

ECOLE NATIONALE POLYTECHNIQUE D'ALGER

D.E.R de Génie Electrique et Informatique

Département d'Electronique



MEMOIRE

En vue de l'obtention du Diplôme

d'Ingénieur d'Etat en Electronique

THEME

**MODELISATION ET COMMANDES
DE BRAS MANIPULATEURS**

Réalisé par :

M. BENKRID
M. MAMMA

Proposé et Dirigé par :

C. LARBES
M.S. AIT CHEIKH

Promotion : Juin 1997

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE D'ALGER

D.E.R de Génie Electrique et Informatique
Département d'Electronique



MEMOIRE

En vue de l'obtention du Diplôme

d'Ingénieur d'Etat en Electronique

THEME

**MODELISATION ET COMMANDES
DE BRAS MANIPULATEURS**

Réalisé par :

M. BENKRID
M. MAMMA

Proposé et Dirigé par :

C. LARBES
M.S. AIT CHEIKH

Promotion : Juin 1997

Dédicaces

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

A ma très chère mère

A mon très cher père

A mes frères et soeurs

A toute ma famille

A tous mes amis, plus particulièrement Daddy, Amine, Houssam et Salim

Khaled

A ma grande mère

A mes très chers parents pour leur soutien durant toutes mes années d'études

A mes frères et soeurs qui m'ont tant soutenu

A toute ma famille

A tous mes amis, plus particulièrement Houssam, Salim, Samir et Amine

Mahieddine

Remerciements

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Nous remercions nos promoteurs M. C.LARBES et M. M.S.AIT CHEIKH, enseignants à l'E.N.P pour l'aide précieuse qu'ils nous ont prodiguée.

Nous tenons à remercier M. S.BACHIR et M. R.BOUBERTACKH pour le soutien qu'ils nous ont apporté afin de mener à bien ce projet.

Enfin, que toutes les personnes ayant contribué, de près où de loin, à la réalisation de ce projet trouvent ici, l'expression de notre profonde gratitude.

تتم هذه الدراسة بالتمثيل الحركي والتحريك لذراع آلية بالإضافة إلى المشاكل المتعلقة بالتحكم فيها.

تقنيات التحكم المستعملة كلاسيكية من نوع PID تهدف إلى إبراز مشاكل التحكم في أجهزة متعددة المتغيرات، لاختية، متغيرة في الزمن ومتراطة.

كل هذه المحاور تم جمعها في برنامج موحد تحت نظام

. WINDOWS

هذا العمل يملح أن يكون منهجية مكملة للمقرر، كما أنه يستطيع أن يكون أداة لتصميم ذراع آلي في المستقبل.

Résumé :

Cette étude concerne la modélisation cinématique et dynamique des bras manipulateurs, ainsi que les problèmes liés à la commande de ces derniers. Les techniques de commande utilisées sont classiques de type PID permettant de mettre ainsi l'accent sur les problèmes de commande de ces systèmes qui sont non linéaires, multivariables, variables dans le temps et couplés.

Tous ces aspects sont englobés dans un logiciel évolutif sous l'environnement WINDOWS. Cette étude peut servir comme un outil didactique ainsi qu'à une éventuelle conception de bras manipulateurs.

Abstract :

This thesis deals with the kinematic and dynamic modelling of robot arms and with the problem associated with the control of this latter. Classical PID control techniques were used in order to show the different problems associated with this type of systems which are non-linear, multivariable, time varying and coupled.

All these aspects and schemes are developed in an interactive and an evolving software. This work could be used as a support for a course in robotics and eventually in the design stage of robot arms.

SOMMAIRE

Introduction Générale

Chapitre I : Généralités sur les robots

I.1 Définition	I.1
I.2 Structure générale d'un robot	I.3
I.3 Classification des robots	I.5
I.4 Sources d'énergie alimentant un robot	I.6

Chapitre II : Modélisation Cinématique

II.1 Introduction et définitions	II.1
II.2 Modélisation géométrique directe	II.2
II.2.1 Matrice de transformation homogène	II.2
II.2.2 Situation de l'organe terminal	II.5
II.2.2.1 La position	II.6
II.2.2.2 L'orientation	II.6
II.2.3 Présentation de la méthode DH	II.8
II.2.4 Modélisation géométrique directe de quelques bras manipulateurs	II.11
II.2.4.1 Le robot PUMA (2 d.d.l)	II.11
II.2.4.2 Le robot cartésien (T.T.T)	II.13
II.2.4.3 Le robot articulé (R.R.R)	II.15
II.3 Modélisation géométrique inverse	II.17
II.3.1 Introduction	II.17
II.3.2 Présentation de la méthode de transformation inverse	II.17
II.3.3 Pseudo algorithme de la méthode	II.19
II.3.4 Modèle géométrique inverse de quelques bras manipulateurs	II.20

II.3.4.a Le bras PUMA (2 d.d.l)	II.20
II.3.4.b Le bras T.T.T	II.20
II.3.4.c Le bras R.R.R	II.21
II.3.5 Conclusion	II.22

Chapitre III : Génération de trajectoires

III.1 Introduction	III.1
III.2 Considérations générale sur la description et la génération de mouvements	III.1
III.3 Génération de trajectoires dans l'espace des variables articulaires	III.3
III.3.1 Interpolation par des polynômes cubiques	III.3
III.3.2 Interpolation par des polynômes cubiques avec des points intermédiaires imposés	III.5
III.3.3 Polynômes d'interpolation d'ordre élevés	III.8
III.3.4 Fonctions linéaires avec des limites paraboliques	III.8
III.3.5 Fonctions linéaires avec des limites paraboliques pour une courbe passant par des points intermédiaires imposés	III.10
III.3.6 Génération de trajectoires cycloïdales	II.14
III.4 Génération de trajectoires dans l'espace cartésien	III.15
III.4.1 Mouvements linéaire dans l'espace cartésien	II.16
III.4.2 Problèmes géométriques	III.17
III.5 Génération de trajectoires en temps réel	III.17
III.6 Elaboration des trajectoires en utilisant le modèle dynamique du robot	III.20
III.7 Perspectives	III.20

Chapitre IV : Modélisation Dynamique

IV.1 Introduction	IV.1
IV.2 Le formalisme d'Euler-Lagrange	IV.2
IV.2.1 Vitesse d'un point du manipulateur	IV.3
IV.2.2 Energie cinétique d'un segment "i"	IV.4
IV.2.3 Energie potentielle du bras manipulateur	IV.6
IV.2.4 Le "Lagrangien"	IV.7
IV.2.5 Equations dynamiques du bras manipulateur	IV.7
IV.2.6 Simplification du modèle dynamique	IV.11
IV.3 Modèle dynamique du bras articulé	

Chapitre V : Simulation et commande

V.1 Introduction	V.1
V.2 Réponse en boucle ouverte	V.1
V.2.1 Procédure	V.1
V.2.2 Analyse des résultats	V.6
V.3 Implémentation d'une loi de commande classique de type PID	V.7
V.3.1 Introduction	V.7
V.3.2 Commande axe par axe	V.7
V.3.3 Résultats et commentaires	V.10
V.3.4 Implémentation d'une loi de commande de type PID	V.11
V.4 Commande linéarisante	V.14
V.5 Commande linéarisante prédictive	V.21
V.6 Conclusion	V.24

Chapitre VI : Graphismes et Animation

VI.1 Introduction	VI.1
VI.2 Représentation plane de l'espace	VI.2
VI.3 Transformations géométriques	VI.3
VI.4 Elimination des parties cachées	VI.5
VI.5 Algorithme de l'animation	VI.7
VI.6 Conclusion	VI.8

Chapitre VII : Présentation du logiciel

VII.1 Introduction	VII.1
VII.2 Modélisation géométrique	VII.2
VII.3 Modélisation dynamique	VII.4
VII.4 Visualisation graphique	VII.5
VII.5 Algorithme de l'animation	VII.6
VII.6 Commande	VII.7
VII.7 Help	VII.9

Conclusion générale

Annexe

Références bibliographiques

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Avec le besoin pressant d'augmenter la productivité et la distribution de produits finis de qualité constante, l'industrie tourne de plus en plus vers l'automatisation des procédés.

A présent, la plupart des tâches automatisées sont réalisées par des machines spéciales conçues de manière à exécuter une fonction prédéterminée dans un processus industriel.

L'inflexibilité et le coût excessif de ces machines, ont amené les gens à s'intéresser à l'emploi de robots capables d'exécuter une variété de tâches dans un environnement plus flexible et avec un coût de production plus bas.

Etymologiquement, le mot **robot** tire sa racine du Bulgare **robu** qui signifie 'serviteur' et qui a donné naissance, entre autres, au mot russe **robota** qui signifie 'travail' et au Tchèque **robota** qui se traduit par 'travail forcé'. C'est justement l'écrivain Tchèque KARL CAPEC qui a popularisé le terme en 1920, au travers d'une pièce de théâtre intitulée 'Rossum's Universal Robots'.

L'étude et le développement des robots industriels remonte aux années 40, où des programmes de recherche ont commencé aux laboratoires d'OAK RIDGE et d'ARGON NATIONAL pour développer un télémanipulateur en structure maître-esclave. Depuis, les développements des robots n'ont pas cessé de croître, c'est les deux dernières décennies qui ont connu la naissance de robots perceptifs hautement adaptatifs, caractérisés par une interaction constante avec l'environnement et dotés de facultés de décision ainsi que des capacités d'apprentissage. Ils sont capables d'adapter ou de modifier d'eux mêmes leurs comportements pour assurer le succès d'une tâche en dépit des variations des conditions d'exécution.

Aujourd'hui, la robotique est vue comme un ensemble de champs de travail et de recherche sur des disciplines diverses comme le modélisation cinématique et dynamique, la commande, la perception, la vision et l'intelligence artificielle. Les robots trop sophistiqués de science fiction sont écartés des préoccupations de la robotique.

On se propose dans ce travail d'étudier certains aspects classiques de la robotique (englobés dans un logiciel) à savoir : la cinématique (directe et inverse), la dynamique, la poursuite de trajectoires, et de mettre l'accent sur les problèmes liés à la commande des bras manipulateurs.

Tous ces axes seront accompagnés d'une simulation et d'une visualisation animée.

Notre but est donc de réaliser un outil servant à la fois comme complément de cours et de support de base pour les travaux pratiques dans le cadre du module de **Robotique** enseignée au département d'Electronique à l'E.N.P. En outre, il peut servir comme un élément de départ pour une future conception d'un bras manipulateur au département. Le type de robot proposé est un bras articulé à trois degrés de liberté rotatifs, dont toute l'étude le concerne essentiellement.

Notre travail s'articule autour des chapitres suivants :

Premier chapitre :

C'est une vue d'ensemble sur la robotique. On y trouve les définitions de base d'un robot, la structure de base d'un système robotique ainsi que les différentes classes de robots. A l'occasion de ce chapitre introductif, on précisera le vocabulaire utilisé.

Deuxième chapitre :

Il traite la modélisation cinématique avec un outil simple qu'est le calcul matriciel.

Après avoir établi les notions de transformation homogène et le paramétrage de DENAVIT-HARTENBERG, l'accent est mis sur les procédures générales d'obtention des relations entre la situation de l'organe terminal et les variables actionneurs, dans le problème direct et inverse.

Troisième chapitre :

On traitera dans ce chapitre les techniques qui nous permettent de faire passer l'effecteur du bras manipulateur par une trajectoire désirée. On verra dans ce chapitre que ces trajectoires doivent satisfaire certaines conditions. Nous évoquerons les problèmes inhérents à la génération de trajectoires dans l'espace cartésien.

Quatrième chapitre :

Ce chapitre présente en détail la modélisation dynamique en utilisant le formalisme de LAGRANGE-EULER.

Cinquième chapitre :

Dans ce chapitre, on mettra l'accent sur les principes de la commande pratique et les problèmes liés au contrôle d'un système aussi complexe qu'un robot via quelques types de commandes.

Sixième chapitre :

On présentera dans ce chapitre quelques algorithmes graphiques de base. Celles-ci nous ont permis la mise en oeuvre d'un outil de visualisation animée de bras manipulateurs.

Septième chapitre :

On présentera dans ce chapitre le logiciel que nous avons réalisé. Celui-ci (ROBSICOM) englobe toutes les notions évoquées dans ce mémoire.

Enfin, une conclusion générale terminera ce mémoire.

Chapitre I

GENERALITES SUR LES ROBOTS

Chapitre I

Généralités sur les robots

I. 1 Définitions

On trouve une définition du terme « robot » évidemment dans les dictionnaires ou encyclopédies, mais surtout dans les textes des associations de robotique de divers pays, où des groupes de réflexion ont mené leur travail avec deux objectifs principaux :

Tout d'abord, la clarté et l'uniformité du vocabulaire afin que les spécialistes ou les utilisateurs puissent être certains de parler le même langage dans leurs discussions, et ensuite, le souci d'aller vers des pratiques de normalisation nationale et internationale, ce qui oblige à être concret et précis.

- Ainsi, le dictionnaire **Robert** propose comme définition :

« Mécanisme automatique pouvant se substituer à l'homme pour effectuer certaines opérations, et capable d'en modifier de lui même le cycle et d'exercer un certain choix (par la détection photoélectrique, des "cerveaux " électroniques, etc ...) ».

- La **JIRA** (**J**apan **I**ndustrial **R**obot **A**ssociation), quant à elle, met plus l'accent sur la notion d'automorphisme en donnant comme définition :

« Dispositifs versatiles et flexibles offrant des fonctions de placement similaires à celles des membres humains ou dont les fonctions de déplacement sont commandées par ses capteurs et ses moyens de reconnaissance. »

- La **RIA** (**R**obotic **I**ndustries **A**ssociation) aux U.S.A insiste sur l'utilisation en proposant :

« Manipulateur multifonctions reprogrammable conçu pour déplacer des matériaux, des pièces, des outils ou des dispositifs spécialisés au travers des déplacements variables et programmables pour accomplir diverses tâches. »

- L'**AFRI** (**A**ssociation **F**rançaise de **R**obotique **I**ndustrielle) a, de son côté, renoncé à donner une

définition générale, pour se focaliser sur la définition de deux grandes classes de robot (définition enregistrée par l'AFNOR) on y trouve ainsi défini :

Le manipulateur :

« Mécanisme généralement composé d'éléments en série, articulés ou coulissants l'un par rapport à l'autre, dont le but est la saisie et le déplacement d'objets suivant plusieurs degrés de liberté. Il est multifonctionnel et peut être commandé directement par un opérateur humain ou par tout système logique. »

Le robot industriel :

« Manipulateur automatique, asservi en position, reprogrammable, polyvalent, capable de positionner et d'orienter des matériaux, pièces, outils ou dispositifs spécialisés au cours de mouvements variables et programmés pour l'exécution de tâches variées.

Il se présente souvent sous la forme d'un ou plusieurs bras se terminant par un poignet. Son unité de commande utilise notamment, un dispositif de mémoire et éventuellement de perception de l'environnement. 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. »

Il faut insister dans la définition d'un robot sur deux caractéristiques essentielles qui le distinguent de certaines machines automatisées, par exemple les machines à commande numérique.

- La première caractéristique est la **versatilité** ou **flexibilité**. Elle concerne la possibilité, pour un robot, d'exécuter des tâches différentes et/ou d'exécuter une même tâche de plusieurs manières. La plupart des robots modernes comprennent un ordinateur qui détermine les opérations que le robot est capable d'effectuer. En changeant le programme ou les instructions, on modifie les équations.

- La seconde caractéristique semble la plus importante et elle n'est pas pleinement maîtrisée. Elle correspond à ce que l'**exécution de la tâche soit totalement automatisée** indépendamment d'une connaissance « complète et à l'avance » de l'environnement dans lequel la tâche est réalisée. Elle suppose donc que la machine puisse appréhender seule la situation de l'environnement, et établir les conditions à respecter en permanence dans la mise en oeuvre de son travail.

I. 2 Structure générale d'un robot [COI 95]

Un robot manipulateur est l'ensemble formé par :

- Une structure mécanique qui supporte l'organe terminal à situer; c'est la partie la plus visible du robot.
- Des **actionneurs** qui servent à agir sur la structure précédente pour en modifier la configuration. Sous la direction des instructions que leurs fournit le programme de l'ordinateur du robot, les mécanismes de commande ou actionneurs (source d'énergie) produisent les forces régissant la manière dont la machine exécute l'opération désirée.
- Les **capteurs** divers nécessaires à la commande, parmi lesquels on distingue les capteurs proprioceptifs qui permettent la saisie de l'état mécanique du robot et les capteurs extéroceptifs qui indiquent l'état de l'environnement du robot manipulateur.
- Un **système de commande** qui pilote les actionneurs du robot à partir de la définition du mouvement à exécuter transmise par le système décisionnel et des informations délivrées par les capteurs proprioceptifs. C'est le système de contrôle, dans lequel l'ordinateur joue en général le rôle principal.

La programmation est le moyen d'introduire dans la mémoire du dispositif de commande les informations nécessaires au travail du robot. On distingue plusieurs modes de programmation :

Programmation par apprentissage:

Dans ce mode de programmation l'opérateur fait décrire à l'organe terminal du robot le parcours nécessaire pour accomplir le travail demandé et le cycle de travail est exécuté par reproduction fidèle de l'opération réalisée manuellement.

Programmation hors ligne :

Ce type de programmation est effectué soit par définition des trajectoires et des séquences, soit par description de la tâche à accomplir sans définition préalable de la trajectoire. Ces séquences d'opérations sont décrites comme instructions d'un programme écrit dans un langage évolué.

- Un **système décisionnel** qui assure la fonction de raisonnement et élabore le mouvement du robot manipulateur à partir de la définition de la tâche à exécuter transmise par l'opérateur à l'aide du système de **communication** (les fonctions de base d'un système décisionnel sont l'interprétation et la compréhension des messages émis par l'opérateur, la résolution des problèmes posés ainsi que la

gestion de la base de données nécessaire à ce système).

- Un **système de communication** qui gère les messages transmis entre le système décisionnel et l'opérateur.

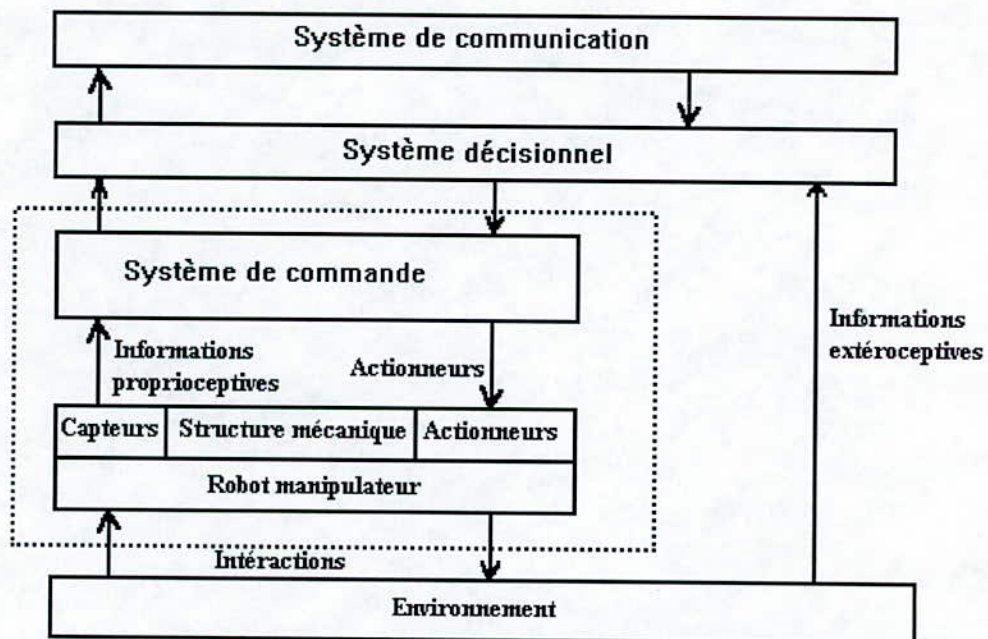


Figure I.1 Structure générale d'un robot manipulateur.

I.3 CLASSIFICATION DES ROBOTS [COI 95] :

On peut classer les robots de plusieurs manières, on peut citer :

* **Classification selon le niveau d'automatisation :**

Dans cette classification les robots sont répartis en 6 classes (selon le développement technologique de leur commande), allant du simple manipulateur manuel jusqu'au robot intelligent.

* **Classification selon leur type d'asservissement :**

En boucle ouverte, ou en boucle fermée.

* **Classification selon le type de l'application :**

Selon la nature de la tâche à réaliser, les robots sont répartis en 3 catégories principales :

- Non asservis de point à point.
- Asservis de point à point.
- A trajectoire continue asservis.

* **Classification morphologique :**

On classe les robots selon leur architecture, cette description donne des indications sur la façon dont les divers segments d'un robot sont organisés et comment ils se déplacent les uns par rapport aux autres.

- Une classification grossière de la morphologie des robots peut résumer ceux-ci à 4 types de machines :

a) Robot cartésien (T T T) :

Robot dont le bras est constitué de 3 articulations prismatiques.

Ce type de structure recouvre environ 15% du parc des robots industriels.

b) Robot a structure cylindrique (R T T) ou (T R T) :

Robot dont le bras est constitué d'une liaison rotative et deux liaisons prismatiques, dont les axes sont disposés selon un système de coordonnées cylindrique. Ce type recouvre environ 45% des robots industriels.

c) Robot polaire ou sphérique (R R T) :

Robot dont le bras est constitué de 2 articulations pivot et une articulation prismatique dont les axes sont disposés selon un système de coordonnées polaires.

Ce robot couvre environ 13% des robots industriels.

d) Robot articulé (R R R) :

Il présente 3 liaisons rotatives et permet donc un mouvement dans différents plans. Ce robot est plus flexible et simple que les autres robots, il présente 25% des robots industriels.

II.4 Sources d'énergie alimentant les robots :

Les robots industriels étant essentiellement à poste fixe, l'énergie qui leur est nécessaire fait presque toujours appel soit à des fluides compressibles (énergie pneumatique) ou non compressibles (énergie hydraulique), soit à l'électricité (énergie électrique). Il n'est pas de même des robots mobiles, qui, étant des véhicules pouvant utiliser les types d'énergie habituellement rencontrés sur ces derniers (énergie thermique, énergie nucléaire etc...). De même, en microrobotique, si les dimensions du robot atteignant le millimètre, c'est l'énergie électrostatique qui devient prépondérante et les moteurs sont directement usinés dans des microblocs de silicium en utilisant les techniques du type VLSI.

Trois types d'actionneurs sont courants dans l'industrie robotique :

a) Actionneurs électriques :

Les actionneurs les plus fréquents utilisent des moteurs électriques à aimant permanent, à courant continu, à commande par l'induit; L'énergie électrique possède plusieurs avantages :

-Utilisation d'une simple prise de courant, classicisme des asservissements, propreté, niveau de bruit réduit, mais atteint ces limites pour des charges élevées.

b) Actionneurs hydrauliques :

Pour des robots devant manipuler de très lourdes charges, les actionneurs sont le plus souvent hydrauliques, agissant en translation (Vérins hydrauliques) ou en rotation (moteurs hydrauliques).

