \ 152.5965 \ 102.5647 \ -107.4035 \ -100.7494 \ -123.2813 \\ 66.8765 \ 136.7187 \ 104.8311 \ 80 \ -180 \ -40.1370 \ -40.1370 \ -40.1370].$$

$$p2 = [109.4388 \ 109.4388 \ 180 \ 178.4809 \ 180 \ 161.4191 \ -138.5809 \ -180 \ -120 \\ 180 \ 180 \ 130 \ 116.5643 \ 109.4631 \ 109.4631 \ 109.4631].$$

Le coût optimisé est :

$$J_c = 2.1 \cdot 10^4$$

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

$$p_w = 45.7781$$

Le facteur d'échelle est

$$p_t = 10$$

Discussion des résultats

Après la simulation on remarque que :

- La trajectoire optimale améliore le coût.
- Le poids soulevé par le robot est maximisé.
- Les contraintes sur les positions articulaires et les vitesses sont vérifiées.
- Le couple est limité entre le couple maximum et le couple minimum (pas de surcharge sur les actionneurs).

Deuxième essai

Pour ces essais nous allons changer les contraintes comme suit :

- vitesses articulaires :

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

- positions articulaires :

$$-180 * \frac{\pi}{180} \leq \theta_1 \leq 180 * \frac{\pi}{180} \text{ et } -190 * \frac{\pi}{180} \leq \theta_2 \leq 190 * \frac{\pi}{180},$$

- couples articulaires :

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

Résultats du deuxième essai

On obtient les points de contrôle optimaux suivants :

$$p1=[-42.4273 \quad -42.4273 \quad -150.7606 \quad -70.6212 \quad 146.0455 \quad 57.6367 \quad 118.7903 \quad 53.4607 \\ 153.7216 \quad -62.9451 \quad -150.1161 \quad -18.1315 \quad -179.8130 \quad -40.1374 \quad -40.1374 \quad -40.1374]$$

$$p2=[109.4388 \quad 109.4388 \quad 190.0000 \quad 190.0000 \quad 163.3017 \quad 190.0000 \quad 171.5077 \quad 190.0000 \\ 168.5303 \quad 186.5109 \quad 174.3565 \quad 184.7984 \quad 190.0000 \quad 109.4631 \quad 109.4631 \quad 109.4631]$$

Le poids optimisé est :

$$p_w = 92.3757$$

Le coût optimisé est :

$$J_c = 1.50 \cdot 10^4$$

Le facteur d'échelle est :

$$p_t = 10$$

Troisième essai :

- On prend les limites sur les vitesses articulaires :

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

- Et les limites sur les positions articulaires :

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

- Et les limites sur les couples :

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

Résultats du Troisième essai :

On obtient les points de contrôle suivants :

$$p1 = [-42.4273 \ -42.4273 \ -67.4273 \ -104.9273 \ -112.676 \ -62.1676 \ -12.1676 \ 37.8324 \ 87.8324 \\ 75.2774 \ 90.0000 \ 47.3626 \ -2.6374 \ -40.1374 \ -40.1374 \ -40.1374]$$

$$p2 = [109.4388 \ 109.4388 \ 130.0000 \ 130.0000 \ 130.0000 \ 130.0000 \ 130.0000 \ 130.0000 \ 130.0000 \\ 130.0000 \ 130.0000 \ 130.0000 \ 130.0000 \ 130.0000 \ 109.4631 \ 109.4631 \ 109.4631]$$

Le coût optimisé est :

$$J_c = 1.6 \cdot 10^4 ;$$

Le poids maximal que peut soulever par le robot :

$$p_w = 64 ;$$

Le facteur d'échelle est :

$p_t = 10$;

- La trajectoire optimale améliore le coût ;
- le poids est maximisé ; mais moins maximisé Néanmoins, la diminution de l'intervalle de variation des variables articulaires a pour effet la diminution de ce poids ceci est dû au fait que l'on a moins de possibilités pour éviter les configurations qui nécessitent de plus grands efforts.
- les contraintes sur les positions et les vitesses articulaires sont vérifiées (pas de saturation).
- le couple est limité entre le couple maximum et le couple minimum (pas de surcharge sur les actionneurs) ;

Conclusion :

Après l'exécution des trois essais, nous sommes parvenus à mettre en application des paramètres étudiés dans les chapitres antérieurs et ainsi nous avons prouvé l'efficacité de la fonction d'optimisation utilisée et la fiabilité des résultats obtenus.

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

La deuxième étape était consacrée à la génération de la trajectoire utilisant la méthode du B-spline d'ordre 5.

Dans la troisième étape, nous avons établi analytiquement les relations théoriques nécessaires pour le calcul du gradient de la fonction du coût. Ce gradient joue un rôle majeur dans l'optimisation.