En général, un moteur hydraulique pourra être intégré directement dans une articulation compte tenu de son faible volume. En effet, le couple massique (rapport couple moteur/poids) est intéressant dans les dispositifs hydrauliques. Cependant, les constructeurs et les utilisateurs cherchent à les éviter pour les raisons suivantes : coût élevé, asservissement onéreux, problèmes de maintenance (fuites dans les canalisations).

c) Actionneurs pneumatiques :

On rencontre ce type d'actionneurs sur presque tous les robots séquentiels, car l'air comprimé est un fluide bon marché et disponible dans toutes les usines. Les robots séquentiels sont destinés à faire du transfert de pièces à cadence élevée d'une position à une autre sans possibilité de maîtriser la

trajectoire, en allant d'une butée à une autre, ce qui signifie qu'ils n'ont pas besoin d'asservissement. Et justement l'asservissement pneumatique est un système délicat à réaliser, essentiellement parce que l'air est compressible, ce qui peut engendrer des comportements instables.

Chapitre II

MODELISATION CINEMATIQUE

Chapitre II

Modélisation Cinématique

II.1 Introduction et définitions

La structure mécanique d'un bras manipulateur est constituée généralement par une chaîne de corps solides appelés **segments**, connectés en série par des articulations rotatives ou prismatiques. Généralement un seul degré de liberté est considéré entre deux segments successifs. Une extrémité de cette chaîne est attachée à une base fixe ou mobile, quant à l'autre elle est libre et attachée à un organe terminal pour effectuer une opération spécifiée. La tâche à réaliser pour un robot est d'assurer le déplacement et l'orientation dans l'espace opérationnel de l'effecteur par rapport à un repère fixe de référence.

Chaque configuration géométrique du manipulateur est associée à chaque position et orientation de l'organe terminal et peut être définie par un ensemble de variables dites **articulaires** ou **géométriques**. Une variable articulaire traduit les déplacements relatifs intervenants entre deux segments adjacents sous l'action d'une force extérieure produite par un actionneur.

La cinématique des robots manipulateurs traite les caractéristiques géométriques du mouvement du robot comme une fonction du temps sans s'intéresser aux forces et aux moments qui causent ce mouvement.

Dans ce chapitre, on s'intéresse à la description de la position et de l'orientation des différents segments du manipulateur dans des situations statiques, et en particulier à la relation espace articulaire-espace opérationnel des tâches.

Le modèle géométrique direct d'un robot manipulateur est la fonction **f** qui permet d'exprimer la situation de l'organe terminal du robot en fonction de la configuration du robot lui-même.

La situation de l'organe terminal est définie par des coordonnées que nous appellerons **opérationnelles**:

$$X^T = (X_1, X_2, \dots, X_m)$$

La configuration du robot manipulateur est définie par **n** coordonnées appelées **généralisées** :

$$q^T = (q_1, q_2, \dots, q_n)$$

Le modèle géométrique est donc :

$$X = f(q) \tag{II.1}$$

Le problème inverse peut se poser ainsi ; connaissant la position, l'orientation de l'effecteur et les paramètres géométriques du robot par rapport à un repère de référence, le robot peut-il atteindre cette position ? Et s'il le peut, quelles seraient les différentes configurations possibles ?

Le problème précédent se pose plus fréquemment, car de nombreuses applications industrielles de robots réclament que ceux-ci puissent être commandés et programmés à partir des coordonnées opérationnelles. Il est en effet plus naturel d'exprimer le déplacement absolu de l'élément terminal d'un robot plutôt que de considérer la variation de ces variables articulaires.

Pour décrire la relation de rotation et de translation entre segments, DENAVIT et HARTENBERG, en 1955, proposaient une méthode matricielle qui détermine systématiquement les coordonnées du système pour chaque segment de la chaîne articulée. La représentation D.H détermine une matrice de transformation homogène A_i , (4×4), qui représente les coordonnées de chaque segment du système, ainsi le problème de la cinématique direct se réduit à la détermination d'une matrice de transformation homogène équivalente qui donne la position et l'orientation de l'élément terminal par rapport au repère fixe (R_0) de la base.

Le problème géométrique inverse peut se résoudre par différentes techniques, celles qui sont le plus souvent utilisées sont les méthodes itératives[GOR 84], la méthode analytique[PAU 81] et la méthode géométrique[FU 87].

Dans ce qui suit, nous allons uniquement présenter la méthode analytique. La méthode itérative a été déjà traitée [DJE 96]. Nous allons citer les avantages et les inconvénients de chaque méthode.

II.2 Modélisation géométrique directe

On va tout d'abord présenter la notion de transformation homogène et définir quelques méthodes décrivant la position et l'orientation de l'outil terminal d'un robot, ensuite nous présenterons la méthode de DENAVIT_HARTENBERG.

II.2.1 Matrice de transformation homogène :

Le terme de transformations regroupe un ensemble d'opérations mathématiques relatives aux systèmes de coordonnées. Avant tout développement mathématique, il convient de définir des systèmes d'axes de coordonnées sur lesquels on va travailler. On attache à chaque segment d'indice "i" d'un manipulateur un repère de coordonnées cartésiennes en son extrémité noté $R_i(o_i, x_i, y_i, z_i)$ (figure II.1).

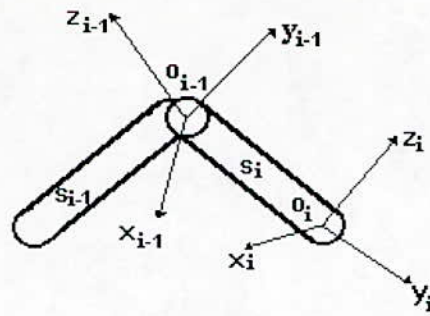


Figure II.1 : Repères associés aux segments d'un manipulateur

Les matrices de transformations définissent les relations géométriques existantes entre deux repères de coordonnées consécutifs $R_{i-1} (O_{i-1}, x_{i-1}, y_{i-1}, z_{i-1})$ et $R_i(O_i, x_i, y_i, z_i)$ en fonction de la variable articulaire $q_i = \theta_i$ s'il s'agit d'un axe de rotation ou $q_i = d_i$ s'il s'agit d'une translation.

Quelle que soit la disposition relative de R_{i-1} et de R_i , on peut les lier par deux transformations élémentaires:

- Un vecteur de translation ${}^{i-1}P_i$ décrivant la position de O_i dans R_{i-1}
- Une matrice de rotation ${}^{i-1}R_i$ (3×3) appelée matrice de passage de R_{i-1} à R_i dont les colonnes sont les composantes des vecteurs de R_i dans R_{i-1} . Ainsi ${}^{i-1}R_i$ s'écrit sous la forme suivante :

$${}^{i-1}R_i = \begin{bmatrix} x_{i-1} \cdot x_i & x_{i-1} \cdot y_i & x_{i-1} \cdot z_i \\ y_{i-1} \cdot x_i & y_{i-1} \cdot y_i & y_{i-1} \cdot z_i \\ z_{i-1} \cdot x_i & z_{i-1} \cdot y_i & z_{i-1} \cdot z_i \end{bmatrix}$$

La matrice de transformation homogène (4×4) contient à la fois la rotation et la translation et est définie par :

$${}^{i-1}T_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}P_i \\ 0 & 1 \end{bmatrix}$$

◆ **Remarques:**

1. En coordonnées homogènes, la représentation d'un vecteur s'obtient en lui ajoutant une 4^{ème} composante de valeur unitaire.
2. L'intérêt de la matrice ${}^{i-1}T_i$ réside dans sa forme compacte, d'où son intérêt algorithmique pour l'obtention du modèle géométrique du bras manipulateur.

Propriétés :

1. Le déterminant de ${}^{i-1}T_i$ est égale à 1
2. L'inverse de ${}^{i-1}T_i$ est ${}^i T_{i-1}$ donnée par [LALL 94] :

$$({}^{i-1}T_i)^{-1} = {}^i T_{i-1} = \begin{bmatrix} {}^i R_{i-1} & {}^i P_{i-1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} ({}^{i-1}R_i)^T & -{}^i R_{i-1} \cdot {}^{i-1}P_i \\ 0 & 1 \end{bmatrix}$$

Quelques transformations homogènes simples

Si on a une rotation de R_i d'un angle (θ_i) autour de l'axe x_{i-1} (figure II.2); alors:

$${}^{i-1}T_i = \text{Rot}(X, \theta_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_i) & -\sin(\theta_i) & 0 \\ 0 & \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.2})$$

De la même manière :

$$\text{Rot}(Y, \theta_i) = \begin{bmatrix} \cos(\theta_i) & 0 & \sin(\theta_i) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_i) & 0 & \cos(\theta_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.3})$$

et

$$\text{Rot}(Z, \theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.4})$$

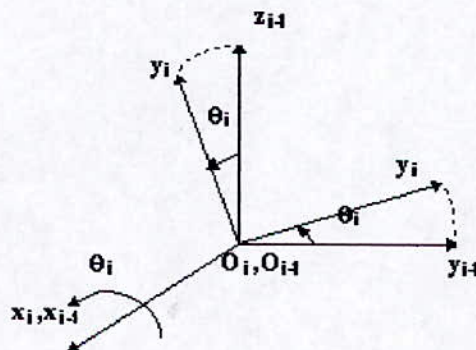


Figure II.2 Rotation du repère R_i autour de l'axe x_i

II.2.2 Situation de l'organe terminal [GOR 84]

La description de l'effecteur d'un bras manipulateur à n degrés de liberté est réalisée par la matrice :

$${}^0T_n = \begin{bmatrix} {}^0R_n & {}^0P_n \\ 0 & 1 \end{bmatrix}$$

liant le repère $R_n (O_n, x_n, y_n, z_n)$ au repère $R_0 (O, x, y, z)$ (figure III.3).

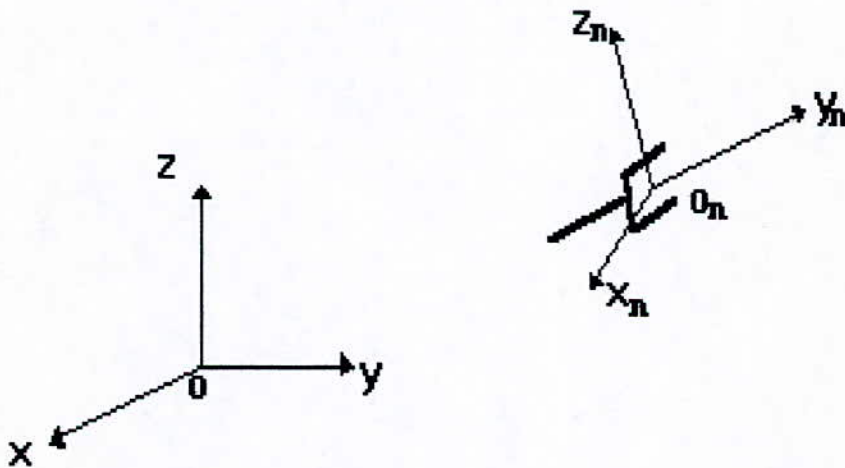


Figure II.3 : Description de l'effecteur d'un manipulateur

Six coordonnées opérationnelles suffisent en général pour définir 0T_n :

- Trois paramètres définissant la position O_n
- Trois paramètres définissant l'orientation (trois angles)

II.2.2.1 La position

La position du repère R_n par rapport au repère R_0 est définie par trois coordonnées de O_n dans R_0 (figure II.4) :

1. Coordonnées cartésiennes : ${}^0P_n = [P_x, P_y, P_z]^T$
2. Coordonnées cylindriques (ρ, θ, z) : ${}^0P_n = [\rho \cos \theta, \rho \sin \theta, z]^T$
3. Coordonnées sphériques (r, θ, ϕ) : ${}^0P_n = [r \cos \Phi \cos \theta, r \cos \Phi \sin \theta, r \sin \Phi]^T$

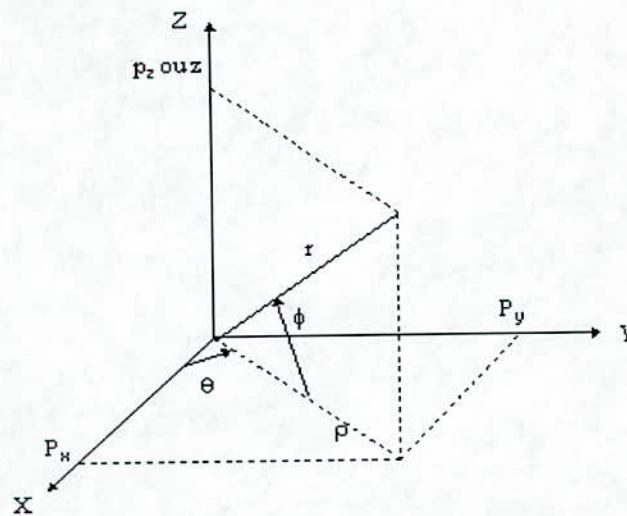


Figure II.4 : Système de coordonnées sphériques.

La structure de base du manipulateur permet de déterminer le type de coordonnées le mieux adapté.

II.2.2.2 L'orientation

L'orientation de R_n par rapport à R_0 peut être définie de diverses manières, nous présenterons les méthodes suivantes car elles sont les plus intéressantes et les plus utilisées.

II.2.2.2.a Angles d'Euler

Le repère R_n se déduit de R_0 par une rotation Ψ autour de l'axe Z, suivi d'une rotation θ autour de l'axe Y, et finalement une rotation Φ encore autour de l'axe Z (figure II.5).

La transformation d'Euler est donc le produit de trois rotations :

$${}^0R_n = \text{Euler}(\Phi, \theta, \Psi) = \text{Rot}(z, \Phi) \cdot \text{Rot}(y, \theta) \cdot \text{Rot}(z, \Psi)$$

$${}^0R_n = \text{Euler}(\Phi, \theta, \Psi) = \begin{bmatrix} C\Phi C\theta C\Psi - S\Phi S\Psi & -C\Phi C\theta S\Psi - S\Phi C\Psi & C\Phi S\theta \\ S\Phi C\theta C\Psi + C\Phi S\Psi & -S\Phi C\theta S\Psi + C\Phi C\Psi & S\Phi S\theta \\ -S\theta C\Psi & S\theta S\Psi & C\theta \end{bmatrix} \quad (II.5)$$

II.2.2.2.b Transformation RPY (Roll,Pitch,Yaw) :

Une autre transformation très utilisée en robotique est la représentation RPY qui correspond à une rotation Ψ autour de X, suivi d'une rotation θ autour de l'axe Y, et enfin une rotation Φ autour de l'axe Z.(figure II.6)

$${}^0R_n = RPY(\Phi, \theta, \Psi) = \text{Rot}(z, \Phi) \cdot \text{Rot}(y, \theta) \cdot \text{Rot}(x, \Psi)$$

$${}^0R_n = \begin{bmatrix} C\Phi C\theta & C\Phi S\theta S\Psi - S\Phi C\Psi & C\Phi S\theta C\Psi + S\Phi S\Psi \\ S\Phi C\theta & S\Phi S\theta S\Psi + C\Phi C\Psi & S\Phi S\theta C\Psi - C\Phi S\Psi \\ -S\theta & C\theta S\Psi & C\theta C\Psi \end{bmatrix} \quad (II.6)$$

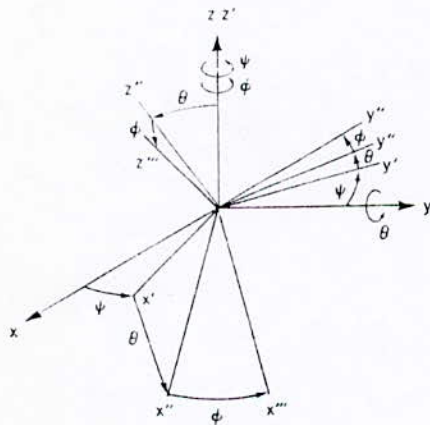


Figure II.5.a Angles d'Euler

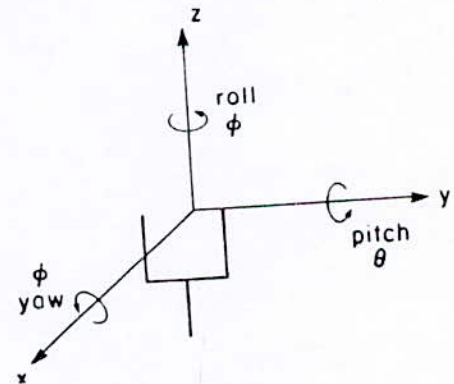


Figure II.5.b Angles RPY

La matrice de transformation homogène 0T_n s'écrit souvent sous la forme suivante (figure II.6)

$${}^0T_n = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (II.7)$$

avec:

\mathbf{p} : vecteur position de l'effecteur , $\mathbf{p} = [p_x \ p_y \ p_z]^T$

\mathbf{n} : vecteur normal, $\mathbf{n} = [n_x \ n_y \ n_z]^T$

\mathbf{o} : vecteur d'orientation , $\mathbf{o} = [o_x \ o_y \ o_z]^T$

\mathbf{a} : vecteur d'approche, $\mathbf{a} = [a_x \ a_y \ a_z]^T$

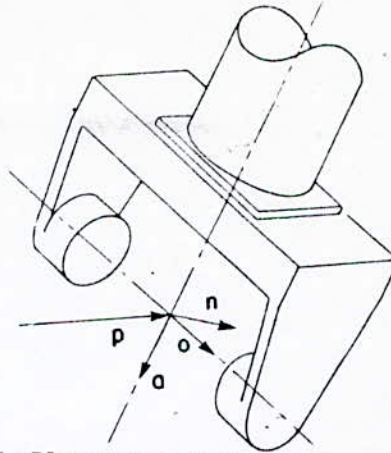


Figure II.6 : Vecteurs n , o , a , p .

II.2.3 Présentation de la méthode D-H [PAU 81] [KOR 84]

La complexité structurelle des systèmes mécaniques articulés rend difficile leur analyse cinématique et dynamiques. L'introduction d'un outil mathématique facilitant cette analyse s'impose. Cette difficulté réside dans la représentation analytique de la position spatiale de l'organe considéré avec un minimum de paramètres.

La position d'une articulation d'un robot manipulateur est déterminée par la connaissance de l'orientation de son axe. Donc, on cherche à représenter une droite dans l'espace avec un minimum de paramètres; C'est ainsi que le problème se pose. La droite (D) représentée par la figure (II.7), qui passe par le point $p(x,y,z)$, nécessite les paramètres β et γ pour compléter sa description [GUE 95].

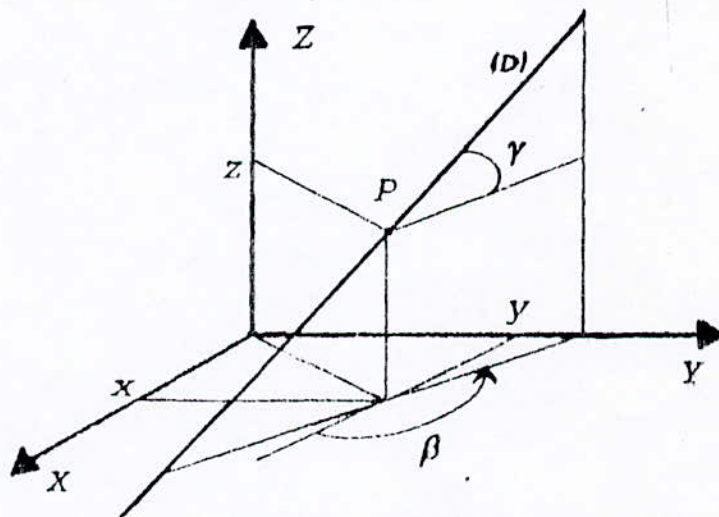


Figure II.7: Représentation de (D) dans l'espace.

Donc les cinq paramètres (x, y, z, β, γ) sont suffisants pour déterminer l'orientation de (D) . Mais la droite peut être définie par ces deux équations:

$$\begin{cases} x = az + b \\ y = cz + d \end{cases}$$

On remarque qu'on a besoin de quatre paramètres (a,b,c,d). La figure (II.8.a) donne une description basée sur la détection de l'unique perpendiculaire (OH') entre (D) et OZ. Cette droite est tangente à l'unique cylindre de rayon a et d'axe OZ. Donc les quatre paramètres sont α , θ , a et d.

Ayant déterminé le nombre de paramètres nécessaires pour la détermination de l'orientation d'une droite (D) dans le repère R (x, y, z), on peut alors élaborer la transformation de DENAVIT-HARTENBERG «T». Cette transformation permet le passage d'un repère $R_{i-1} (o_{i-1}, x_{i-1}, y_{i-1}, z_{i-1})$ à un autre repère $R_i (o_i, x_i, y_i, z_i)$. Celle-ci peut être décomposée en quatre transformations (figure II.8.b).

$${}^{i-1}T_i = \text{Rot}(z_{i-1}, \theta_i) \cdot \text{Trans}(0,0,d_i) \cdot \text{Trans}(a_i,0,0) \cdot \text{Rot}(x_i, \alpha_i)$$

Où :

Rot(z_{i-1}, θ_i) : Rotation autour de z_{i-1} de θ_i

Rot(x_i, α_i) : Rotation autour de x_i de α_i

Trans (0,0,d) : Translation d'une distance d le long z_{i-1}

Trans (a,0,0) : Translation d'une a le long de x_{i-1} .

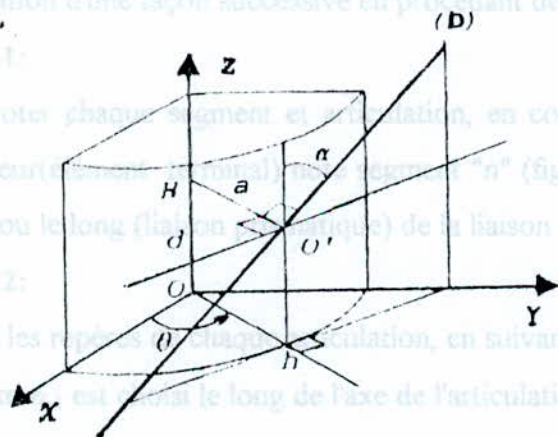


Figure II.8.a

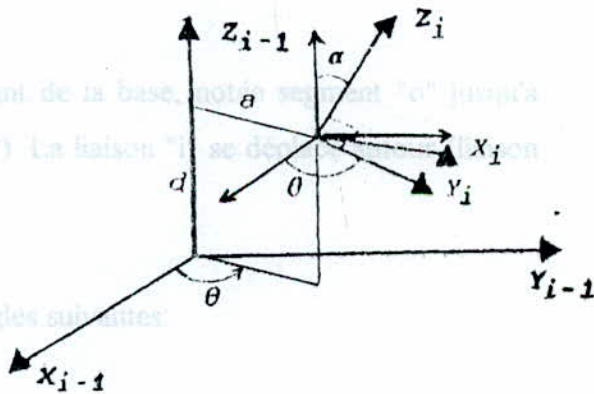


Figure II.8.b

D'où :

1. Deux paramètres de localisation de l'axe z_i dans le segment (S_{i-1}) :

- $\alpha_i = \text{angle}(z_{i-1}, z_i)$ obtenu en vissant z_{i-1} vers z_i autour de x_i .
- a_i : Distance(z_{i-1}, z_i) mesuré le long de l'axe x_i .

2. Deux paramètres de mouvement (rotation et/ou translation) de S_i par rapport à S_{i-1} :

- $\theta_i = \text{angle}(x_{i-1}, x_i)$ obtenu par rotation de x_{i-1} vers x_i autour de z_{i-1} .
- d_i : Valeur de la coordonnée de l'origine o_i sur l'axe z_{i-1} , autrement dit c'est la distance entre o_{i-1} et l'intersection de z_{i-1} avec l'axe x_i .

Etape 4 :

Former la matrice de transformation homogène relative à l'articulation "i" donnée par l'équation (II.8).

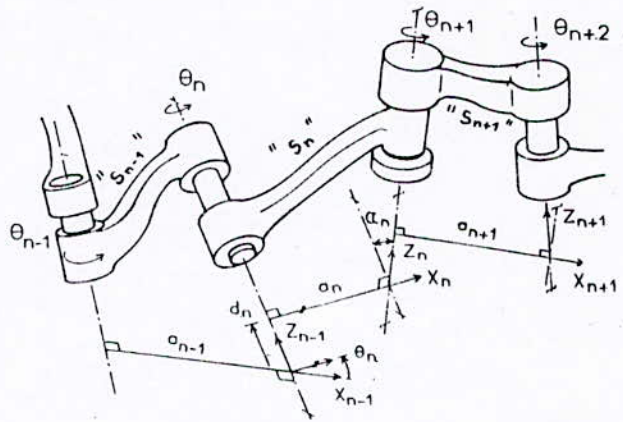


Figure II.9 : Numérotation des segments.

◆ Remarque :

Les matrices de transformations ${}^{i-1}T_i$ ($i=1, \dots, n$) seront notées par la suite A_i .

II.2.4 Modélisation géométrique direct de quelques bras manipulateurs

On applique la matrice de transformation D-H pour les trois robots manipulateurs :

- Bras PUMA à deux degrés de liberté (2 d.d.l).
- Bras cartésien TTT (cartésien).
- Bras articulé à trois degrés de liberté.

On retrouve leurs simulations dans le logiciel final.

II.2.4.1 Le robot PUMA à 2 d.d.l

La figure (II.10) représente un bras du type PUMA à deux degrés de liberté de types rotatifs.

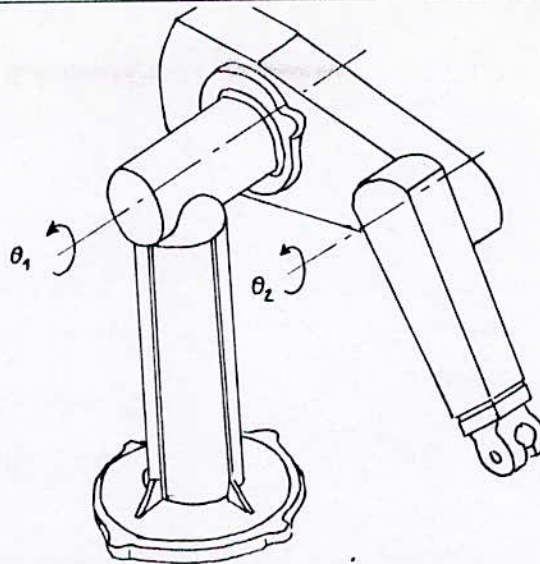


Figure II 10 : Représentation du robot PUMA à 2 d.d.l.

L'implémentation des repères dans chaque liaison en utilisant la procédure D-H est représentée dans la figure (II.11).

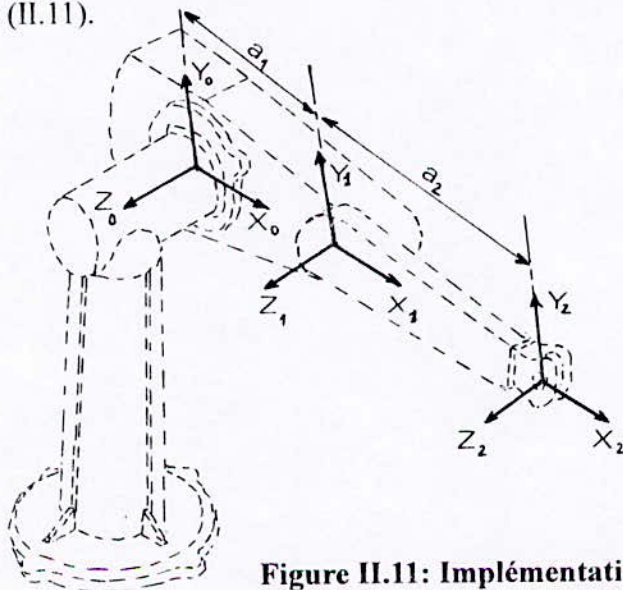


Figure II.11: Implémentation des repères.

Tous les axes de rotation des articulations sont autour de l'axe OZ, cet axe est perpendiculaire au plan de la feuille.

On définit les paramètres suivants :

θ_1, θ_2 : Variables articulaires.

l_1, l_2 : Longueurs des deux segments.

Les paramètres D-H sont donnés dans le tableau suivant :

Segments	Variables	$\theta_{initial}$	α_i	d_i	a_i
Segment 1	θ_1	0	0	0	a_1
Segment 2	θ_2	0	0	0	a_2

Avec $a_1 = l_1$, $a_2 = l_2$.

Les matrices A_1 et A_2 sont données par :

$$A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.9})$$

$$A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.10})$$

La matrice de transformation globale du repère R_0 à R_2 est :

$${}^0T_2 = A_1 A_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_1 C_{12} + a_2 C_1 \\ S_{12} & C_{12} & 0 & a_1 S_{12} + a_2 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.11})$$

II.2.4.2 Le robot cartésien TTT

Les figures (II.12) et (II.13) donnent respectivement la représentation spatiale du robot TTT et l'implémentation de ces liaisons.

Les paramètres D-H sont donnés dans le tableau suivant :

Segments	Variables	$\theta_{i-1,i}$	α_i	d_i	a_i
Segment 1	d_1	0	-90	0	d_1
Segment 2	d_2	90	-90	0	d_2
Segment 3	d_3	-90	0	0	d_3

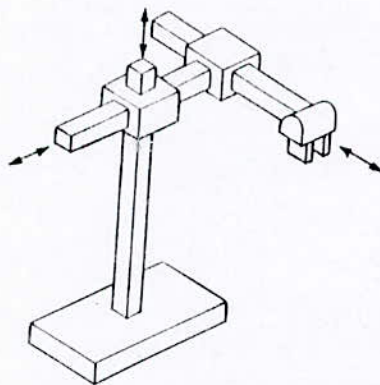


Figure II.12 : La représentation du robot TTT

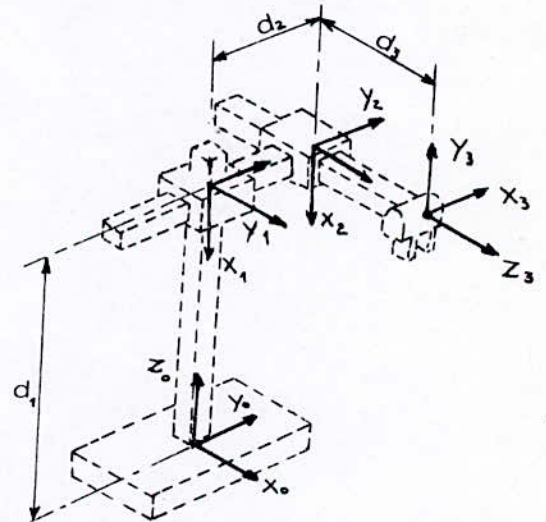


Figure 11.13 : Implémentation des repères

Les différentes matrices de passage, ainsi que la matrice de passage de R_0 à R_3 sont :

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; A_2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}; A_3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ +1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_3 = A_1 A_2 A_3 = \begin{bmatrix} 0 & 0 & -1 & -d_3 \\ -1 & 0 & 0 & d_2 \\ 0 & +1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{II.12}$$

II.2.4.3 Le bras articulé (RRR) :

Le schéma représentatif ainsi que l'implémentation des repères dans chaque liaison sont représentés sur les figures II.14 et II.15 respectivement.

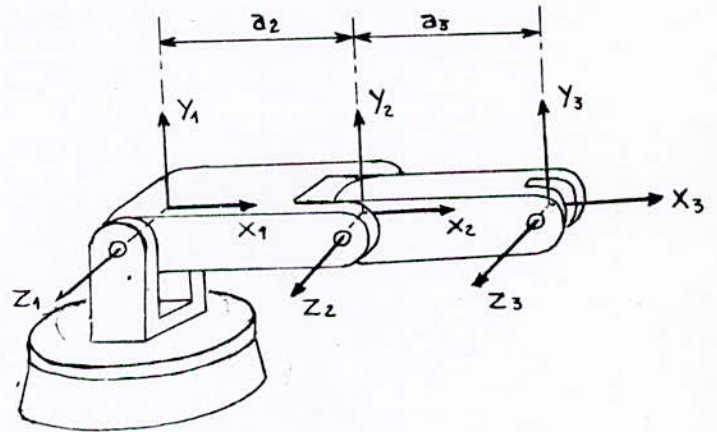
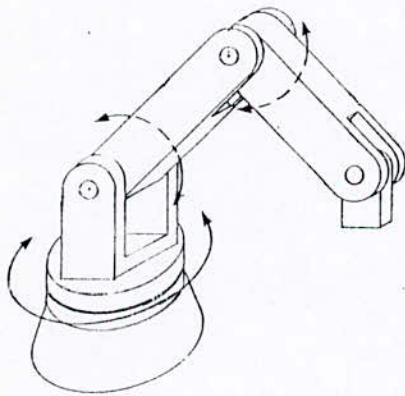


Figure II.14 : La représentation du robot RRR

Figure II.15 : Implémentation des repères

On construit alors le tableau D-H :

Segments	Variables	θ_{mi}	α_i	d_i	a_i
Segment 1	θ_1	0	90	d_1	0
Segment 2	θ_2	0	0	0	a_2
Segment 3	θ_3	0	0	0	a_3

Les matrices de transformation homogènes entre chaque paire de repères de la structure données (ils seront nécessaires pour la cinématique inverse ainsi que la modélisation dynamique du bras.

$$A_1 = {}^0T_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{II.13}$$

$$A_2 = {}^1T_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.14})$$

$$A_3 = {}^2T_3 = \begin{bmatrix} C_3 & -S_3 & 0 & a_3 C_3 \\ S_3 & C_3 & 0 & a_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.15})$$

$${}^1T_3 = A_2 A_3 = \begin{bmatrix} C_{23} & -S_{23} & 0 & a_3 C_{23} + a_2 C_2 \\ S_{23} & C_{23} & 0 & a_3 S_{23} + a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.16})$$

$${}^0T_2 = A_1 A_2 = \begin{bmatrix} C_1 C_2 & -C_1 S_2 & 0 & a_2 C_1 C_2 \\ S_1 C_2 & -S_1 S_2 & -C_1 & a_2 S_1 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.17})$$

$$T_3 = A_1 A_2 A_3 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1 (a_3 C_{23} + a_2 C_2) \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & S_1 (a_3 C_{23} + a_2 C_2) \\ S_{23} & C_{23} & 0 & a_2 S_{23} + a_2 S_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.18})$$

II.3 Modélisation géométrique inverse [PAU 81] [FU 87] :

II.3.1 Introduction:

La commande d'un bras manipulateur est généralement effectuée dans l'espace articulaire, par contre la manipulation des objets et la réalisation d'une tâche sont effectués dans l'espace opérationnel; donc le contrôle de la position et l'orientation de l'effecteur nécessite la connaissance du modèle géométrique inverse.

Le problème géométrique direct est toujours soluble, on peut en effet toujours déduire une position géométrique à partir d'un vecteur de commande. Par contre, le problème inverse n'est pas toujours soluble. En effet, résoudre ce problème consiste à calculer une fonction réciproque de \mathbf{f} , équation (II.1), si elle existe. La difficulté de cette résolution réside dans le fait que la fonction \mathbf{f} est non linéaire; les théorèmes d'existence et d'unicité font alors défaut. Cela signifie physiquement que l'on ne peut pas atteindre n'importe quelle position (espace de travail) sous une position quelconque avec un bras donné.

En général, le problème géométrique inverse peut se résoudre par plusieurs méthodes; on peut citer la méthode analytique, les méthodes itératives ou par des approches géométriques. Dans ce qui suit, nous allons présenter une méthode analytique introduite par R.P.PAUL [PAU 81], et donner ensuite ces avantages et inconvénients relativement aux autres méthodes.

II.3.2 Présentation de la méthode de la transformation inverse :

Cette méthode se base sur l'utilisation des matrices de transformations homogènes directes et inverses. La solution est obtenue d'une manière séquentielle par isolation de chaque variable articulaire à chaque étape de calcul.

Pour un bras manipulateur à "n" degrés de liberté, la matrice 0T_n est connue (donnée sous la forme de l'équation (II.7)) et on cherche les "n" variables articulaires q_i ($i=1,2,\dots,n$ avec $n \leq 6$).

0T_n s'écrit sous la forme:

$${}^0T_n = A_1(q_1)A_2(q_2)A_3(q_3)\dots\dots A_n(q_n) \quad (\text{II.19})$$

On obtient alors "n-1" équations matricielles en multipliant successivement cette équation par les matrices inverse $(A_i)^{-1}$:

$$\begin{cases} (A_1)^{-1} {}^0T_n = {}^1T_n \\ (A_2)^{-1} (A_1)^{-1} {}^0T_n = {}^2T_n \\ \vdots \\ (A_i)^{-1} (A_{i-1})^{-1} \dots {}^0T_n = {}^iT_n \\ \vdots \\ (A_{n-1})^{-1} (A_{n-2})^{-1} \dots {}^0T_n = {}^{n-1}T_n \end{cases} \quad (\text{II.20})$$

Le coté gauche de l'équation (II.20; $i=1,2,\dots,n-1$) est fonction des premières "i-1" variables articulaires; le coté droit contient alors soit des termes nuls, constants ou des termes fonctions des variables articulaires restantes (q_i, q_{i+1}, \dots, q_n).

On obtient donc 12 équations pour chaque équation matricielle (II.19), soit une équation pour chaque élément des vecteurs $\mathbf{n}, \mathbf{o}, \mathbf{a}$ et \mathbf{p} de 0T_n . Ces équations sont généralement données sous 4 formes [PAU 81]:

1) Première forme : $\alpha \cos(q_i) - \beta \sin(q_i) = 0$,

α et β étant des constantes alors que q_i est une variable articulaire rotative

Deux solutions sont possibles :

$$q_i = \text{atan2}(\alpha, \beta) \text{ et } q_i = q_i + 180^\circ.$$

Si α et β sont nulles, l'angle q_i est indéfinie; le manipulateur est dans une position singulière et il perd un degré de liberté.

2) Deuxième forme : $-\sin(q_i)\alpha + \cos(q_i)\beta = \gamma$

Une solution n'existe que si $\alpha^2 + \beta^2 \geq \gamma^2$. On trouve deux solutions :

$$q_i = \text{atan2}(\beta, \alpha) - \text{atan2}(\gamma, \pm \sqrt{\alpha^2 + \beta^2 - \gamma^2}).$$

On a encore ici une possibilité de singularité.

3) Troisième forme :

$$\begin{cases} d_j \cdot \sin(q_i) = \alpha \\ d_j \cdot \cos(q_i) = \beta \end{cases}$$

Avec q_i : Variable articulaire rotative.

d_j : Variable articulaire prismatique, ($i \neq j$), avec $d_j \geq 0$.

La solution unique est :

$$q_i = \text{atan2}(\alpha, \beta)$$

Un cas d'indétermination se posera si $d_j = 0$.

4) Quatrième forme:

$$\begin{cases} \alpha \cos(q_i + q_j) + \beta \cos(q_i) = x_1 \\ \alpha \sin(q_i + q_j) + \beta \sin(q_i) = x_2 \end{cases}$$

Ce cas se pose généralement pour deux liaisons pivots successives d'axes parallèles.

On obtient deux solutions pour q_j :

$$q_i = \text{atan2}(S_j, C_j) \text{ avec } \begin{cases} C_j = \frac{x_1^2 + x_2^2 - \alpha^2 - \beta^2}{2\alpha\beta} \\ S_j = \pm \sqrt{1 - C_j^2} \end{cases}$$

Une solution pour q_i :

$$q_i = \text{atan2}(x_2(\alpha C_j + \beta) - \alpha x_1 S_j, x_1(\alpha C_j + \beta) + \alpha x_2 S_j)$$

◆ Remarques

1. Pour l'obtention des résultats, on évite d'utiliser les fonctions trigonométriques inverses 'arcsin' et 'arccos', car ces fonctions ne sont pas bijectives sur le domaine $[-\pi, \pi]$. Ainsi, la solution n'est pas bien connue.
2. On évite aussi les expressions fractionnaires dont le dénominateur peut s'annuler ou approcher zéro conduisant à une solution indéfinie ou non précise respectivement.

II.3.3 Pseudo algorithme de la méthode

Pour un bras à "n" degrés de liberté, le modèle géométrique inverse s'obtient en suivant les étapes suivantes:

Pour $i=1, \dots, n$ faire

- 1) Lecture de la situation de l'organe terminal (matrice 0T_n).
- 2) Evaluer la matrice de passage homogène correspondante à la situation désirée :
 ${}^0T_n = A_1(q_1)A_2(q_2)A_3(q_3)\dots\dots A_n(q_n)$ fonction des q_i ($i=1, \dots, n$) variables articulaires.
- 3) Chercher dans le coté droit de (II.20), est-il possible de trouver une solution en respectant la remarque précédente?
- 4) Multiplier l'équation (II.19) par $(A_i)^{-1}$, et chercher les termes constants ou nuls dans le second membre de l'équation matricielle obtenue. Si c'est le cas passer à 5, sinon répéter 4 avec $i=i+1$.
- 5) Résoudre les équations obtenues (selon l'une des 4 formes précédentes) et extraire q_i . $i=i+1$, revenir à 4.

II.3.4 Modèle géométrique inverse de quelques bras manipulateurs:

On applique la méthode de transformation inverse pour trois bras manipulateurs :

- Le robot PUMA à deux degrés de liberté.
- Le robot cartésien TTT.
- Le bras manipulateur RRR.

II.3.4.a Bras PUMA (2 d.d.l):

La matrice de transformation de ce bras obtenue en géométrie directe est :

$${}^0T_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_1 C_{12} + a_2 C_1 \\ S_{12} & C_{12} & 0 & a_1 S_{12} + a_2 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Egalant (II.17) à cette matrice, nous obtenons le système suivant :

$$\begin{cases} a_1 C_{12} + a_2 C_1 = P_x \\ a_1 S_{12} + a_2 S_1 = P_y \end{cases}$$

C'est une équation de la quatrième forme, deux couples de solutions (θ_1, θ_2) sont obtenus :

$$\begin{cases} \theta_1 = a \tan 2 \left(P_y (a_1 C_2 + a_2) - a_1 P_x S_2, P_x (a_1 C_2 + a_2) + a_1 P_y S_2 \right) \\ \theta_2 = a \tan 2(S_2, C_2) \end{cases} \text{ avec } \begin{cases} C_2 = \frac{P_x^2 + P_y^2 - a_1^2 - a_2^2}{2a_1 a_2} \\ S_2 = \pm \sqrt{1 - C_2^2} \end{cases}$$

II.3.4.b Le bras TTT

La matrice de transformation globale est : ${}^0T_3 = \begin{bmatrix} 0 & 0 & -1 & -d_3 \\ -1 & 0 & 0 & d_2 \\ 0 & +1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Le modèle inverse ne nécessite aucun calcul. En effet, on obtient directement $\begin{cases} d_1 = P_z \\ d_2 = P_y \\ d_3 = -P_x \end{cases}$

II.3.4.c Le bras RRR :

On rappelle que la matrice de passage 0T_3 est donnée par :

$${}^0T_3 = A_1 A_2 A_3 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1 (a_3 C_{23} + a_2 C_2) \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & S_1 (a_3 C_{23} + a_2 C_2) \\ S_{23} & C_{23} & 0 & a_2 S_{23} + a_2 S_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

On voit qu'on peut isoler S_1 et C_1 , on a :

$$\begin{cases} S_1 = a_x \\ -C_1 = a_y \end{cases} \quad \text{D'où } \theta_1 = \text{atan2}(a_x, -a_y) \text{ ou } \theta_1 = \theta_1 + 180^\circ.$$

Cette solution est fonction du vecteur d'approche \mathbf{a} et non de la position.

Le passage à la seconde étape donne :

$$A_1^{-1} {}^0T_3 = A_2 A_3$$

$$\text{Donc } \begin{bmatrix} f_{11}(n) & f_{11}(o) & f_{11}(a) & f_{11}(p) \\ f_{12}(n) & f_{12}(o) & f_{12}(a) & f_{12}(p) - d_1 \\ f_{13}(n) & f_{13}(o) & f_{13}(a) & f_{13}(p) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_{23} & -S_{23} & 0 & a_3 C_{23} + a_2 C_2 \\ S_{23} & C_{23} & 0 & a_3 S_{23} + a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Avec } \begin{cases} f_{11} = .x C_1 + .y S_1 \\ f_{12} = .z \\ f_{13} = .x S_1 - .y C_1 \end{cases}$$

On voit qu'en égalant les éléments 3.1 des deux matrices, on aura :

$f_{13}(p) = 0$, ce qui conduit à $S_1 p_x - C_1 p_y = 0$ d'où $\theta_1 = \text{atan2}(p_y, p_x)$ ou $\theta_1 = \theta_1 + 180^\circ$.

Une fois θ_1 calculé, le terme en gauche de l'équation précédente est connu.

Les éléments (1,3) et (2,3) de chaque matrice donnent :

$$\begin{cases} a_3 C_{23} + a_2 C_2 = C_1 P_x + S_1 P_y \\ a_3 S_{23} + a_2 S_2 = P_z - d_1 \end{cases}$$

$$\text{Si on pose : } \begin{cases} x_1 = C_1 P_x + S_1 P_y \\ x_2 = P_z - d_1 \end{cases}$$

On se retrouve avec une équation de la forme 4 dont les solutions sont :

$$\begin{cases} \theta_2 = a \tan 2(x_2 (a_3 C_3 + a_2) - a_3 x_1 S_3, x_1 (a_3 C_3 + a_2) + a_3 x_2 S_3) \\ \theta_3 = a \tan 2(C_3, S_3) \end{cases}$$

$$\text{Avec } \begin{cases} C_3 = \frac{x_1^2 + x_2^2 - a_2^2 - a_3^2}{2a_2a_3} \\ S_3 = \pm \sqrt{1 - C_3^2} \end{cases}$$

On a donc quatre triplets de solutions $(\theta_1, \theta_2, \theta_3)$

◆ **Remarque:**

Pour le calcul du modèle inverse, on n'a considéré que les coordonnées opérationnelles de position (P_x, P_y, P_z) car les manipulateurs sont à trois degrés de liberté (sauf pour le PUMA).