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

Conclusion générale

Dans ce travail, nous nous sommes fixés comme objectif, l'étude de la planification de trajectoires optimales pour un robot manipulateur à 2 degrés de liberté en prenant compte des critères de performance comme le coût et le poids maximal à soulever par le robot.

De là, nous avons poursuivi avec la présentation des points, vecteurs, plans, et matrices de transformation, tâche primordiale pour notre étude. Après cela, vient le sujet de la modélisation des robots, dont nous avons passé en revue les différents modèles indispensables à notre travail. Sur les deux méthodes possibles pour le calcul du modèle dynamique (Newton et Lagrange), notre préférence a tôt fait pour la simulation. Celle de Newton semble mieux convenir aux applications en temps réel.

Les actionneurs ont également fait l'objet d'un bref aperçu, en insistant entre autre sur leurs critères de sélection qui sont d'un concours évident dans l'éclaircissement des choix qui sont généralement faits entre actionneurs électriques, hydrauliques ou pneumatiques.

La génération de la trajectoire se réalise via diverses méthodes, tous dépend de la nature de notre mouvement (point-à-point ou point-à-point avec points intermédiaires). Pour le premier cas les méthodes d'interpolation polynomiale linéaire, cubique ou de degré 5, sont les plus fréquemment utilisées. Mais pour le deuxième cas, on est obligé de passer par des points bien définis (par exemple pour l'évitement d'obstacles) alors on utilise les polynômes définis par morceaux dont le degré est directement lié au nombre de points de passage.

Passées ces étapes, nous avons abordé les différentes méthodes connues par lesquelles on résolvait le problème de la commande optimale. Nous avons trouvé nettement plus avantageux de convertir celui-ci en un problème d'optimisation.

Pour assurer la minimisation du coût par l'optimisation de la trajectoire suivie par notre modèle (robot à 2 ddl), nous avons adopté une démarche qui commence par la supposition d'une trajectoire initiale (mouvement initial aux extrémités bien définies). Partant de ce point, nous avons pu approximer la trajectoire à temps optimal. Enfin, pour venir à une fonction du **Matlab** dénommée **Fmincon**, réputée par sa simplicité et son rendement.

A la fin de notre travail s'est avéré atteint l'objectif fixé dès la problématique : l'obtention d'une trajectoire qui, effectivement, « minimise le cout », qui es fonction du couple (l'effort appliqué par les actionneurs) et aussi fonction de l'inverse du poids (la charge soulevé par le manipulateurs) .alors pour un couple minimum ,la charge soulevée par les actionneurs est maximisée et cela est fait par l'utilisation de la fonction *fmincon* matlab .

Références Bibliographiques

- [1]- **P. COIFFET**, "La robotique, principes et applications", Edition HERMES, 1988.
- [2]. **J-P. Lallemand**, S. Zeghloul , « Robotique ,Aspect fondamentaux, Modélisation mécanique, CAO robotique, commande » , Masson .1994
- [3]. **Y. Koren** . « La Robotique, pour l'ingénieur », Mc Gray-Hill. New York 1986
- [4]. **W.Khalil** ,**E. Dombre**, « Modélisation, Identification et Commande des robots Hermès 2002
- [5]- **AZOUÏ N.**, 'Commande non linéaire d'un bras manipulateur ' Mémoire de magister en électrotechnique option 'Robotique', Université de Batna, 2008/2009.
- [6]- **Frank L. Lewis, Darren M. Dawson, Chaouki T. Abdallah**, "Robot Manipulator Control", second edition, Marcel Dekker, Inc, USA, 2006.
- [7]- **M. CASIN et J. METGE**, " Mécanique de la robotique ", Edition Dunod, 1989.
- [8]- **M.O Demaurex**, "Approche théorique de la conception de la structure mécanique d'un robot industriel", Thèse pour l'obtention du grade de Docteur en Sciences techniques.
- [9] **Fergani Soheib**. « commande d'un robot manipulateur » Séminaire 4ème année, Ecole nationale polytechnique 2009.
- [10]- **Bobrow J.E. DUBOYSKY S., & GIBSON J.S.**, 'Time Optimal of Rbotic Manipulators along specified path' Int.J.Robotics Research, Vol.4,3-17.
- [11]- **J.L.Fanchon**, "Guide des sciences et technologies industrielles", Edition AFNOR-NATHAN, 1994.
- [12] - **Donald E. Kirk**, 'Optimal Control Theory, An Introduction', Dover Publications, Inc.1970.
- [13] - **Minh Tuan TRAN**, 'Thèse En vue de l'obtention du Doctorat de l'université de Toulouse'.2004
- [14] **Bellman R. E. et S. E. Dreyfus**, *Applied Dynamic Programming*, Princeton NJ : Princeton University Press, 1962.
- [15] **Bryson A. et Y. C. Ho**, *Applied Optimal Control*, Hemisphere publishing Corporation, Washington D.C, 1975.
- [16] **Pontryagin L.S., V. G. Boltyanskii, R. V. Gamkrelidze et E. F. Mischenko**, *The Mathematical Theory of Optimal Process*, Wiley, New York, 1962.
- [17] **Lukes D. L.**, Optimal Regulation of Nonlinear Dynamical System, *Automatica*, 1969.
- [18] **Willemstein A. P.**, Optimal Regulation of Nonlinear Dynamical Systems on a Finite Interval, *SIAM J. Control*, 15, 1050-1069, 1977.
- [19] **Garrard W. et J. Jordan**, Design of Nonlinear Automatic Flight Control Systems, *Automatica*, 13, 497-505, 1977.