L'extension aux bras à six degrés de liberté reste possible si le poignet est sphérique à trois axes concourants, car on peut toujours séparer dans ce cas la résolution du porteur (3 d.d.l) de la résolution du poignet (par les angles d'EULER ou RPY)[LALL 94].

II.3.4 Conclusions :

La méthode analytique (la transformation inverse) est une méthode trigonométrique, elle fait appel aux transformations homogènes, et elle peut s'appliquer sur la plupart des manipulateurs industriels pour lesquels le poignet est sphérique (à trois liaisons rotoïdes concourantes) [FU 87].

Cette méthode analytique nous permet d'obtenir tous les modèles géométriques inverses sous forme littérale. Cependant, elle souffre du fait que le résultat ne donne pas une indication claire sur le choix d'une solution appropriée parmi toutes les solutions possibles pour une configuration particulière du bras; En outre, il n'y a pas d'algorithme unifié applicable à tous les bras manipulateurs possibles, et beaucoup d'intuition géométrique sera réclamée pour l'obtention de la solution. Mis à part ces inconvénients, il est possible de calculer numériquement par un très petit nombre d'opérations, les solutions du problème géométrique inverse. Ces opérations peuvent être facilement intégrés dans un système de commande en temps réel.

Pour les méthodes itératives, on peut citer : La méthode de linéarisation de la matrice de passage et la méthode utilisant la matrice jacobienne du robot manipulateur. Ces méthodes présentent trois inconvénients communs :

- Il est nécessaire de se fixer une estimation initiale de la solution (q_{initiale}) correspondant à la position désirée de l'organe terminal, en plus elles ne garantissent pas une convergence vers la solution exacte.
- La résolution numérique ne permet la connaissance que d'une seule solution.

- Le temps de calcul important peut être une source de difficultés si ces méthodes sont utilisées dans un contexte temps réel.

De plus comme pour la méthode analytique, ces méthodes ne donnent pas un sens géométrique aux solutions trouvées.

Chapitre III

GENERATION DES TRAJECTOIRES

Chapitre III

Génération de trajectoires

III.1 Introduction :

On s'intéresse dans ce chapitre aux méthodes de calculs de trajectoires dans l'espace tridimensionnel décrivant un mouvement désiré du bras manipulateur.

Le terme *trajectoire* veut dire ici une certaine dépendance temporelle de la position, vitesse et accélération et ce pour chaque degré de liberté.

Afin de simplifier la tâche de description du mouvement du bras manipulateur pour l'utilisateur humain, on ne doit pas exiger à ce dernier l'écriture exhaustive des équations de mouvement. On doit plutôt lui donner la possibilité de spécifier les trajectoires par une description simple et laisser au système robot le soin de la description des détails; L'utilisateur pourra à titre d'exemple spécifier la position et l'orientation désirée de l'outil terminal et laisser le système décider de la forme exacte de la trajectoire, de la durée du mouvement, du profil de la vitesse et d'autres détails.

III.2 Considérations générales sur la description et la génération de mouvements:

Dans ce qui suit, on va considérer les mouvements du bras manipulateur comme des mouvements du repère de l'outil terminal **OUTIL** par rapport à celui de la base **BASE**.

Ceci présente les avantages suivants:

A- Découpler la description du mouvement de n'importe quel autre robot, outil terminal ou pièces de travail.

Ceci permettra une certaine modularité du fait que la même description d'une trajectoire donnée pourrait être utilisée pour un autre bras manipulateur ou pour le même bras avec un autre outil terminal.

b- Possibilité de spécifier des mouvements relatifs à une base mobile (un convoyeur par exemple) en spécifiant les mouvements par rapport au repère de la base .

En temps réel, ceci fera que la définition du repère de la base soit dépendante du temps.

Le problème essentiel est de déplacer l'extrémité du bras manipulateur d'une position initiale à une autre finale. En d'autres termes, déplacer le repère de l'outil terminal d'une valeur initiale $\mathbf{OUTIL}_{\text{init}}$ à une valeur finale désirée $\mathbf{OUTIL}_{\text{fin}}$.

Notons que ce mouvement entraînera en général un changement d'orientation aussi bien qu'un changement de position.

Parfois, il nous faut spécifier le mouvement avec plus de précision; On pourra spécifier une séquence de points intermédiaires entre les deux points initiale et finale, c'est à dire que l'outil terminal doit passer par ces points pendant son mouvement.

A noter que quoiqu'on utilise le terme *point*, ce sont en fait des repères \mathbf{OUTIL} intermédiaires qui donnent la position et l'orientation de l'outil terminal.

En plus de ces contraintes *spatiales*, l'utilisateur pourra aussi vouloir spécifier des contraintes temporelles, par exemple le temps qui s'écoule entre deux positions intermédiaires.

Pratiquement, on désire que le mouvement du bras manipulateur soit *lisse*. Dans notre cas, on définira une fonction *lisse* comme une fonction *continûment dérivable*. Dans d'autres cas une continuité d'ordres deux est aussi désirable.

Des mouvements *secoués* constituent une charge supplémentaire au mécanisme et causent des vibrations en excitant les résonances du bras manipulateur.

Dans le but de garantir des mouvements lisses, on doit poser certaines contraintes sur les qualités temporelles et spatiales de la trajectoire entre les points intermédiaires.

A ce stade ils existent divers choix possibles. En effet, chaque fonction lisse qui passe par les points intermédiaires pourrait être utilisée pour spécifier l'allure exacte de la trajectoire.

Dans ce chapitre, on va discuter deux choix fréquemment pris, d'autres approches pourront être trouvées dans la littérature[CRA 86].

III.3 Génération de trajectoires dans l'espace des variables articulaires :

Dans ce paragraphe, on va présenter des méthodes de génération de trajectoires dans le cas où la forme de ces dernières (dans l'espace et dans le temps) est décrite en terme de fonctions de variables articulaires. Chaque point de la trajectoire est spécifié par une position et une orientation désirée du repère de l'outil terminal **OUTIL** par rapport à celui de la base **BASE**.

Chacun de ces points est converti en un ensemble de variables articulaires désirées en appliquant la cinématique inverse, ensuite une fonction lisse est trouvée pour chaque articulation du bras manipulateur. Cette dernière doit passer par les points intermédiaires préspecifiés.

La durée requise pour chaque segment de temps doit être le même pour chaque articulation.

On notera que la forme de la fonction variable articulaire ne dépend pas forcément de celle des autres variables.

La forme de la trajectoire entre deux points intermédiaires quoique simple dans l'espace des variables articulaires, est très compliquée, si on veut la calculer, dans l'espace cartésien.

La génération de trajectoires dans l'espace des variables articulaires est généralement plus facile à calculer que celle dans l'espace cartésien, et comme on ne fait pas de correspondance continue entre ces deux espaces, il n'y aura pas de problèmes de singularité du mécanisme.

III.3.1 Interpolation par des polynômes cubiques :

Considérons le problème de déplacement de l'outil terminal d'une position initiale à une position finale désirée. En utilisant la cinématique inverse, l'ensemble des valeurs des variables articulaires qui correspondent à la position et à l'orientation finales peut être calculée. La position initiale étant aussi connue dans l'espace des variables électriques, nous devons trouver pour chaque variable articulaire une fonction qui a pour valeur à t_0 la position initiale de l'articulation et à t_f la position finale de l'articulation.

Comme on peut le constater sur la figure (III.1), il existe plus qu'une fonction lisse $q(t)$ satisfaisant la condition précédente.

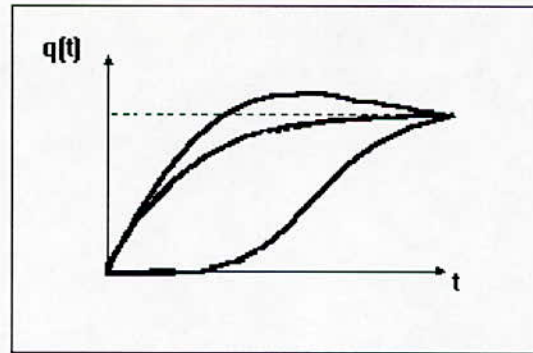


Figure III.1 : Courbes lisses liants deux points préspecifiés

Au moins quatre contraintes sur $q(t)$ sont évidentes, deux sont relatives aux valeurs de la fonction au point initial et final :

$$\begin{cases} q(t_0) = q_0, \\ q(t_f) = q_f, \end{cases} \quad (\text{III.1})$$

Les deux autres contraintes découlent de la condition de continuité de la fonction $q(t)$ en vitesse, ce qui implique dans notre cas que les vitesses initiales et finales sont nulles:

$$\begin{cases} \dot{q}(t_0) = 0, \\ \dot{q}(t_f) = 0, \end{cases} \quad (\text{III.2})$$

Ces quatre conditions peuvent être satisfaites par un polynôme d'ordre au moins égal à trois :

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (\text{III.3})$$

Ceci implique que:

$$\begin{cases} \dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2, \\ \ddot{q}(t) = 2a_2 + 6a_3 t, \end{cases} \quad (\text{III.4})$$

En combinant (III.3) et (III.4) avec les quatre conditions (III.1) et (III.2), on obtient un système algébrique de quatre équations à quatre inconnues :

$$\begin{cases} q_0 = a_0, \\ q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3, \\ 0 = a_1, \\ 0 = a_1 + 2a_2 t_f + 3a_3 t_f^2, \end{cases} \quad (\text{III.5})$$

Ceci implique que :

$$\begin{cases} a_0 = q_0, \\ a_1 = 0, \\ a_2 = 3(q_f - q_0) / t_f^2, \\ a_3 = -2(q_f - q_0) / t_f^3, \end{cases} \quad (\text{III.6})$$

En utilisant (III.6), on peut calculer le polynôme d'ordre 3 qui lie les positions initiales et finales dans l'espace électrique. Cette solution s'applique pour des vitesses initiales et finales nulles.

III.3.2 Interpolation par des polynômes cubiques avec des points intermédiaires imposés

Dans le paragraphe précédant, on a considéré des mouvements décrits par une durée et une position finale données.

En général, on veut spécifier des points intermédiaires de la trajectoire. Si on veut que le bras soit au repos pour chacun de ces points, la solution décrite précédemment pourrait être utilisée.

En pratique, on veut être capable de faire passer le manipulateur par ces points sans repos; La généralisation de la méthode précédente s'impose.

Comme dans le cas d'un seul point cible (point final), chaque point intermédiaire est pratiquement donné par une position et une orientation du repère de l'outil terminal relativement à celui de la base. Chacun de ces points intermédiaires est converti en un ensemble de valeurs de variables articulaires en appliquant la cinématique inverse du bras. Ensuite, on considérera le problème de calcul de polynômes cubiques liants entre deux points intermédiaires pour chaque articulation. Si on connaît les vitesses désirées, on peut déterminer les coefficients des polynômes cubiques comme précédemment à la seule différence que les vitesses aux points intermédiaires ne sont plus nulles mais égales à certaines valeurs imposées.

Les conditions (III.2) deviennent alors :

$$\begin{cases} \dot{q}(0) = \dot{q}_0 \\ \dot{q}(t_f) = \dot{q}_f \end{cases} \quad (\text{III.7})$$

Le système d'équations d'ordre 4 devient :

$$\begin{cases} q_0 = a_0 \\ q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \\ \dot{q}_0 = a_1 \\ \dot{q}_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 \end{cases} \quad (\text{III.8})$$

Ce qui donne:

$$\begin{cases} a_0 = q_0 \\ a_1 = \dot{q}_0 \\ a_2 = \frac{3(q_f - q_0)}{t_f^2} - \frac{2\dot{q}_0 + \dot{q}_f}{t_f} \\ a_3 = -\frac{2(q_f - q_0)}{t_f^3} + \frac{(\dot{q}_f + \dot{q}_0)}{t_f^2} \end{cases} \quad (\text{III.9})$$

En utilisant (III.9), on peut calculer le polynôme cubique liant n'importe quelles positions initiales et finales avec des vitesses initiales et finales quelconques.

Si on a les vitesses articulaires désirées en chaque point, on pourra tout simplement utiliser (III.9) pour chaque segment afin de trouver les polynômes requis.

Ils existent diverses méthodes pour spécifier les vitesses aux points intermédiaires :

- 1- L'utilisateur spécifie la vitesse désirée en chaque point intermédiaire en terme de vitesses linéaires (cartésiennes) et angulaires de l'outil terminal à l'instant voulu.
- 2- Le système choisit automatiquement les vitesses aux points intermédiaires en appliquant une heuristique convenable et ce dans l'espace cartésien ou électrique.
- 3- Le système choisit automatiquement les vitesses aux points intermédiaires de telle manière à ce que l'accélération soit continue en ces points.

Dans la première option, les vitesses cartésiennes aux points intermédiaires sont reliées aux vitesses angulaires par la *jacobienne inverse* évaluée en ces points. Si la *jacobienne* est singulière en un point intermédiaire, alors l'utilisateur n'a pas le choix d'une vitesse arbitraire en ce point.

Quoique c'est très utile qu'un algorithme de génération de trajectoires soit capable de faire passer le manipulateur par un point avec une vitesse imposée par l'utilisateur, ceci risque de constituer un lourd fardeau à l'utilisateur car il est contraint de donner toujours ces spécifications. C'est ainsi qu'un système convenable doit inclure la deuxième et/ou la troisième option.

Dans la deuxième option, le système choisit automatiquement les vitesses intermédiaires en utilisant une heuristique convenable. En effet, considérons la trajectoire spécifiée par la figure suivante :

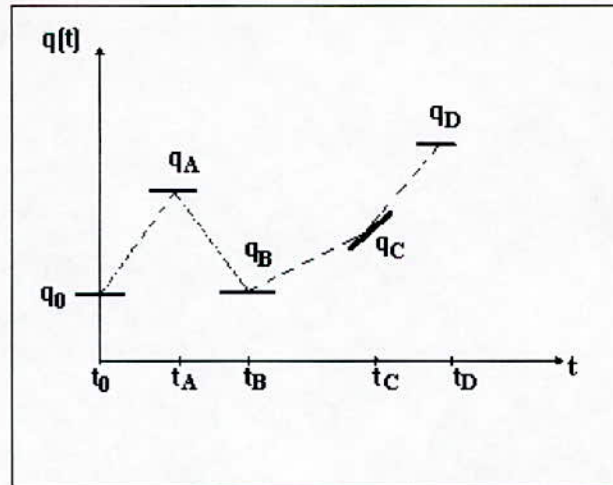


Figure III.2 : Heuristique de détermination des vitesses aux points intermédiaires.

Dans cette figure, on a fait un choix raisonnable des vitesses intermédiaires comme indiqué par les petits segments représentant les tangentes de la courbe en chaque point intermédiaire. Ce choix est le résultat de l'application d'un algorithme simple. Relions les points intermédiaires par des segments linéaires : Si la pente change de signe à un point donné alors la vitesse est choisie nulle alors que si elle ne change pas de signe, on choisit la moyenne arithmétique des deux pentes au point intermédiaire.

De cette manière, la spécification des points intermédiaires seule permet au système de calculer les vitesses en chaque point.

Dans la troisième option, le système choisit la vitesse de telle manière à ce que l'accélération soit continue aux points intermédiaires. Pour le faire, on remplace les deux conditions sur les vitesses par les deux conditions suivantes:

- 1- Vitesse continue.
- 2- Accélération continue.

III.3.3 Polynômes d'interpolation d'ordre élevé :

Parfois, des polynômes d'ordre élevé sont utilisés dans l'interpolation de la trajectoire.

Si on veut avoir la possibilité de spécifier la position, la vitesse et l'accélération au début et à la fin de chaque segment, un polynôme d'ordre au moins égal à cinq est requis :

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \tag{III.10}$$

Où les conditions sont :

$$\begin{cases} q_0 = a_0 \\ q_f = a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 + a_4t_f^4 + a_5t_f^5 \end{cases} \tag{III.11}$$

Ces conditions donnent un système linéaire de 6 équations à 6 inconnues :

$$\begin{cases} a_0 = q_0 \\ a_1 = \dot{q}_0 \\ a_2 = \ddot{q}_0 \\ a_3 = \frac{20q_f - 20q_0 - (8\dot{q}_f + 12\dot{q}_0)t_f - (3\ddot{q}_0 - \ddot{q}_f)t_f^2}{2t_f^3} \\ a_4 = \frac{30(q_0 - q_f) + (14\dot{q}_f + 16\dot{q}_0)t_f + (3\ddot{q}_0 - 2\ddot{q}_f)t_f^2}{2t_f^4} \\ a_5 = \frac{12(q_f - q_0) - 6(\dot{q}_f + \dot{q}_0)t_f + (\ddot{q}_f - \ddot{q}_0)t_f^2}{2t_f^5} \end{cases} \tag{III.12}$$

Divers algorithmes sont utilisables pour calculer des fonctions lisses (polynomiales ou autres) qui passent par un ensemble donné de points [CRA 86].

III.3.4 Fonctions linéaires avec des limites paraboliques

Un autre choix de la forme de la trajectoire est la forme linéaire :

Figure III.4 : Trajectoire linéaire avec des limites paraboliques.

On utilise la fonction linéaire et les deux fonctions paraboliques de telle manière à ce que la courbe finale soit continue en position et en vitesse.

On va assurer que les limites paraboliques ont la même base, donc la même accélération (modulo un signe) est utilisée pour les deux parties.

La figure (III.5) montre un ensemble de points intermédiaires dans l'espace articulaire pour une articulation q donnée. Des fonctions linéaires lient ces points intermédiaires avec des régions paraboliques ajoutées autour de chaque point intermédiaire.

Considérons trois points consécutifs de la courbe que nous allons appeler les points j , k et l . La durée de la région parabolique au point k est t_k , celle de la partie linéaire entre les points j et k est t_{jk} alors que la durée totale du segment liant les points j et k est t_{djk} . La vitesse durant la portion linéaire est \dot{q}_{jk} et l'accélération durant la région parabolique au point j est \ddot{q}_j .

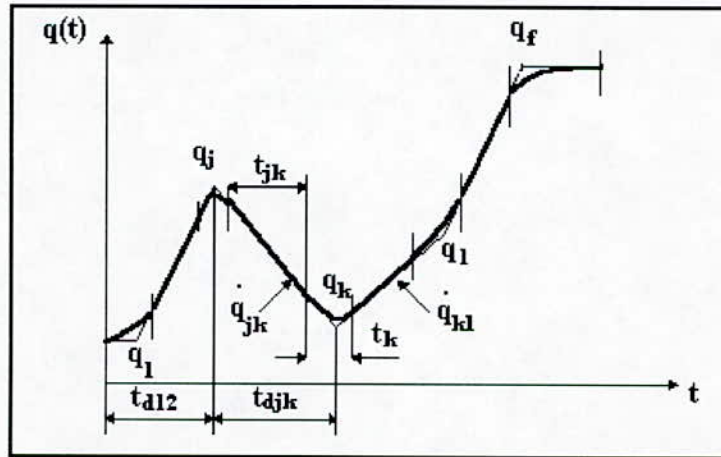


Figure III.5 : Fonctions linéaires avec des limites paraboliques (points intermédiaires imposés)

Comme dans le cas d'un segment unique, il y'a beaucoup de solutions dépendantes du choix de l'accélération dans la région parabolique.

Etants donnés tous les points j,k,l , la durée désirée t_{djk} et la valeur absolue (Amplitude) de l'accélération à utiliser à chaque point intermédiaire de la courbe $|\ddot{q}_k|$, on peut calculer la durée de la région parabolique t_k . Pour des points intérieurs à la courbe, on a:

$$\left\{ \begin{array}{l} \dot{q}_{jk} = \frac{q_k - q_j}{t_{djk}} \\ \ddot{q}_{jk} = \operatorname{sgn}(\dot{q}_{kl} - \dot{q}_{jk}) \left| \ddot{q}_k \right| \\ t_k = \frac{\dot{q}_{kl} - \dot{q}_{jk}}{\ddot{q}_k} \\ t_{jk} = t_{djk} - \frac{(t_j + t_k)}{2} \end{array} \right. \quad (\text{III.18})$$

Le premier et le dernier segment doivent être étudiés un peu différemment du fait qu'on doit comptabiliser une région parabolique complète en une extrémité du segment.

Pour le premier segment, on résout en t_1 en égalant les deux expressions de la vitesse durant la phase linéaire du segment :

$$\frac{q_2 - q_1}{t_{d12} - \frac{t_1}{2}} = \ddot{q}_1 t_1 \quad (\text{III.19})$$

Ceci donnera t_1 , d'où on tire facilement \dot{q}_{12} et t_{12} :

$$\left\{ \begin{array}{l} \ddot{q}_1 = \operatorname{sgn}(\theta_2 - \theta_1) \left| \ddot{q}_1 \right| \\ t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(\theta_2 - \theta_1)}{\ddot{\theta}_1}} \\ \dot{q}_{12} = \frac{q_2 - q_1}{t_{d12} - \frac{t_1}{2}} \\ t_{12} = t_{d12} - t_1 - \frac{t_2}{2} \end{array} \right. \quad (\text{III.20})$$

De même, on procède pour le dernier segment (celui liant les points $n-1$ et n), on a:

$$\frac{q_{n-1} - q_n}{t_{d(n-1)n} - \frac{t_n}{2}} = \ddot{q}_n t_n \quad (\text{III.21})$$

Ce qui donne:

$$\left\{ \begin{array}{l} \ddot{q}_n = \text{sgn}(q_{n-1} - q_n) \left| \ddot{q}_n \right| \\ t_n = t_{d(n-1)n} - \sqrt{t_{d(n-1)n}^2 + \frac{2(q_n - q_{n-1})}{\ddot{q}_n}} \\ \dot{q}_{(n-1)n} = \frac{q_n - q_{n-1}}{t_{d(n-1)n} - \frac{t_n}{2}} \\ t_{(n-1)n} = t_{d(n-1)n} - t_n - \frac{t_{n-1}}{2} \end{array} \right. \quad (\text{III.22})$$

En utilisant (III.18) (III.20) et (III.22), on pourra trouver les durées des parties paraboliques (durées d'accélération et de décélérations) et les vitesses pour une courbe multi-segments.

En pratique, l'utilisateur spécifie les points intermédiaires et les durées des segments seulement. Dans ce cas le système utilise des valeurs d'accéléérations par défaut.

Parfois, pour rendre les choses plus simples à l'utilisateur, le système calcule lui même les durées en se basant sur des vitesses par défaut. Pour chaque partie parabolique, on doit utiliser des accélérations suffisamment élevées pour qu'il y ait un temps suffisant pour passer à la portion linéaire du segment avant que la portion parabolique suivante ne commence.

Les résultats de ces calculs constituent un plan de la trajectoire. En temps d'exécution, ces valeurs calculées t_k , t_{jk} , \dot{q}_{jk} et \ddot{q}_k sont utilisées par un *générateur de courbe* pour évaluer q , \dot{q} et \ddot{q} à un pas d'échantillonnage donné. Celui-ci varie entre 20 et 200Hz dans les manipulateurs typiques.

Notons que dans cette méthode d'interpolation, les points intermédiaires ne sont pas atteints que si le manipulateur s'y arrête.

Dans le cas où des accélérations élevées peuvent être réalisées, la courbe va de plus en plus s'approcher des points intermédiaires. Si on veut faire passer le manipulateur par un point en s'arrêtant, le point intermédiaire est simplement répété dans les spécifications de la courbe.

Si l'utilisateur veut faire passer le manipulateur exactement par un point donné sans y arrêter, ceci peut être accommodé en utilisant la même formulation précédente avec les ajouts suivants:

-Le système remplace automatiquement le point intermédiaire par deux *pseudo points intermédiaires* aux deux cotés du point originel et la génération de la courbe prend place comme précédemment (voir figure (III.6) suivante).

-Le point intermédiaire originel va être dans la région linéaire de la courbe liant les deux pseudo points intermédiaires. L'utilisateur peut exiger que le manipulateur passe par un point avec une vitesse à spécifier. Celle-ci sera spécifiée par le système en utilisant une *heuristique* convenable.

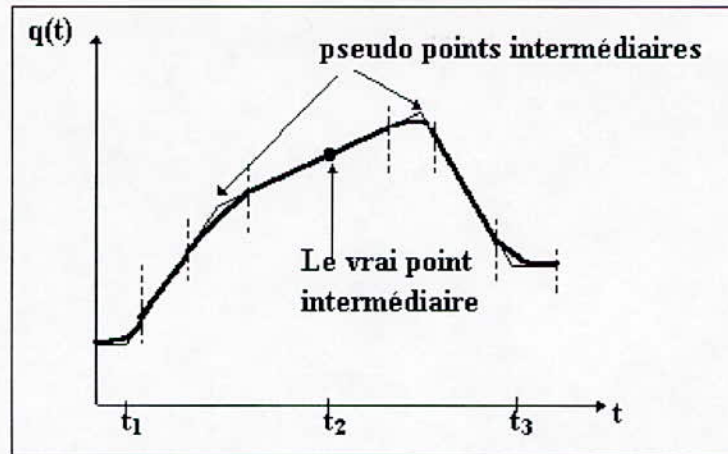


Figure III.6 : Point intermédiaire avec une vitesse préspecifiée.

III.3.6 Génération de trajectoires cycloïdales

Les fonctions d'interpolation que nous avons abordée jusqu'ici appartiennent à C^1 (continuité d'ordre 1).Lorsqu'on veut commander l'accélération, celle ci doit être continue. Une continuité d'ordre 2 est alors exigée pour la fonction d'interpolation.

Une fonction assurant cette condition est la fonction cycloïdale(cycloïde):

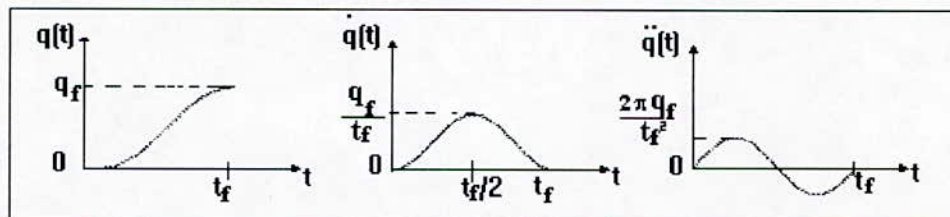


Figure III.7 : Fonction d'interpolation cycloïdale.

$$\begin{cases} q(t) = A(\omega t - \sin(\omega t)) + B \\ \dot{q}(t) = A\omega(1 - \cos(\omega t)) \\ \ddot{q}(t) = A\omega^2 \sin(\omega t) \end{cases} \quad (III.23)$$

Cette fonction doit avoir comme valeur à $t=0$, la valeur de la position initiale de l'articulation, et à $t=t_f$, la position finale de l'articulation " q_f ".

La condition de continuité de la vitesse exige que les vitesses initiale et finale soient nulles. Celle de l'accélération impose que les accélérations initiale et finale soient nulles. Ceci conduit au système suivant:

$$\begin{cases} B = q_0 \\ A(\omega t_f - \sin(\omega t_f)) + B = q_f \\ A\omega(1 - \cos(\omega t_f)) = 0 \\ A\omega^2 \sin(\omega t_f) = 0 \end{cases} \quad (\text{III.24})$$

La résolution de ce système en A, B et ω donne :

$$\begin{cases} A = \frac{q_f - q_0}{2\pi} \\ B = q_0 \\ \omega t_f = 2\pi \end{cases} \quad (\text{III.25})$$

Ainsi :

$$q(t) = \frac{q_f - q_0}{2\pi} \left(2\pi \frac{t}{t_f} - \sin\left(2\pi \frac{t}{t_f}\right) \right) + q_0 \quad (\text{III.26})$$

C'est cette fonction d'interpolation qu'on utilisera pour la commande PD linéarisante au chapitre V.

III.4 Génération de trajectoires dans l'espace cartésien

Comme il a été mentionné dans la section précédente, les courbes calculées dans l'espace des variables électriques peuvent assurer le passage de l'outil terminal par des points intermédiaires préspecifiés même si ces derniers ont été spécifiés dans des repères cartésiens.

Cependant, la forme spatiale de la courbe décrite par l'outil terminal n'est pas linéaire mais plutôt une certaine forme complexe dépendant de la cinématique du bras manipulateur utilisé.

Dans cette section, on va considérer des méthodes de génération de courbes dans lesquelles les formes des courbes sont décrites en terme de fonctions qui représentent les positions et les orientations dans l'espace cartésien.

De cette manière, on peut spécifier une forme spatiale désirée entre deux points de la trajectoire. La forme la plus usuelle est la ligne droite, mais des formes circulaires ou sinusoïdales peuvent être aussi utilisées. Chaque point de la trajectoire est spécifié par une position et une orientation désirée du repère de l'outil terminal par rapport au repère de la base.

Dans cette section, les fonctions d'interpolation sont des fonctions du temps qui décrivent des variables cartésiennes. Ces courbes peuvent être tirées directement de la définition des points

intermédiaires par l'utilisateur sans passer par la cinématique inverse. Cependant, la génération de trajectoires dans l'espace cartésien est gourmande en temps de calcul. En effet, on fait appel dans ce cas à la cinématique inverse à la fréquence d'échantillonnage de la trajectoire pour tirer les valeurs des variables articulaires nécessaires pour la commande du bras.

On trouve dans la littérature diverses méthodes de génération de trajectoires dans l'espace cartésien. Dans ce qui suit, on va introduire un exemple qui utilise le même interpolateur linéaire/parabolique utilisé dans l'espace des variables articulaires qui a été décrit dans la section précédente.

III.4.1 Mouvement linéaire dans l'espace cartésien

Souvent, on désire pouvoir spécifier une trajectoire spatiale qui entraîne un mouvement linéaire de l'extrémité de l'outil terminal.

Il est évident que si on spécifie des points intermédiaires très rapprochés s'étendant sur une ligne droite, alors le mouvement de l'extrémité de l'outil terminal va apparaître linéaire et ce pour n'importe quel choix des fonctions d'interpolation.

Cependant, il est plus convenable que l'outil terminal suit une trajectoire linéaire entre des points intermédiaires même largement séparés. La définition de mouvements linéaires cartésiens est un sous ensemble de mouvements cartésiens plus généraux dans lesquels des fonctions cartésiennes arbitraires peuvent être utilisées; Dans un système qui permet des mouvements cartésiens généraux, des trajectoires ellipsoïdales ou sinusoïdales peuvent être exécutés.

Dans l'élaboration et la génération de mouvements linéaires cartésiens, les fonctions linéaires/paraboliques sont convenables.

Durant la portion linéaire de chaque segment, l'outil terminal va décrire une trajectoire linéaire du fait que les trois composantes de la position change d'une manière linéaire.

Cependant, si on spécifie l'orientation comme une matrice de rotation en chaque point intermédiaire, on ne pourra pas interpoler linéairement les éléments de la matrice car cela ne va pas toujours donner naissance à une matrice de rotation valide. En effet, une matrice de rotation doit être composée de colonnes orthonormaux, chose qui n'est pas garantie si on interpole linéairement les éléments de la matrice entre deux matrices valides. Une solution possible à ce problème est d'utiliser une autre représentation de la rotation qui pallie à ce problème[CRA 86].

III.4.2 Problèmes géométriques

Dans la génération de trajectoires dans l'espace cartésien, on fait une correspondance continue entre l'espace cartésien et l'espace électrique. Les trajectoires cartésiennes font face à des problèmes variés relatifs à l'espace de travail et aux singularités.

Quoique la position initiale et finale du manipulateur font partie de l'espace de travail, il est possible de trouver un ou plusieurs points de la ligne droite liant ces deux points en dehors de cet espace de travail. Ceci est un exemple où une trajectoire peut être exécutée dans l'espace des variables articulaires alors qu'elle est vouée à l'échec dans l'espace cartésien.

Un autre problème est lié au fait que certaines configurations du manipulateur ne peuvent être réalisées qu'avec une fréquence d'échantillonnage très élevée et ce pour produire une vitesse donnée de l'outil terminal dans l'espace cartésien. Il n'est pas donc étonnant que certaines trajectoires décrites dans l'espace cartésien soit non réalisables.

Si par exemple, un bras manipulateur effectuant une trajectoire linéaire approche une position singulière, alors une ou plusieurs vitesses articulaires peuvent diverger à l'infini. Comme les vitesses du mécanisme sont majorées, cette situation entraînera une déviation de la trajectoire désirée.

Une approche possible est de fixer une limite supérieure pour la vitesse globale de la trajectoire de telle manière à ce que la vitesse de chaque articulation reste dans les limites possibles. Quoique les spécifications temporelles peuvent ne pas être respectées, au moins l'aspect spatial préspecifié de la trajectoire le sera.

A cause de ces problèmes inhérents à la génération de trajectoires dans l'espace cartésien, la plupart des systèmes de manipulateurs industriels supportent les deux formes de génération de trajectoires. C'est pour les raisons évoquées plus haut que la génération de trajectoire est effectuée dans l'espace des variables articulaires par défaut. La génération dans l'espace cartésien n'est utilisée que si l'application l'exige.

III.5 Génération de trajectoires en temps réel

En temps réel, la routine de génération de trajectoire construit cette dernière en terme de q , \dot{q} et \ddot{q} qui sont injectés au système de contrôle du bras manipulateur, et ce à la fréquence d'échantillonnage du système. Le résultat de l'élaboration de la trajectoire, en utilisant l'une des méthodes

d'interpolation décrites dans la section 3, est un ensemble de données pour chaque segment de trajectoire. Ces données sont utilisées par le générateur de trajectoire pour calculer q , \dot{q} et \ddot{q} en temps réel. Dans le cas de l'interpolation cubique, le générateur de trajectoire calcule simplement (III.3) au fur et à mesure que t augmente.

Lorsqu'un segment arrive à sa fin, un autre ensemble de coefficients est appelé, t est remis à zéro et la génération se poursuit.

Dans le cas d'interpolation linéaire avec des limites paraboliques, la valeur de t est testée à chaque période d'échantillonnage pour déterminer si le point en question est dans la partie linéaire ou dans la partie parabolique du segment.

Dans la partie linéaire du segment, la trajectoire est calculée pour chaque articulation par :

$$\begin{cases} q = q_j + \dot{q}_{jk} t \\ \dot{q} = \dot{q}_{jk} \\ \ddot{q} = 0 \end{cases} \quad (\text{III.27})$$

t : temps depuis le $j^{\text{ème}}$ point intermédiaire. \dot{q}_{jk} est calculé au moment de l'élaboration de la trajectoire et est donné par (III.18).

Dans la partie parabolique, la trajectoire est calculée par:

$$\begin{cases} t_{\text{inb}} = t - \left(\frac{t_j}{2} + t_{jk}\right) \\ q = q_j + \dot{q}_{jk} (t - t_{\text{inb}}) + \frac{\ddot{q}_k t_{\text{inb}}^2}{2} \\ \dot{q} = \dot{q}_{jk} + \ddot{q}_k t_{\text{inb}} \\ \ddot{q} = \ddot{q}_k \end{cases} \quad (\text{III.28})$$

\dot{q}_{jk} , \ddot{q}_k , t_j et t_{jk} sont calculés à l'étape de l'élaboration de la trajectoire par (III.18 ... III.22). Ceci

continue jusqu'à la portion linéaire suivante où t est remis à $\frac{t_j}{2}$. Cette opération se poursuit jusqu'au passage par le dernier segment.

Génération de trajectoire dans l'espace cartésien

Dans la méthode présentée dans la section (III.4), on a utilisé un générateur de trajectoire utilisant l'interpolation linéaire avec des limites paraboliques.

Cependant, les valeurs calculées représentent une position et une orientation cartésienne plutôt que des valeurs des variables électriques. Ainsi, on réécrit (III.27) et (III.28) avec 'x' représentant la composante de la position et de l'orientation cartésiennes.

Dans la partie linéaire du segment, on a pour chaque degré de liberté χ :

$$\begin{cases} \dot{x} = \dot{x}_j + \dot{x}_{jk} t \\ \ddot{x} = \ddot{x}_{jk} \\ \dddot{x} = 0 \end{cases} \quad (\text{III.29})$$

t: temps jusqu'au $j^{\text{ème}}$ point intermédiaire. \dot{x}_{jk} est déterminé lors de l'élaboration de la trajectoire et est donné par une équation analogue à (III.18-a).

Dans la partie parabolique, on a:

$$\begin{cases} t_{\text{inb}} = t - \left(\frac{t_j}{2} + t_{jk} \right) \\ \dot{x} = \dot{x}_j + \dot{x}_{jk} (t - t_{\text{inb}}) + \frac{\ddot{x}_k t_{\text{inb}}^2}{2} \\ \ddot{x} = \ddot{x}_{jk} + \ddot{x}_k t_{\text{inb}} \\ \dddot{x} = \ddot{x}_k \end{cases} \quad (\text{III.30})$$

Avec \dot{q}_{jk} , \ddot{q}_k , t_j et t_{jk} déterminés comme dans le cas de l'espace des variables électriques.

Finalement, Cette trajectoire cartésienne $(\chi, \dot{\chi}, \ddot{\chi})$ doit être convertie en quantités équivalentes dans l'espace des variables électriques.

Une solution analytique complète du problème utilise : la cinématique inverse pour calculer les valeurs des variables articulaires, la jacobienne inverse pour calculer les vitesses articulaires et la jacobienne plus ces dérivées pour calculer les accélérations[CRA 86]. Une méthode simple utilisée en pratique est la suivante:

- A la fréquence d'échantillonnage, on convertit χ sous forme de transformation homogène \mathbf{T} .
- On utilise la cinématique inverse pour tirer le vecteur de variables articulaires \mathbf{Q} .
- On utilise la différentiation numérique pour calculer $\dot{\mathbf{Q}}$ et $\ddot{\mathbf{Q}}$.

D'où l'algorithme suivant:

$$\begin{aligned}
 & \chi \rightarrow T \\
 & Q(t) = \text{Cin_Inv}(T) \\
 & \dot{Q}(t) = \frac{Q(t) - Q(t - \delta t)}{\delta t} \\
 & \ddot{Q}(t) = \frac{\dot{Q}(t) - \dot{Q}(t - \delta t)}{\delta t}
 \end{aligned}
 \tag{III.31}$$

Cin_Inv est une routine qui effectue la cinématique inverse.

Q, \dot{Q} et \ddot{Q} sont par la suite injectés au système de contrôle du bras manipulateur.

III.6 Elaboration des trajectoires en utilisant le modèle dynamique du manipulateur :

En pratique, à l'élaboration des trajectoires, on utilise une accélération par défaut (une accélération maximale). Celle-ci est fonction de la dynamique du bras et des performances des actionneurs. Ces derniers ne sont pas caractérisés par un couple maximum fixe mais plutôt par une courbe *couple/vitesse*.

Notons que le fait d'élaborer une trajectoire en prenant l'accélération maximale des actionneurs constitue une grande simplification du problème réel. Pour être sûr de ne pas dépasser les performances du matériel, on doit choisir cette valeur maximale de l'accélération avec une marge de sécurité suffisante.

Ainsi, on ne profite pas des performances en vitesse de notre manipulateur dans les méthodes introduites dans ce chapitre. Une question légitime se pose alors :

Etant donnée une trajectoire spatiale désirée de l'outil terminal, quelle est le temps minimal pour que le manipulateur atteigne le but. De tels problèmes sont résolus dans la littérature [CRA 86].

La solution est faite numériquement et tient compte de la dynamique du système ainsi que de la courbe *couple/vitesse* des actionneurs.

III.7 Perspectives :

Il serait très souhaitable de donner au système robot le point final (point cible) du manipulateur et de laisser le système choisir le nombre de points intermédiaires requis pour que le manipulateur atteigne sa cible sans heurter le moindre obstacle.

Pour arriver à cette fin, le système doit avoir un modèle du bras manipulateur, de l'espace de travail et des obstacles possibles dans cette espace. Un autre manipulateur peut travailler dans le même

espace et donc chaque bras doit être considéré comme un obstacle mobile par rapport à l'autre. Il n'y a pas en ce moment des systèmes possédants ces possibilités en commerce. Cependant, beaucoup de chercheurs travaillent sur de tels systèmes.

Chapitre IV

MODELISATION DYNAMIQUE

Chapitre IV

Modélisation Dynamique

IV.1 Introduction :

Après avoir défini la modélisation cinématique du bras manipulateur et la spécification de ses tâches, on s'intéresse dans ce chapitre à l'élaboration du modèle dynamique des robots manipulateurs à structure de chaîne simple.

Les équations dynamiques du mouvement d'un bras manipulateur, sont l'ensemble des équations mathématiques décrivant le comportement dynamique du manipulateur. Il s'agit d'établir les équations différentielles non linéaires complexes qui relient les efforts délivrés par les actionneurs $\Gamma_i(t)$ aux variables articulaires $q_i(t)$, aux vitesses articulaires $\dot{q}_i(t)$ et aux accélérations articulaires $\ddot{q}_i(t)$.

L'ensemble de ces équations constitue ce qui est d'usage d'appeler le modèle dynamique du manipulateur.

Dans le cas simple, on peut établir le modèle dynamique du système articulé facilement et aboutir aux équations dynamiques, mais dans le cas général (bras à "n" degré de liberté) la complexité structurelle du système nous oblige à choisir une approche systématique pour résoudre le problème. Il faut noter que la détermination des équations dynamiques n'est possible que si l'on dispose d'une description complète du manipulateur : nombre de degrés de liberté, dimensions, moments d'inerties.

L'élaboration des équations dynamiques d'un système mécanique articulé nécessite de nombreux calculs et transformations mathématiques, cette opération est généralement confiée à un ordinateur qui génère de façon systématique les équations sous forme littérale.

Le modèle dynamique d'un bras manipulateur peut être déterminé à partir des lois connues de la physique, telles les lois de Newton et de Lagrange ; Les formalismes les plus courants sont le formalisme d'Euler-Lagrange (E-L), la méthode de Newton-Euler (N-E), le formalisme de d'Alembert qui fait appel au principe des puissances virtuelles et la méthode de Bond-Graphs

[FU 87]. Cependant il faut noter qu'il existe plusieurs variantes de ces approches, telles que la "Uicker's-L-E" , "Hollerbach's récursive Lagrange", Loi Lee's generalized d'Alembert G.D" [FU 87].

Ces approches peuvent aboutir à des équations dont la structure est différente, car elles ont été conçues pour différentes tâches, telles que : la synthèse et l'analyse de la loi de commande, ou bien la simulation par ordinateur du mouvement du manipulateur.

Deux exploitations du modèle dynamique sont possibles :

- Résoudre les équations différentielles pour obtenir les mouvements $q_i(t)$ pour des efforts actionneurs $\Gamma_i(t)$ données et ce pour simuler sur ordinateur le comportement dynamique du bras manipulateur, on parle alors de modèle dynamique direct.
- Calculer les efforts actionneurs $\Gamma_i(t)$ pour des mouvements articulaires $q_i(t)$ connus. Cette approche est nécessaire pour la conception d'une commande en vue de satisfaire les performances imposées. On parle alors du modèle dynamique inverse.

IV.2 Le formalisme d'Euler-Lagrange [PAU 81] [FU 87]

Le formalisme de Lagrange-Euler (E-L) est utilisé pour modéliser le comportement dynamique d'un robot.

Cette approche particulière est assez simple à mettre en oeuvre et elle est bien adaptée aux techniques de calcul manuel ainsi qu'aux méthodes de calculs assistés par ordinateur.

L'application directe du formalisme d'E-L associé à la représentation D-H aboutit à un algorithme compacte est efficace pour décrire les équations dynamiques du mouvement.

Le formalisme de mise en équation d'un système est assez pratique car les énergies cinétiques et potentielles sont des grandeurs additives, ainsi leur détermination est aisée.

L'algorithme est exprimé sous forme matricielle, ce qui facilite l'analyse et l'implémentation sur ordinateur.

L'équation d'E-L est donnée par [PAU 81] [FU 87] :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L_d}{\partial q_i} \right) + \left(\frac{\partial E_d}{\partial \dot{q}_i} \right) = \Gamma_i \quad (\text{IV.1})$$

$i = 1, \dots, n$ pour un bras à "n" degrés de liberté.

Où :

L désigne le Lagrangien qui est défini par : $L = K - P = E_c - E_p$.

Avec :

K ou E_c : énergie cinétique.

P ou E_p : énergie potentielle.

E_d : Fonction de dissipation de Rayleigh.

Γ_i : Forces généralisées (efforts actionneurs).

- si " q_i " est une coordonnée linéaire, Γ_i est une force (liaison glissière).
- si " q_i " est une coordonnée angulaire, Γ_i est un couple effort.

Pour aboutir aux équations dynamiques de (E-L) (IV.1), il sera fait appel aux matrices jT_i de Denavit-Hartenberg, ainsi qu'aux cinq étapes de calcul suivantes :

- 1) Calcul de la vitesse d'un point quelconque appartenant à un segment (i) du manipulateur
 $i=1, \dots, 6$
- 2) Calcul de l'énergie cinétique du segment (i) et totale
- 3) Calcul de l'énergie potentielle du segment (i) et l'énergie potentielle totale
- 4) Evaluation du Lagrangien L
- 5) Procéder par dérivation de L pour aboutir aux effets dynamiques (IV.1)

IV.2.1 Vitesse d'un point du manipulateur

- Soit un point caractérisé par sa position par rapport au segment (i) auquel il appartient par le vecteur ${}^i r$.