- [20] **Nishikawa Y., N. Sannomiya et H. Itakura**, A Method for Suboptimal Design of Nonlinear Feedback Systems, *Automatica*, 7, 703-712, 1971.
- [21] **Yoshida T. et K. Loparo**, Quadratic Regulatory Theory for Analytic Nonlinear System with Additive Controls, *Automatica*, 25-4, 531-544, 1989.
- [22] **Troch I., F. Breitenecker et M. Graeff**, Computing Optimal Controls for Systems with State and Control Constraints, *IFAC Control Applications of Nonlinear Programming and Optimization*, France, 39-44, 1989.
- [23] **Stryk O. and R. Bulirsch**, Direct and indirect methods for trajectory optimization, *Annals of Operations Research*, 37, 357-373, 1992.
- [24] **Shien- yu W.**, On the Block pulse operator method to solving system with time delay, *16th Annual Conference of IEEE-IECON 90*, vol.1, 120-125, 1990.
- [25] **Emmanuel Trélat**, 'Commande Optimale', Notes du cours. 2007/2008.
- [26] **ROUX Patrick**, 'Nombres Aléatoires & Optimisation Non-linéaire', mémoire d'ingénieur, école polytechnique fédérale de Lausanne, 2003-2004.
- [28] **Marcel-Yves Bachman**, Hubert Cattin, Paul Epiney, Francois Haerberli et
- **Gérald Jenny**, Méthodes Numériques. Editions du Tricorne, 1992.
- [29] **Gerald Farin, Courbes** et surfaces pour la CGAO. Masson, 1992.
- [30] **R.G. Morvan**, Encyclopédie internationale des sciences et des techniques. Presses de la Cité, 1973
- [31] **Jean-Baptiste Yunes**, Interpolation et Approximation de courbes. 21.01.2002.
- [32] **Vincent Lesage**, Les Courbes de Bezier et les B-Splines. 10.2001. UR ac-lille.fr/~math/classes/analyse/bezierspline/bezier-spline.htm.
- [33] **Documentation du logiciel Matlab.**
- [34] **Samir LAHOUAR** : CONTRIBUTION A LA MODELISATION ET A LA PLANIFICATION DE TRAJECTOIRES DE ROBOTS PARALLELES' Thèse doctorat l'Université de Poitiers.
- [35] **Mark W. Spong and M. Vidyasagar**, ' Robot Dynamics and Control', QuinnWoodbine, USA, 1981.
- [36] **Isma Akli**, 'Elaboration d'une stratégie de coordination de mouvements pour un manipulateur mobile redondant', Thèse de Magister, USTHB, Algérie, 2007.
- [37] **John J. Craig**, 'Introduction to Robotics, Mechanics and Control', 2nd ed, Addison-Wesley Publishing Company, Canada, 1989.

Résumé

Dans ce travail, nous avons traité le problème de planification de trajectoire optimale pour un robot à deux degrés de liberté. Nous avons utilisé le B-spline pour paramétrer la trajectoire.

. Nous avons considéré un critère de coût physique qui compte pour l'effort des actionneurs, la maximisation du poids solvable par le robot et la minimisation de surcharge. La méthode prend aussi en considération d'autres contraintes à savoir les limites sur les variables articulaires et sur les vitesses articulaires ainsi que les dynamiques du robot. Nous avons calculé analytiquement l'expression du gradient de la fonction du coût par rapport aux paramètres. Ce gradient est précieux pour guider l'optimisation et la rendre fiable.

Les résultats de simulation, montrent l'efficacité de la méthode. La trajectoire optimale obtenue satisfait toutes les contraintes et minimise le coût.

Abstract

In this work, we dealt with the problem of planning of optimal trajectory for a robot with two degrees of freedom. We used B-spline to parameterize the trajectory.

We considered a physical criterion of cost which counts for the effort of the actuators, the maximization of the weight by the robot and the minimization of overload. The method also takes into account other constraints with knowing the limits about the articular variables and articular speeds as well as the dynamic ones of the robot. We analytically calculated the expression of the gradient of the function of the cost compared to the parameters. This gradient is invaluable to guide optimization and to make it reliable.

The results of simulation, show the effectiveness of the method. The optimal trajectory obtained satisfied all the constraints and minimizes the cost.