- Sa position par rapport au repère de base (R_o) sera :

$${}^0 r = r = T_i \cdot {}^i r = (A_1 \cdot A_2 \cdot \dots \cdot A_i) \cdot {}^i r = T_i(q_1, q_2, q_3, \dots, q_i) \cdot {}^i r$$

$$\text{Sa vitesse est : } v = \frac{dr}{dt} = \left(\sum_{j=1}^i \frac{\partial T_i}{\partial q_j} \dot{q}_j \right) \cdot {}^i r$$

(Différentielle totale de $T_i(q_1, q_2, q_3, \dots, q_i)$)

$$\text{et son carré vaut : } v^2 = \left(\frac{dr}{dt} \right)^2 = \text{Trace} \left[\left(\frac{dr}{dt} \right) \otimes \left(\frac{dr}{dt} \right)^T \right]$$

Avec \otimes produit tensoriel :

Si $A = [a, b, c]^T$ Alors

$$A \otimes A^T = \begin{bmatrix} a^2 & ab & ac & a \\ ba & b^2 & bc & b \\ ca & cb & c^2 & c \\ a & b & c & 1 \end{bmatrix}$$

D'où :

$$\begin{aligned} v^2 &= \text{Trace} \left[\sum_{j=1}^i \left(\frac{\partial T_i}{\partial q_j} \dot{q}_j \cdot {}^i r \right) \otimes \sum_{k=1}^i \left(\frac{\partial T_i}{\partial q_k} \dot{q}_k \cdot {}^i r \right)^T \right] \\ &= \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \cdot ({}^i r) \otimes ({}^i r)^T \left(\frac{\partial T_i}{\partial q_k} \right)^T \dot{q}_j \dot{q}_k \right] \end{aligned}$$

IV.2.2 Energie cinétique d'un segment (i)

* L'énergie cinétique d'une particule de masse dm localisé sur le segment (i) autour du point ${}^i r$

$$\begin{aligned} \text{est } dK_i &= \frac{1}{2} \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \cdot ({}^i r) \otimes ({}^i r)^T \left(\frac{\partial T_i}{\partial q_k} \right)^T \dot{q}_j \dot{q}_k \right] dm \\ &= \frac{1}{2} V^2 dm \end{aligned}$$

$$= \frac{1}{2} \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \cdot ({}^i r \otimes {}^i r^T \cdot dm) \left(\frac{\partial T_i}{\partial q_k} \right)^T \dot{q}_j \dot{q}_k \right]$$

L'énergie cinétique du segment (i) sera :

$$\begin{aligned} K_i &= \int_{(\text{segment } i)} dK_i \\ &= \frac{1}{2} \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \cdot \left(\int_i {}^i r \otimes {}^i r^T \cdot dm \right) \left(\frac{\partial T_i}{\partial q_k} \right)^T \dot{q}_j \dot{q}_k \right] \end{aligned}$$

Le terme sous le signe somme représente l'inertie du segment (i) et est appelé matrice pseudo-inertie du segment (i). Il sera noté J_i .

Avec :

$$J_i = \begin{bmatrix} \int_i (x^i)^2 dm & \int_i x^i y^i dm & \int_i x^i z^i dm & \int_i x^i dm \\ \int_i x^i y^i dm & \int_i (y^i)^2 dm & \int_i y^i z^i dm & \int_i y^i dm \\ \int_i x^i z^i dm & \int_i z^i y^i dm & \int_i (z^i)^2 dm & \int_i z^i dm \\ \int_i x^i dm & \int_i y^i dm & \int_i z^i dm & \int_i dm \end{bmatrix} \quad (IV.2)$$

Sachant que ${}^i r = [{}^i x, {}^i y, {}^i z]^T$.

Autre écriture de la matrice " J_i " :

a) Moment d'inertie d'un corps solide :

Tout corps solide (S) se caractérise dynamiquement par :

- Six moments d'inertie d'ordre 2 :

$$I_{xx} = \int_{(s)} (y^2 + z^2) dm \quad ; \quad I_{yy} = \int_{(s)} (x^2 + z^2) dm$$

$$I_{zz} = \int_{(s)} (x^2 + y^2) dm \quad ; \quad I_{xy} = \int_{(s)} xy dm$$

$$I_{xz} = \int_{(s)} xz dm \quad ; \quad I_{yz} = \int_{(s)} yz dm$$

- Trois moments d'inertie d'ordre 1 :

$$m \bar{x} = \int_{(s)} x dm \quad ; \quad m \bar{y} = \int_{(s)} y dm \quad ; \quad m \bar{z} = \int_{(s)} z dm$$

Avec $(\bar{x}, \bar{y}, \bar{z})$ coordonnées du centre de gravité du solide (S), exprimés dans le repère auquel il est lié.

- Un moment d'inertie d'ordre zéro :

$$\text{Il s'agit de la masse } m = \int_{(s)} dm$$

On peut alors écrire la matrice (J_i) à l'aide de ces moments d'inertie par rapport au référentiel

$R_i(x_i, y_i, z_i)$:

$$J_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (IV.3)$$

Où l'indice (i) fait référence au segment (i) du bras manipulateur.

L'énergie cinétique totale du bras manipulateur à "n" degré de liberté s'exprime en sommant les énergies cinétiques de chaque segment ce qui donne :

$$K = \sum_{i=1}^n k_i \\ = \frac{1}{2} \sum_{i=1}^n \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \dot{q}_j \dot{q}_k \right]$$

Cette équation représente l'énergie cinétique de la structure mécanique du manipulateur. Il existe en fait une autre importante contribution de l'énergie cinétique; C'est celle des actionneurs :

$$K_{\text{actionneur}(i)} = \frac{1}{2} I_{ai} \dot{q}_i^2$$

Où :

I_{ai} représente le moment d'inertie de l'actionneur (i).

Ainsi, l'énergie cinétique totale devient alors :

$$K = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_j} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \dot{q}_j \dot{q}_k \right) + \frac{1}{2} \sum_{i=1}^n I_{ai} (\dot{q}_i)^2 \quad (IV.4)$$

IV.2.3 Energie potentielle du bras manipulateur

Soit P_i l'énergie potentielle du segment (i), elle sera donnée par :

$$P_i = -m_i \cdot g^T \cdot \bar{r}_i \quad (IV.5)$$

Avec $g = [g_x, g_y, g_z]^T$ champ de l'apesanteur .

\bar{r}_i : Coordonnés du centre de masse du segment (i) par rapport au référentiel de base (R_0).

Or $\bar{r}_i = T_i \cdot r_i$; r_i représente dans ce cas le centre de masse du segment (i) par rapport au repère

$R_i = (x_i, y_i, z_i)$ auquel il est lié.

P_i sera donc : $P_i = -m_i \cdot g^T \cdot T_i \cdot r_i^i$, et l'énergie potentielle totale du manipulateur sera :

$$P_i = \sum_{i=1}^n -m_i \cdot g^T \cdot T_i \cdot r_i^i \quad (\text{IV.6})$$

IV.2.4 Le Lagrangien

La fonction de Lagrange sera donnée par ($L = K - P$) :

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_j} \cdot J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \dot{q}_j \cdot \dot{q}_k \right) + \frac{1}{2} \sum_{i=1}^n I_{ai} (\dot{q}_i)^2 + \sum_{i=1}^n m_i \cdot g^T \cdot T_i \cdot r_i^i \quad (\text{IV.7})$$

IV.2.5 Equations dynamiques du bras manipulateur

Nous obtenons les équations dynamiques par application de l'équation (IV.1) :

$$\Gamma_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right) + \left(\frac{\partial E_d}{\partial \dot{q}_i} \right)$$

Avec E_d : Fonction de dissipation du Rayleigh que l'on écrira généralement sous la forme :

$$E_d = \frac{1}{2} \sum_{i=1}^n f_i (\dot{q}_i)^2 \quad \text{où } f_i \text{ est un coefficient de frottement visqueux ramené à l'articulation.}$$

. Développons tout d'abord le 1^{er} terme : $\frac{\partial L}{\partial \dot{q}_p}$:

$$\frac{\partial L}{\partial \dot{q}_p} = \frac{1}{2} \sum_{i=1}^n \sum_{k=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_p} \cdot J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \dot{q}_k + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_j} \cdot J_i \left(\frac{\partial T_i}{\partial q_p} \right)^T \right) \dot{q}_j + I_{ap} \dot{q}_p$$

Avec $p = 1, 2, \dots, n$

Remplaçons pour la seconde sommation l'indice (j) par (k) :

$$\frac{\partial L}{\partial \dot{q}_p} = \frac{1}{2} \sum_{i=1}^n \sum_{k=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_p} \cdot J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \dot{q}_k + \frac{1}{2} \sum_{i=1}^n \sum_{k=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_k} \cdot J_i \left(\frac{\partial T_i}{\partial q_p} \right)^T \right) \dot{q}_k + I_{ap} \dot{q}_p$$

$$\begin{aligned} \text{Sachant aussi que : } \text{Trace} \left(\frac{\partial T_i}{\partial q_p} \cdot J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) &= \text{Trace} \left(\frac{\partial T_i}{\partial q_p} \cdot J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right)^T \\ &= \text{Trace} \left(\frac{\partial T_i}{\partial q_k} \cdot J_i \left(\frac{\partial T_i}{\partial q_p} \right)^T \right) \end{aligned}$$

Car la matrice J_i est symétrique : ($J_i = J_i^T$).

d'autres part : $\frac{\partial T_i}{\partial q_p} = 0$, pour $p > i$; ce qui conduit finalement à :

$$\frac{\partial L}{\partial \dot{q}_p} = \sum_{i=1}^n \sum_{k=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_p} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \dot{q}_k + I_{ap} \dot{q}_p$$

La dérivation de ce terme par rapport au temps donne :

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_p} = \sum_{i=p}^n \sum_{k=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_p} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \ddot{q}_k + I_{ap} \ddot{q}_p + \sum_{i=p}^n \sum_{k=1}^i \sum_{m=1}^i \text{Trace} \left(\frac{\partial^2 T_i}{\partial q_m \partial q_p} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \dot{q}_k \dot{q}_m + \sum_{i=p}^n \sum_{k=1}^i \sum_{m=1}^i \text{Trace} \left(\frac{\partial^2 T_i}{\partial q_m \partial q_k} J_i \left(\frac{\partial T_i}{\partial q_p} \right)^T \right) \dot{q}_k \dot{q}_m \quad (IV.8)$$

Le second terme de l'équation d'Euler -Lagrange est :

$$\frac{\partial L}{\partial q_p} = \frac{1}{2} \sum_{i=p}^n \sum_{k=1}^i \sum_{j=1}^i \text{Trace} \left(\frac{\partial^2 T_i}{\partial q_j \partial q_p} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \dot{q}_j \dot{q}_k + \frac{1}{2} \sum_{i=p}^n \sum_{j=1}^i \sum_{k=1}^i \text{Trace} \left(\frac{\partial^2 T_i}{\partial q_k \partial q_p} J_i \left(\frac{\partial T_i}{\partial q_j} \right)^T \right) \dot{q}_j \dot{q}_k + \sum_{i=p}^n m_i \cdot g^T \cdot \frac{\partial T_i}{\partial q_p} \cdot r_i$$

Interchangeons les indices "j" et "k" dans la seconde sommation, on obtient une quantité identique au 1^{er} terme :

$$\frac{\partial L}{\partial q_p} = \sum_{i=p}^n \sum_{k=1}^i \sum_{j=1}^i \text{Trace} \left(\frac{\partial^2 T_i}{\partial q_j \partial q_p} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \dot{q}_j \dot{q}_k + \sum_{i=p}^n m_i \cdot g^T \cdot \frac{\partial T_i}{\partial q_p} \cdot r_i \quad (IV.9)$$

A partir des équations (IV.1), (IV.8) et (IV.9), on peut calculer les efforts actionneurs Γ_p :

$$\Gamma_p = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_p} - \frac{\partial L}{\partial q_p} = \sum_{i=p}^n \sum_{k=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_p} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \ddot{q}_k + I_{ap} \ddot{q}_p + f_i \dot{q}_i + \sum_{i=p}^n \sum_{k=1}^i \sum_{m=1}^i \text{Trace} \left(\frac{\partial^2 T_i}{\partial q_k \partial q_m} J_i \left(\frac{\partial T_i}{\partial q_p} \right)^T \right) \dot{q}_k \dot{q}_m - \sum_{i=p}^n m_i \cdot g^T \cdot \frac{\partial T_i}{\partial q_p} \cdot r_i \quad (IV.10)$$

Les différents termes de l'équation IV.10 sont indépendants de l'ordre de sommation, donc écrivons (IV.10) sous cette forme :

$$\Gamma_i = \sum_{j=1}^n D_{ij} \ddot{q}_j + I_{ai} \ddot{q}_i + f_i \dot{q}_i + \sum_{j=1}^n \sum_{k=1}^n D_{ijk} \dot{q}_j \dot{q}_k + D_i \quad (\text{IV.11})$$

Avec le changement d'indice : "p", "i", "k" par "i", "p", "j" respectivement.

Où :

$$D_{ij} = \sum_{p=\max(i,j)}^n \text{Trace} \left(\frac{\partial T_p}{\partial q_j} J_p \left(\frac{\partial T_p}{\partial q_i} \right)^T \right) \quad (\text{IV.11.a})$$

$$D_{ijk} = \sum_{p=\max(i,j,k)}^n \text{Trace} \left(\frac{\partial^2 T_p}{\partial q_j \partial q_k} J_p \left(\frac{\partial T_p}{\partial q_i} \right)^T \right) \quad (\text{IV.11.b})$$

$$D_i = - \sum_{p=i}^n m_p \cdot g^T \cdot \frac{\partial T_p}{\partial q_i} \cdot r_p \quad (\text{IV.11.c})$$

◆ **Remarques :**

a) $D_{ij} = D_{ji}$

b) $D_{ijk} = D_{ikj}$

c) Signification physique des termes D_i , D_{ij} , D_{ijk} :

- Les coefficients D_{ij} , D_{ijk} , D_i dépendent de la position q_i et de la vitesse angulaire \dot{q}_i de chaque articulation, ainsi que des paramètres d'inertie telles que la masse et la longueur des segments. A partir des équations données ci-dessus, il est possible de donner à chacun d'eux le sens physique correspondant.

- Les termes D_{ii} représentent l'inertie propre que présente l'articulation (i)

- D_{ij} est l'inertie comptée entre l'articulation (i) et (j) et est interprété comme un effet d'accélération angulaire sur l'articulation j causé par l'effort Γ_i sous l'articulation (i) et vice-versa.

Ce terme dépend, en général, de la configuration cinématique du bras manipulateur, ie de la position relative des articulations i et j.

- D_{ijk} est relié aux vitesses angulaires des articulations j et k (\dot{q}_j et \dot{q}_k), qui, en se combinant induisent un couple de réaction exercé sur l'articulation "i".

En particulier pour $j = k$, il s'agira de la force centrifuge générée par la vitesse angulaire \dot{q}_j et exercée sur l'articulation i, autrement dit, il s'agira de la force de Coriolis générée par les vitesses angulaires des articulations j et k ie \dot{q}_j et \dot{q}_k .

- D_i représente l'effet de la gravitation sur l'articulation "i".

d) Hypothèses de modélisation :

-Lors de l'établissement du modèle précédent du robot articulé, on a fait quelques hypothèses simplificatrices qui approchent la réalité physique :

- Les frottements sont de nature visqueuse, et linéaires par rapport à la vitesse généralisée \dot{q}_j .
- Les différentes liaisons sont rigides.
- Les actionneurs sont idéaux, c'est à dire que la force généralisée est directement proportionnelle au signal de commande d'où : $\Gamma_i = k_i u_i$ ($i = 1, 2, 3, \dots$)

Avec :

k_i : Coefficient de proportionnalité de l'actionneur.

u_i : Tension de commande de l'actionneur "i".

- Les capteurs ont des gains unitaires et de dynamique négligé.

e) Ecriture matricielle :

. Le modèle dynamique du bras manipulateur peut s'écrire sous une forme matricielle comme:

$$T = H(q)\ddot{q} + h(q, \dot{q}) + g(q) + f(\dot{q}) \quad (IV.12)$$

Avec : $T = [\Gamma_1, \Gamma_2, \dots, \Gamma_n]^T$ représente le vecteur effort actionneurs de dimension $n \times 1$.

$H(q) = [D_{ij}(q)]$ matrice d'inertie de dimension $n \times n$ dont les termes sont fonctions du vecteur composé des coordonnées généralisées du manipulateur $q = (q_1, q_2, \dots, q_n)^T$.

$h(q, \dot{q}) = q^T \cdot C(q) \dot{q}$: vecteur de dimension $n \times 1$.

Avec $c(q) = [D_{ijk}(q)]$ matrice traditionnelle $n \times n \times n$ contenant les termes centrifuges et de Coriolis

$g(q)$: Vecteur $n \times 1$ des efforts gravitationnels.

Une autre écriture matricielle du modèle dynamique serait:

$$\Gamma = A(q)\ddot{q} + B(q)\left[\dot{q}\dot{q}\right] + 2.C(q).\left[\dot{q}^2\right] + D(q) \quad (IV.13)$$

Où : $A(q)$: Matrice des coefficients inertiels de dimension $n \times n$.

$B(q)$: Matrice représentant les effets de Coriolis, de dimension $n \times C_n^2$.

$$\text{Avec : } C_n^2 = \frac{n!}{(n-2)!2!} = \frac{n(n-1)}{2}.$$

$C(q)$: Matrice traduisant les effets centrifuges de dimension $n \times n$.

$\left[\dot{q}\dot{q}\right] = \left[\dot{q}_1\dot{q}_2 \quad \dot{q}_1\dot{q}_3 \quad \dots \quad \dot{q}_{n-1}\dot{q}_n\right]$: Vecteur des produits des vitesses articulaires.

$$\begin{bmatrix} \dot{q}^2 \end{bmatrix} = \begin{bmatrix} \dot{q}_1^2 & \dot{q}_2^2 & \dots & \dot{q}_n^2 \end{bmatrix}^T : \text{Vecteur des vitesses articulaires au carré.}$$

Γ : Vecteur des couples de commande de dimension $n \times 1$.

IV.2.6 Simplification du modèle Dynamique :

-a- Simplification des termes D_{ij} :

La formule (IV.8.a) précédente est très lourde pour le calcul des paramètres D_{ij} , pour cela on va essayer de présenter ces paramètres sous une autre forme, mieux adapté au calcul.

On peut montrer que pour une matrice de transformation homogène $T_p = A_1 A_2 \dots A_p$, la relation suivante :

$$\frac{\partial T_p}{\partial q_i} = T_p \cdot \Delta_i ; \text{ avec } 1 \leq i \leq p \quad (\text{IV.14})$$

Avec : Δ matrice de translation et rotation différentielle [PAU 81] correspondant à un changement différentiel autour de l'axe z_i du repère $R_i (x_i, y_i, z_i)$ par rapport au repère T_p .

Ainsi on peut montrer aussi la relation suivante [PAU 81] :

$$T_p \Delta_i = (A_i A_{i+1} \dots A_p)^{-1} \Delta_i (A_i A_{i+1} \dots A_p), \text{ avec } A_i A_{i+1} \dots A_p = {}^{i-1}T_p.$$

Les éléments de la matrice ${}^{T_p}\Delta_i$ sont [PAU 81] :

• Si la liaison (i) est rotative :

$$\begin{aligned} {}^{T_p}d_{ix} &= {}^{i-1}n_{py} \cdot {}^{i-1}p_{px} - {}^{i-1}n_{px} \cdot {}^{i-1}p_{py} \\ {}^{T_p}d_{iy} &= {}^{i-1}o_{py} \cdot {}^{i-1}p_{px} - {}^{i-1}o_{px} \cdot {}^{i-1}p_{py} \\ {}^{T_p}d_{iz} &= {}^{i-1}a_{py} \cdot {}^{i-1}p_{px} - {}^{i-1}a_{px} \cdot {}^{i-1}p_{py} \end{aligned} \quad (\text{IV.15})$$

$$\text{Avec } {}^{T_p}d_i = [{}^{T_p}d_{ix}, {}^{T_p}d_{iy}, {}^{T_p}d_{iz}] \quad \text{et } {}^{T_p}\delta_i = [{}^{i-1}n_{pz}, {}^{i-1}o_{pz}, {}^{i-1}a_{pz}]$$

• Si la liaison (i) est prismatique :

$${}^{T_p}d_i = [{}^{i-1}n_{pz}, {}^{i-1}o_{pz}, {}^{i-1}a_{pz}] \quad \text{et } {}^{T_p}\delta_i = [0, 0, 0].$$

$$\text{Avec } : {}^{i-1}T_p = \begin{bmatrix} {}^{i-1}n_{px} & {}^{i-1}o_{px} & {}^{i-1}a_{px} & {}^{i-1}p_{px} \\ {}^{i-1}n_{py} & {}^{i-1}o_{py} & {}^{i-1}a_{py} & {}^{i-1}p_{py} \\ {}^{i-1}n_{pz} & {}^{i-1}o_{pz} & {}^{i-1}a_{pz} & {}^{i-1}p_{pz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{IV.16})$$

◆ **Remarque :**

Pour ne pas alourdir les notations on va abréger ${}^T P \Delta_i, {}^T P d_i, {}^T P \delta_i$ par ${}^P \Delta_i, {}^P d_i, {}^P \delta_i$ respectivement. Si on remplace maintenant l'expression de $\frac{\partial T_p}{\partial q_i}$, de l'équation (IV.14), dans (IV.11.a) on aura :

$$D_{ij} = \sum_{p=\max(i,j)}^n \text{Trace} \left\{ T_p \cdot {}^P \Delta_j \cdot J_p \cdot ({}^P \Delta_i)^T \cdot (T_p)^T \right\}.$$

Par développement des termes du milieu de cette expression, on aboutit à :

$$D_{ij} = \sum_{p=\max(i,j)}^n \text{Trace} \left\{ T_p \times \begin{bmatrix} 0 & -{}^P \delta_{jz} & {}^P \delta_{jy} & {}^P d_{jx} \\ {}^P \delta_{jz} & 0 & -{}^P \delta_{jx} & {}^P d_{jy} \\ -{}^P \delta_{jy} & {}^P \delta_{jx} & 0 & {}^P d_{jz} \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \frac{-I_{pxx} + I_{pyy} + I_{pzz}}{2} & I_{pxy} & I_{pxz} & m_p \bar{x}_p \\ I_{pxy} & \frac{I_{pxx} - I_{pyy} + I_{pzz}}{2} & I_{pyz} & m_p \bar{y}_p \\ I_{pxz} & I_{pyz} & \frac{I_{pxx} + I_{pyy} - I_{pzz}}{2} & m_p \bar{z}_p \\ m_p \bar{x}_p & m_p \bar{y}_p & m_p \bar{z}_p & m_p \end{bmatrix} \times (T_p)^T \right\}$$

Le produit de ces 3 matrices donne une matrice "X" dont sa 4^{ème} ligne et 4^{ème} colonne sont nulles.

On montre aussi que le produit :

$T_p \cdot X \cdot (T_p)^T$ est égal a la matrice X, en tenant compte que les éléments de T_p forment un trièdre direct.

On aboutit finalement à la relation finale suivante :

$$D_{ij} = \sum_{p=\max(i,j)}^n m_p \left[({}^P \delta_i)^T \cdot K_p \cdot {}^P \delta_j + {}^P d_i \cdot {}^P d_j - \frac{p}{r_p} \cdot ({}^P d_i \wedge {}^P \delta_j + {}^P d_j \wedge {}^P \delta_i) \right] \tag{IV.17}$$

$$\text{Où } K_p = \begin{bmatrix} k^2_{pxx} & -k^2_{pxy} & -k^2_{pxz} \\ -k^2_{pxy} & k^2_{pyy} & -k^2_{pzy} \\ -k^2_{pxz} & -k^2_{pzy} & k^2_{pzz} \end{bmatrix}$$

Avec $K_{p\alpha\beta} = \frac{I_{p\alpha\beta}}{m_p}$; $\alpha, \beta = x, y, z$

- Si on suppose encore que les termes non diagonaux (les moments d'inertie : I_{xy}, I_{xz}, I_{yz}) de K_p sont nulles (ce qui est généralement toujours le cas pour un choix judicieux du repère $R_i (x_i, y_i, z_i)$ lié au segment (i) : En effet on montre que pour un segment symétrique par rapport à l'axe z_i ou x_i ou y_i les termes ci dessus seront nuls)

On aura :

$$D_{ij} = \sum_{p=\max(i,j)}^n m_p \left\{ \begin{aligned} & \left[\delta_{ix}^p \cdot k_{pxx}^2 \cdot \delta_{jx}^p + \delta_{iy}^p \cdot k_{pyy}^2 \cdot \delta_{jy}^p + \delta_{iz}^p \cdot k_{pzz}^2 \cdot \delta_{jz}^p \right] + \left[d_i^p \cdot d_j^p \right] + \\ & \left[\bar{r}_p \cdot \left(d_i^p \wedge d_j^p + d_j^p \wedge d_i^p \right) \right] \end{aligned} \right\} \quad (IV.18)$$

-b- Simplification des termes $D_{ii} (i=1, \dots, n)$:

Si ($i=j$) dans l'équation (II.17), d'autres simplifications sont encore possibles :

$$D_{ii} = \sum_{p=i}^n m_p \left\{ \begin{aligned} & \left[\left(\delta_{ix}^p \right)^2 \left(k_{pxx} \right)^2 + \left(\delta_{iy}^p \right)^2 \left(k_{pyy} \right)^2 + \left(\delta_{iz}^p \right)^2 \left(k_{pzz} \right)^2 \right] \\ & \left[d_i^p \cdot d_i^p \right] + \left[2 \cdot \bar{r}_p \cdot \left(d_i^p \times d_i^p \right) \right] \end{aligned} \right\} \quad (IV.19)$$

• Si la liaison "i" est rotative, on obtient à partir de l'équation (IV.15) :

$$D_{ii} = \sum_{p=i}^n m_p \left\{ \begin{aligned} & \left[n_{pz}^2 \cdot k_{pxx}^2 + o_{pz}^2 \cdot k_{pyy}^2 + a_{pz}^2 \cdot k_{pzz}^2 \right] + \left[\bar{p}_p \cdot \bar{p}_p \right] \\ & \left[2 \cdot \bar{r}_p \cdot \left\{ \left(\bar{p}_p \cdot \bar{n}_p \right) \vec{i} + \left(\bar{p}_p \cdot \vec{o}_p \right) \vec{j} + \left(\bar{p}_p \cdot \vec{a}_p \right) \vec{k} \right\} \right] \end{aligned} \right\} \quad (IV.20)$$

Avec $\bar{r}^p = \left[\bar{x}_p, \bar{y}_p, \bar{z}_p \right]^T$.

• Si la liaison "i" est prismatique, alors d'après (IV.16) :

$${}^p\delta_i = 0, {}^p d_i \cdot {}^p d_i = 0 \quad \text{et} \quad D_{ii} = \sum_{p=i}^n m_p \quad (IV.21)$$

-c- Simplification des termes de gravité $D_i (i=1, \dots, n)$:

les équations (IV.8.c) et (IV.11) nous permettent d'écrire :

$$D_i = \sum_{p=i}^n -m_p \cdot g^T \cdot T_p \cdot {}^p \Delta_i \cdot \bar{r}_p \quad (IV.22)$$

Si on remplace T_p par $T_{i-1} \cdot {}^{i-1} T_p$ et en introduisant le terme $({}^{i-1} T_p)^{-1} \cdot ({}^{i-1} T_p)$ après ${}^p \Delta_i$ on aura :

$$D_i = \sum_{p=i}^n -m_p \cdot g^T \cdot T_{i-1} \cdot {}^{i-1}T_p \cdot {}^p\Delta_i \cdot ({}^{i-1}T_p)^{-1} \cdot {}^{i-1}T_p \cdot {}^p r_p \quad (IV.23)$$

Mais comme :

$$\begin{aligned} {}^{i-1}\Delta_i &= {}^{i-1}T_p \cdot {}^p\Delta_i \cdot ({}^{i-1}T_p)^{-1} \\ {}^p r_p &= {}^i T_p \cdot {}^p r_p \end{aligned}$$

Alors :

$$D_i = -g^T \cdot T_{i-1} \cdot {}^{i-1}\Delta_i \sum_{p=i}^n m_p \cdot {}^{i-1} r_p$$

Si on pose maintenant :

$${}^{i-1}g = -g^T \cdot T_{i-1} \cdot {}^{i-1}\Delta_i \quad (IV.24)$$

L'expression finale de D_i devient :

$$D_i = -{}^{i-1}g \sum_{p=i}^n m_p \cdot {}^{i-1} r_p \quad (IV.25)$$

• **Développement du terme ${}^{i-1}g$:**

A partir de (IV.21) :

$${}^{i-1}g = -[g_x, g_y, g_z, 0] \cdot \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -\delta_z & \delta_y & d_x \\ \delta_z & 0 & -\delta_x & d_y \\ -\delta_y & \delta_x & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

• **Si la liaison "i" est rotative :**

Dans ce cas ${}^{i-1}\Delta_i$ correspond à une rotation autour de l'axe (z) du repère (R_{i-1}) et :

$${}^{i-1}g = -[g_x, g_y, g_z, 0] \cdot \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = [-g \cdot o, g \cdot n, 0, 0] \quad (IV.26)$$

• **Si la liaison "i" est prismatique :**

${}^{i-1}\Delta_i$ correspond à une translation suivant l'axe (z) du repère (R_{i-1}) est :

$${}^{i-1}g = -[g_x, g_y, g_z, 0] \cdot \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [0, 0, 0, -g \cdot a] \quad (IV.27)$$

-D- Simplification des termes D_{ijk}

- Le calcul direct des termes D_{ijk} à partir de l'équation (IV.11.b) est très lourd à effectuer à cause des dérivées premières et secondes de la matrice de transformation T_p , c'est pour cela qu'on va essayer de donner une autre forme à ces termes ne comportant que des produits de matrices de transformation déjà calculés (jT_i).

On peut montrer que la dérivé d'une matrice de transformation homogène :

$${}^0T_i = A_1 \cdot A_2 \dots A_i ; (i=1, \dots, n)$$

peut s'écrire sous la forme suivante : [FU 87][GOR 84]

$$\frac{\partial {}^0T_i}{\partial q_j} = \begin{cases} T_{j-1} \cdot Q_j \cdot {}^{j-1}T_i & \text{si } (j \leq i) \\ 0 & \text{si } (j > i) \end{cases} \tag{IV.28}$$

Avec : $T_{j-1} = A_1 \cdot A_2 \dots A_{j-1}$

$${}^{j-1}T_i = A_j \cdot A_{j+1} \dots A_i$$

$$\text{et } Q_j = \bar{Q} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Si la liaison "i" est rotative .}$$

$$Q_j = Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{Si la liaison "i" est prismatique.}$$

• Démonstration :

$$T_i = T_i(q_1, q_2, \dots, q_i) = A_1(q_1) \cdot A_2(q_2) \cdot \dots \cdot A_i(q_i)$$

$$\text{D'où : } \frac{\partial T_i}{\partial q_j} = A_1(q_1) \cdot A_2(q_2) \cdot \dots \cdot \frac{\partial A_j}{\partial q_j} \cdot \dots \cdot A_i(q_i) \quad \text{Si } (j \leq i)$$

La matrice de D-H s'écrit :

$$A_j = \begin{pmatrix} C_j & -S_j \cos \alpha_j & S_j \sin \alpha_j & a_j C_j \\ S_j & C_j \cos \alpha_j & -C_j \sin \alpha_j & a_j S_j \\ 0 & \sin \alpha_j & \cos \alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{pmatrix} = \begin{bmatrix} D_{11} + I_{a1} & 0 & 0 \\ 0 & D_{22} + I_{a2} & D_{23} \\ 0 & D_{23} & D_{33} + I_{a3} \end{bmatrix} \cdot \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 & D_{122} & 0 \\ D_{211} & 0 & D_{233} \\ D_{311} & D_{322} & 0 \end{bmatrix} \cdot \begin{bmatrix} (\dot{q}_1)^2 \\ (\dot{q}_2)^2 \\ (\dot{q}_3)^2 \end{bmatrix} \\
 + 2 \cdot \begin{bmatrix} D_{122} & D_{133} & 0 \\ 0 & 0 & D_{223} \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_1 \dot{q}_2 \\ \dot{q}_1 \dot{q}_3 \\ \dot{q}_1 \dot{q}_3 \end{bmatrix} + \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & f_3 \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ D_1 \\ D_3 \end{bmatrix}$$

Chapitre V

SIMULATION ET COMMANDE

Chapitre V

Simulation et Commande

V.1 Introduction

Pour réaliser une tâche désirée dans l'espace cartésien (espace réel), le bras manipulateur doit suivre une trajectoire spécifiée (commande) pour chacune de ces variables articulaires. Le contrôle du robot consiste à faire en sorte que la configuration prise par le système soit asservie à ces commandes et ne dépende que de celles-ci.

Comme en général ce sont des actionneurs qui conditionnent les variables articulaires d'un robot, c'est par une action appropriée sur leur commande que le contrôle de la position de l'effecteur du robot est fait; La commande est donc réalisée dans l'espace des variables articulaires. Néanmoins, ils existent des techniques de contrôle en espace cartésien qu'on n'a pas envisagé.

Le modèle du robot manipulateur est un système couplé du second ordre et non linéaire. Ceci fait qu'au sens du système asservi, il s'agit d'un asservissement non linéaire multivariables : c'est un problème complexe.

Nous allons en premier lieu présenter les résultats de simulation du système en boucle ouverte pour voir son comportement naturel, nous allons ensuite linéariser le modèle du robot en vue de rendre son comportement identique à un système du second ordre classique. Nous allons exploiter les résultats obtenus pour une commande classique de type PID et discuter les problèmes liés au contrôle du robot. Ensuite, nous établirons une loi de commande basée sur la connaissance du modèle du robot (commande linéarisante directe et prédictive) et dégagerons enfin ces inconvénients .

V.2 Réponse en boucle ouverte

La simulation du modèle dynamique du bras manipulateur en boucle ouverte, face à des entrées de type échelon, permet d'estimer globalement les fonctions de transfert le caractérisant ainsi que sa stabilité, les interactions entre ses différents segments etc...

V.2.1 Procédure:

La mise sous forme d'équations d'états du modèle dynamique trouvé (équation IV.8), nous permet d'utiliser la méthode de RUNGE_ KUTTA d'ordre 4 pour résoudre le système.

Pour cela, on va écrire le modèle dynamique à partir de sa représentation matricielle (IV.9) sous la forme suivante:

$$\ddot{q} = H^{-1}(q) [T - h(q, \dot{q}) - g(q)] \quad (V.1)$$

Avec:

$$H^{-1}(q) = \frac{1}{\det} \begin{bmatrix} D_{22}D_{33} - D_{23}^2 & 0 & 0 \\ 0 & D_{11}D_{33} & -D_{11}D_{23} \\ 0 & -D_{11}D_{23} & D_{11}D_{22} \end{bmatrix}$$

$$T = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{bmatrix} \quad (V.2)$$

$$h(q, \dot{q}) = \begin{bmatrix} D_{122} \dot{q}_2^2 + D_{112} \dot{q}_1 \dot{q}_2 + D_{113} \dot{q}_1 \dot{q}_3 + f_1 \dot{q}_1 \\ D_{211} \dot{q}_1^2 + D_{233} \dot{q}_3^2 + D_{223} \dot{q}_2 \dot{q}_3 + f_2 \dot{q}_2 \\ D_{311} \dot{q}_1^2 + D_{322} \dot{q}_2^2 + f_3 \dot{q}_3 \end{bmatrix} \quad (V.3)$$

$$g(q) = \begin{bmatrix} 0 \\ D_2 \\ D_1 \end{bmatrix} \quad (V.4)$$

Avec :

$$\det = D_{11}(D_{22}D_{33} - D_{23}^2)$$

$$\Gamma_i = k_i u_i ; i = 1, 2, 3$$

k_i : Constante du couple moteur 'i'.

u_i : Signal de commande de l'actionneur 'i'.

L'écriture de (IV.9) sous forme d'équations d'états:

$$\begin{cases} \dot{X}(t) = A(X).X(t) + B(X).U + D(X). \\ Y(t) = C.X(t) \end{cases}$$

$$\text{Avec : } X^t = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6] = [q_1 \ \dot{q}_1 \ q_3 \ \dot{q}_2 \ q_5 \ \dot{q}_3]$$

$$U^t = [u_1 \ u_2 \ u_3] : \text{vecteur de commande.}$$

$$Y^t = [y_1 \ y_2 \ y_3] = [q_1 \ q_2 \ q_3] : \text{vecteur de sortie.}$$

tel que l'on ait initialement $X^t = [0 \ 0 \ 0 \ 0 \ 0 \ 0]$

donnera:

$$A(X) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{f_1}{D_{11}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{D_{33}f_2}{\alpha} & 0 & -\frac{D_{23}f_3}{\alpha} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{D_{23}f_2}{\alpha} & 0 & -\frac{D_{22}f_3}{\alpha} \end{bmatrix} \quad (V.5)$$

Avec $\alpha = \frac{\det}{D_{11}} = D_{22}D_{33} - D_{23}^2$.

$$B(X) = \begin{bmatrix} 0 & 0 & 0 \\ \frac{k_1}{D_{11}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{k_2 D_{33}}{\alpha} & -\frac{D_{23} k_3}{\alpha} \\ 0 & 0 & 0 \\ 0 & -D_{23} k_2 & D_{22} k_3 \end{bmatrix} \quad (V.6)$$

$$D(X) = \begin{bmatrix} 0 \\ D_2(X) \\ 0 \\ D_4(X) \\ 0 \\ D_6(X) \end{bmatrix} \quad (V.7)$$

Avec :

$$\begin{cases} D_2(X) = -\frac{1}{D_{11}} \{ D_{122} x_4^2 + 2D_{112} x_2 x_4 + 2D_{113} x_2 x_6 \} \\ D_4(X) = \frac{1}{\alpha} \{ D_{22} D_{311} - D_{33} D_{211} \} x_2^2 + D_{22} D_{322} x_4^2 - D_{33} D_{233} x_6^2 - 2D_{33} D_{233} x_4 x_6 - D_{33} D_2 + D_{23} D_3 \} \\ D_6(X) = \frac{1}{\alpha} \{ D_{22} D_{211} - D_{22} D_{311} \} x_2^2 - D_{22} D_{322} x_4^2 + D_{23} D_{233} x_6^2 + 2D_{23} D_{223} x_4 x_6 + D_{23} D_2 - D_{22} D_3 \} \end{cases}$$

et enfin :

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (\text{V.8})$$

C'est sous cette forme que le système dynamique sera numériquement simulé.

Les valeurs numériques des différents paramètres du robot sont [STO89]:

$$d_1 = a_2 = a_3 = 1\text{m}$$

$$m_1 = m_2 = m_3 = 10\text{Kgs}$$

$$f_1 = 75\text{N}\cdot\text{Sec}^{-1}; f_2 = 10\text{N}\cdot\text{Sec}^{-1}; f_3 = 1\text{N}\cdot\text{Sec}^{-1}$$

$$k_1 = 40\text{Nm/V}; k_2 = 20\text{Nm/V}; k_3 = 40\text{Nm/V}..$$

$$\text{avec } g=9.81\text{N}\cdot\text{Kg}^{-1}$$

Les réponses du robot articulé en position et en vitesse pour une entrée échelon sur un intervalle de temps de 10 secondes sont présentées par les figures suivantes:

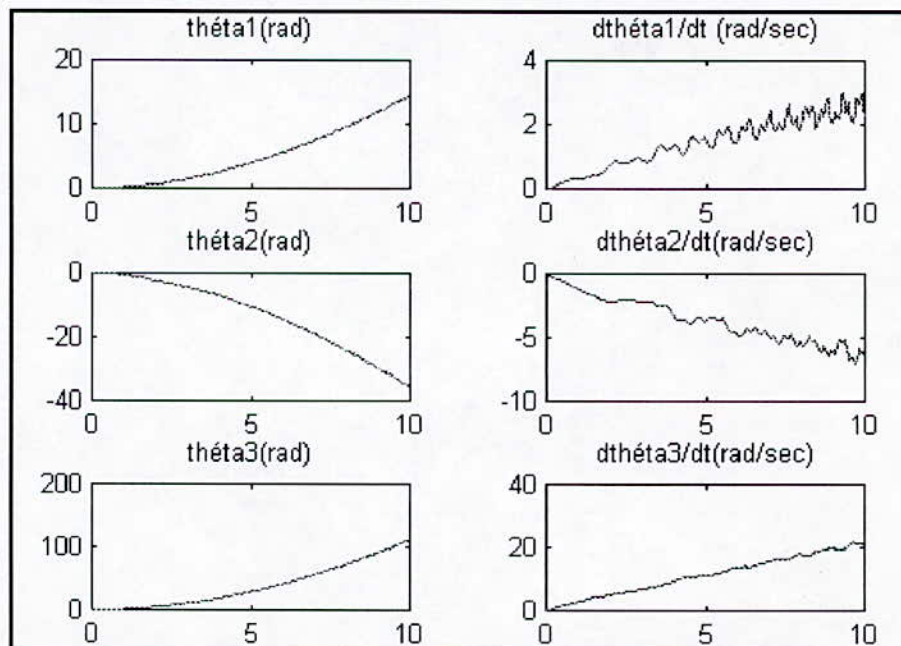
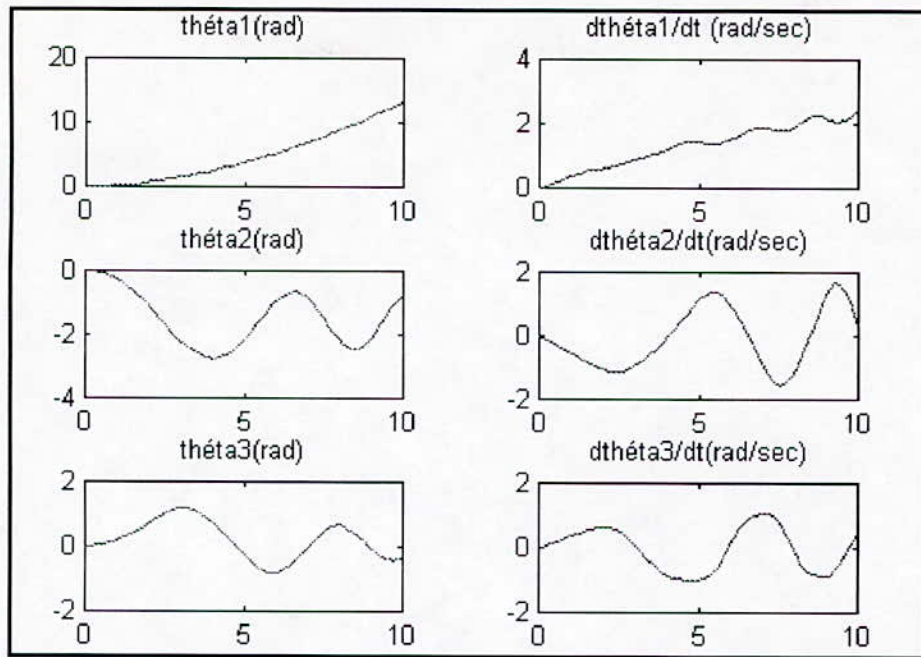


Figure V.1 : Réponse indicielle du bras pour $u_1 = u_2 = u_3 = 10$



FigureV.2 : Réponse indicielle du bras pour $u_1=10, u_2=u_3=0$.

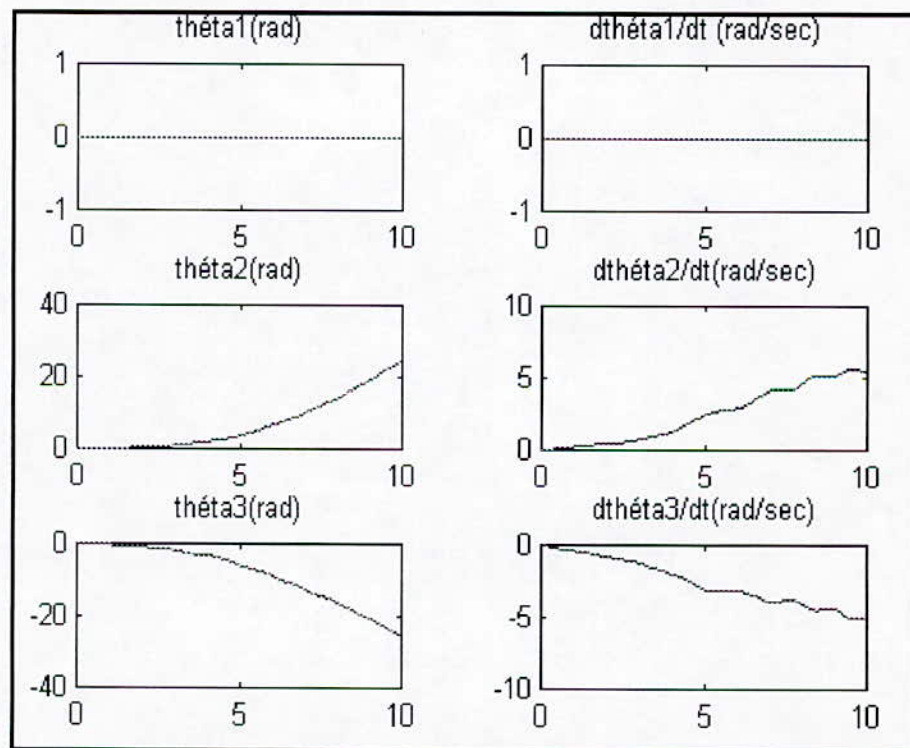


Figure V.3 : Réponse indicielle du bras pour $u_1=u_3=0; u_2=10$.

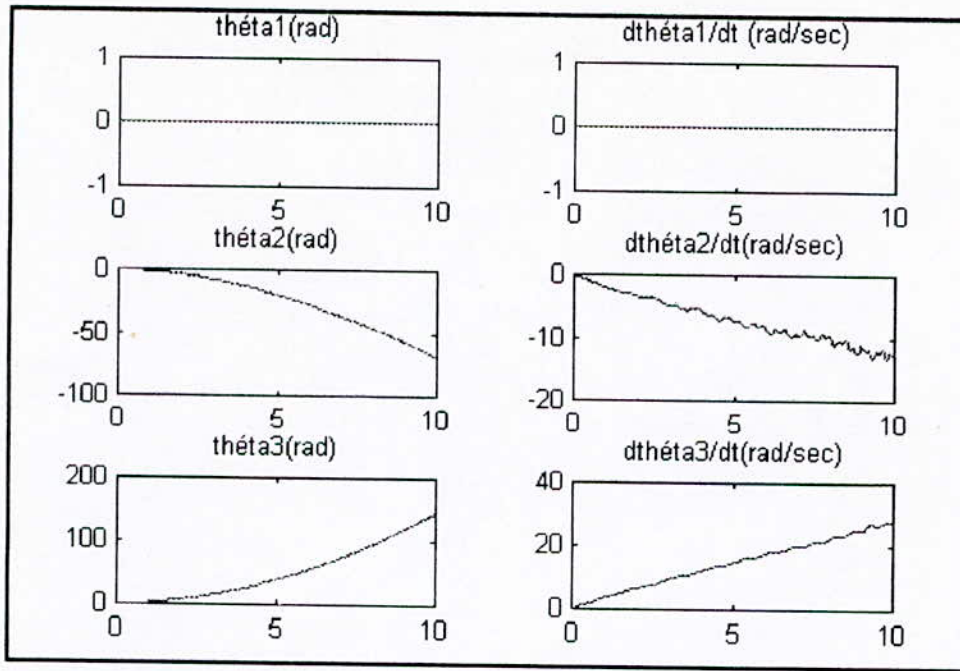


Figure V.4 : Réponse indicielle du bras pour $u_1=u_2=0; u_3=10$.

V.2.2 Analyse des résultats :

Il est clair d'après la figure (V.1) que le comportement du robot est très semblable à celui d'un double intégrateur (les variables articulaires décrivent une parabole). Ainsi c'est les termes D_{ij} qui sont prédominants. Afin de mesurer le degré de couplage entre articulations, on a attaqué une articulation par un échelon alors que les autres sont libres. La figure V.2 montre le couplage entre la première articulation et les deux dernières. Il s'agit d'un couplage assez faible vu la dynamique de θ_1 et de θ_2 .

La figure (V.3) montre l'effet du deuxième segment sur les autres. Alors que le premier segment reste immobile (prévisible), le troisième segment décrit presque le même mouvement que le deuxième mais dans le sens contraire. On peut interpréter cela en disant que le manipulateur tend à revenir à sa position de départ (principe des moindres actions).

La figure (V.4) montre l'effet du deuxième segment sur les autres. Alors que le premier reste immobile comme avant, le deuxième segment s'oppose au mouvement du troisième.

On peut conclure que le robot est instable en boucle ouverte, d'où la nécessité d'une commande en boucle fermée. En outre, il apparaît clairement que le robot est un système fortement couplé.

V.3 Implémentation d'une commande classique de type PID

V.3.1 Introduction

Améliorer les performances d'un système asservis revient à trouver un compromis entre les conditions qui garantissent une bonne stabilité et celles qui garantissent une bonne précision en régime établi. C'est pour cela qu'on utilise des correcteurs qui sont des systèmes linéaires, leur fonction de transfert permet de modifier convenablement la transmittance du système à asservir.

Parmi les correcteurs classiques, on distingue:

- Les correcteurs à avance de phase: PD.
- Les correcteurs à retard de phase : PI.
- Les correcteurs à avance et à retard de phase : PID.

L'idée la plus simple qui vient à l'esprit pour contrôler un robot consiste à réaliser un asservissement classique(Figure V.6).

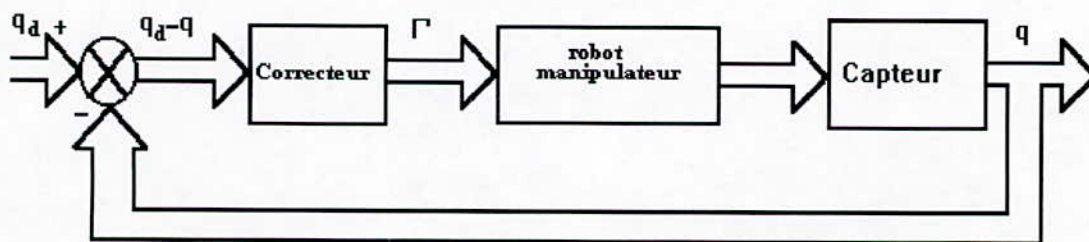


Figure V.6 Schéma bloc d'une commande classique.

Le correcteur qui intervient dans cet asservissement peut être un PI, PD ou PID. L'étude du fonctionnement du système en boucle fermée dépend essentiellement de l'actionneur, du robot manipulateur et des capteurs. Or du fait des hypothèses énoncées auparavant, le bloc dominant reste le modèle dynamique du bras manipulateur qui est en général non linéaire et couplé. L'étude de l'asservissement classique ne peut donc se faire en utilisant la notion de la transmittance. Cependant pour fixer les idées, et montrer les défauts d'un tel système de commande, nous allons dans un premier temps simplifier le modèle du robot afin de le linéariser et de le découpler. Ceci va conduire à contrôler chaque axe séparément des autres et va nous permettre d'étudier le système de point de vue de l'automatique classique. Cette procédure est très utilisée dans l'industrie robotique probablement pour des raisons économiques[CRA 86].

V.3.2-Commande axe par axe :

Dans ce type de commande, les simplifications effectuées sur le modèle dynamique du robot (V.8) vont introduire un comportement séparé de chaque axe qui sera modélisé par :

$$\Gamma_i = k_i u_i = \overline{D_{ii}} \ddot{q}_i + f_i \dot{q}_i + \Gamma_{pi} \quad (i=1,2,\dots,n) \tag{V.9}$$

Où : $\overline{D_{ii}}$ est une majoration du terme D_{ii} de la matrice d'inertie A.

f_i : Coefficient de frottement visqueux.

Γ_{pi} : Représente les perturbations dues à la charge et aux efforts dynamiques non pris en compte lors de la modélisation.

On a en effet négligé les termes non diagonaux de la matrice d'inertie et les effets centrifuges et celle de Coriolis.

♦ **Remarque:** Il faut noter qu'il existe aussi plusieurs méthodes de simplification du modèle, comme celle qui néglige les effets du frottement visqueux à cause de leur modélisation difficile et prend une moyenne des termes D_{ii} autour du centre du travail du robot [CRA 86].

Chaque segment est donc considéré indépendant des autres et est régi par une équation différentielle du second ordre à coefficients constants, il s'agit du second ordre classique en asservissements linéaires.

Le schéma d'une telle commande est le suivant:

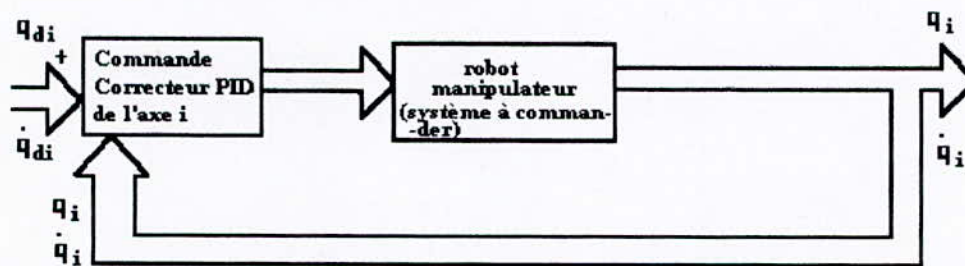


Figure V.7 : Schéma de commande classique axe par axe.

La commande élaborée par le correcteur PID est de la forme :

$$\Gamma_i(t) = k_{pi}(q_{id}(t) - q_i(t)) + k_{vi}(\dot{q}_{id}(t) - \dot{q}_i(t)) + k_{ii} \int_0^t (q_{id}(\xi) - q_i(\xi)) d\xi \tag{V.10}$$

Où : q_{id}, \dot{q}_{id} sont la position et la vitesse articulaires désirées.

q_i, \dot{q}_i sont la position et la vitesse prise par le bras manipulateur.

Si la commande est simplement PD, ce qui souvent le cas, égalons (V.9) et (V.10) avec k_{ii} nulles pour tout i . On obtient l'équation du second ordre suivante:

$$q_i + \frac{2\xi_i}{\omega_{ni}} \dot{q}_i + \frac{1}{\omega_{ni}^2} \ddot{q}_i = q_{id} + \frac{k_{vi}}{k_{pi}} \dot{q}_{id} - \frac{\Gamma_{pi}}{k_{pi}} \tag{V.11}$$

Où on a introduit le coefficient d'amortissement réduit ξ_i et la pulsation propre ω_{ni} liés à k_{pi} et k_{vi} par:

$$\omega_{ni} = \sqrt{\frac{k_{pi}}{D_{ii}}} \text{ et } \xi_i = \frac{1}{2} \frac{k_{vi} + f_i}{\sqrt{k_{pi} \cdot D_{ii}}} \quad (V.12)$$

En pratique, les sur-oscillations et les dépassements dans la réponse indicielle ne sont pas désirables; L'exemple le plus convaincant peut être est lorsqu'un bras manipulateur doit déposer une pièce sur une surface donnée, toute oscillation en fin de chemin conduit à percuter la pièce avec la surface en raison. Donc la condition à prendre en considération est $\xi_i \geq 1$; mais comme on exige un temps de réponse minimale, il faut prendre $\xi_i = 1$.

Toute structure rigide est caractérisée par une certaine fréquence de résonance propre $\omega_{structurel}$, et elle peut entrer en vibration si elle est excitée par cette pulsation.

Pour éviter de déstabiliser le système, on doit limiter ω_n à $0.5\omega_{structurel}$ [PAU 81].

◆ **Remarque:**

En pratique, l'asservissement des systèmes est échantillonné; l'équation(V.10) s'écrit alors sous une forme discrète. En effet si l'on pose :

$$e_i(k\Delta t) = q_{id}(k\Delta t) - q_i(k\Delta t) \quad (V.13)$$

$$\dot{e}_i(k\Delta t) = \dot{q}_{id}(k\Delta t) - \dot{q}_i(k\Delta t) \quad (V.14)$$

$t = k\Delta t$; Δt est la période d'échantillonnage.

On aura :

$$\Gamma_i(k\Delta t) = k_{pi}e_i(k\Delta t) + k_{vi}\dot{e}_i(k\Delta t) + k_{ii}\Delta t \sum_{i=0}^k e_i(i\Delta t) \quad (V.15)$$

Si l'on ne dispose pas de capteurs de vitesse, $\dot{q}_i(t)$ n'est pas disponible et on peut écrire $e_i(k\Delta t)$ sous la forme :

$$\dot{e}_i(k\Delta t) = \frac{1}{\Delta t} (e_i(k\Delta t) - e_i((k-1)\Delta t)) \quad (V.16)$$

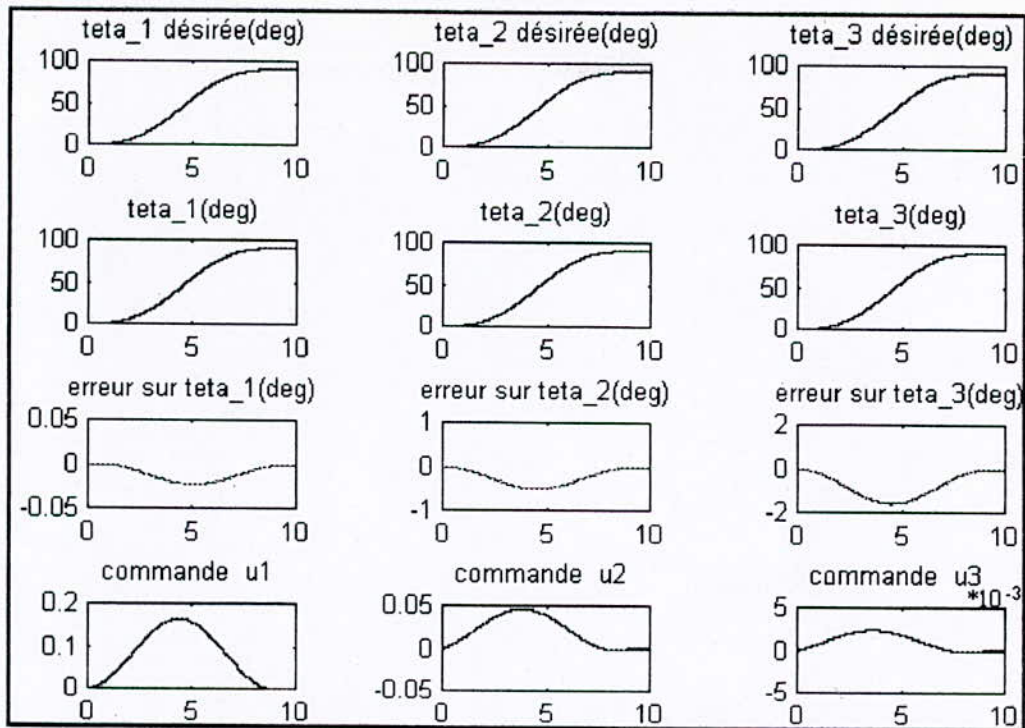


Figure V.8 : Réponse du bras pour une commande axe par axe ($\xi_i=1; i=1,2,3$)

Ce type de commande est satisfaisant si les deux conditions suivantes sont remplies:

1-Les gains k_{p_i} ($i=1,2,3$) sont assez élevés. En effet des gains assez élevés entraînent la minimisation de l'effet des perturbations Γ_{p_i} dans (V.11). Cela signifie que les effets de couplage dynamique des termes dynamiques (éléments non diagonaux de la matrice $H(q)$), ainsi que les effets centrifuges et de Coriolis sur l'articulation 'i' demeurent dans une limite raisonnable. Ce sera le cas si les mouvements ne sont pas trop rapides et sont de faible amplitude.

2-Le terme $D_{ii}(q)$ ne s'éloigne pas de trop de sa borne supérieure $\overline{D_{ii}}$, valeur avec laquelle on aura réglé les gains de l'asservissement pour avoir un bon comportement. En pratique c'est le cas pour des actionneurs qui ont un grand rapport de réduction.

V.3.4 Implantation d'une loi de commande PID :

Lorsque le mouvement de l'effecteur doit être très rapide le long d'une trajectoire donnée, la commande classique n'est plus suffisante pour assurer un bon suivi de la trajectoire.

Il est alors nécessaire d'envisager la prise en compte du modèle complet du manipulateur (IV.12):

$$\Gamma = H(q)\ddot{q} + h(q, \dot{q}) + g(q) + f(\dot{q}) \quad (V.17)$$

Le schéma de commande doit maintenant considérer l'ensemble des actionneurs. Le système doit être envisagé comme un système multivariable.

Comme précédemment le couple de commande est:

$$\Gamma = k_v \dot{e}(t) + k_p e(t) + k_i \int_0^t e(\xi) d\xi \quad (\text{V.18})$$

Où k_p , k_v et k_i sont des matrices diagonales constantes, des gains du correcteur PID.

$e(t)$ et $\dot{e}(t)$ sont respectivement le vecteur d'erreur en variables articulaires et en vitesses articulaires.

Ce type de contrôle est simple, car chaque articulation du robot est commandée séparément des autres.

Les performances d'une telle loi de commande appliquée à un système non linéaire couplé, sont très difficiles à décrire (comme l'erreur de suivi). En outre, le choix convenable des matrices k_p , k_v et k_i est impossible à faire pour toutes les configurations du bras et pour des trajectoires variées. Donc le choix de ces gains reste *arbitraire*.

Dans notre cas, on a exploité les résultats de la commande précédente comme valeurs initiales, d'où apparaît l'intérêt de la commande classique axe par axe qui permet d'estimer globalement un ordre de grandeur des gains des correcteurs. Ainsi, la plage de variations de ces gains appliquées au correcteur PID sera considérablement réduite.

Résultats et interprétation

Les valeurs des gains obtenues pour la moindre erreur sont :

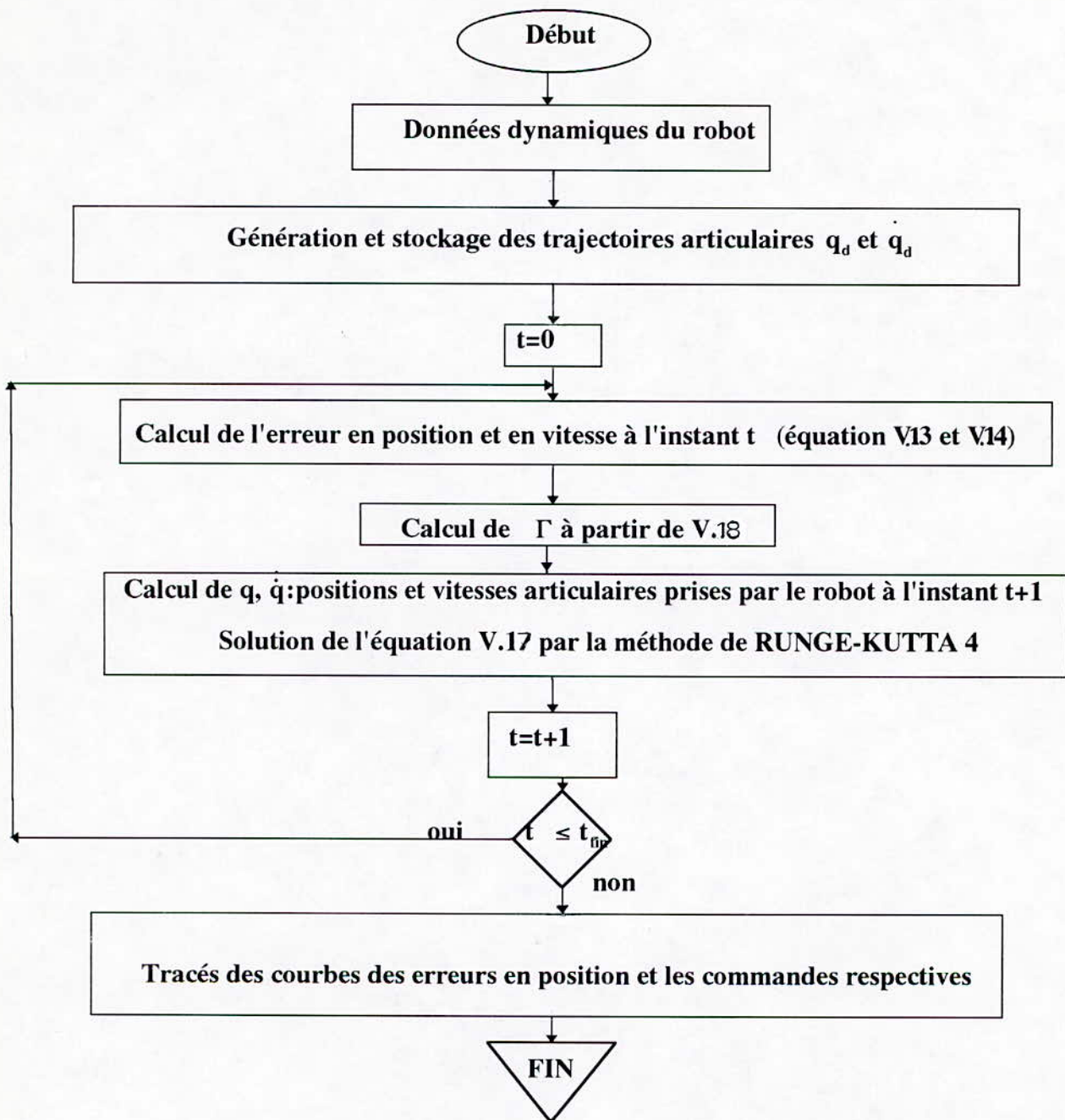
$$k_{p1} = 2670 ; k_{v1} = 30 ; k_{i1} = 5;$$

$$k_{p2} = 2670 ; k_{v2} = 120 ; k_{i2} = 5;$$

$$k_{p3} = 333 ; k_{v3} = 80 ; k_{i3} = 5;$$

Les résultats sont présentés par la figure (V.9).

Algorithme de commande pour une commande PID



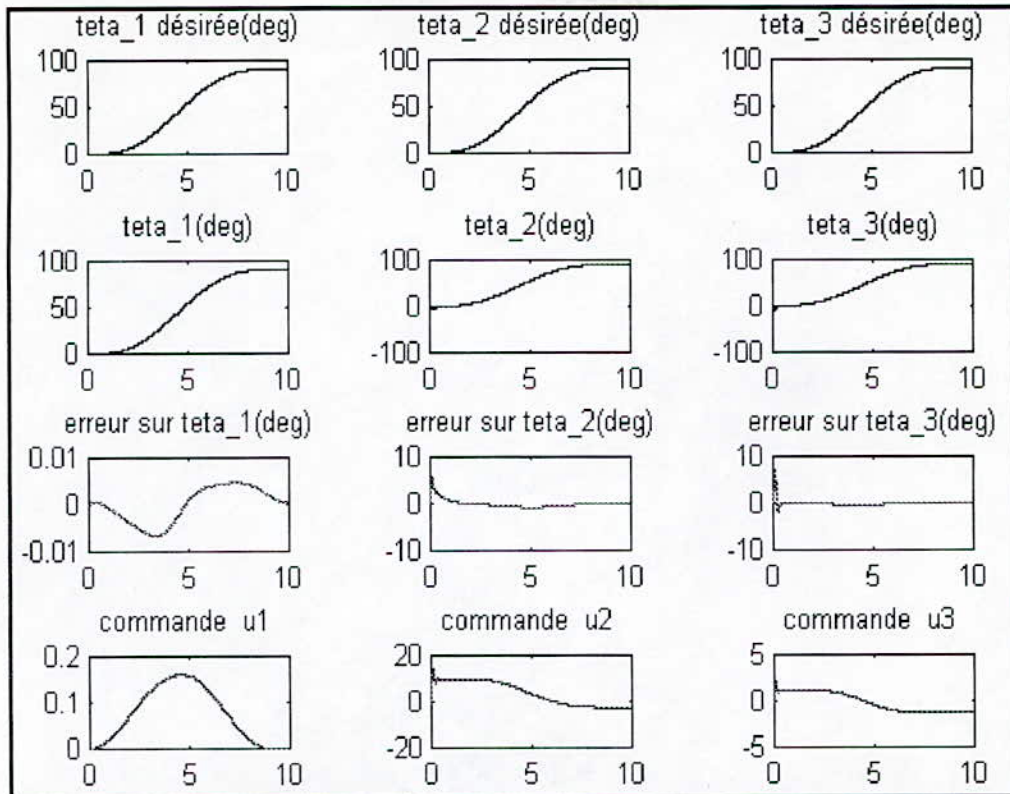


Figure V.9 : Réponse du bras pour une loi PID classique.

Effet du terme intégrateur k_i

En comparant les équations (V.17) et (V.18), on risque d'avoir des erreurs statiques finales de position pour les variables articulaires. Le choix d'une valeur assez faible de k_i permet d'éliminer ces erreurs finales sans pour autant s'éloigner du système de second ordre.

Dans notre cas $k_i = 5$ pour toutes les articulations.

V.4 Commande linéarisante [CRA 86]

Dans le cas où les paramètres dynamiques du bras manipulateur sont connus, le modèle du robot s'écrira sous la forme de l'équation différentielle suivante:

$$H(q)\ddot{q} + h(q, \dot{q}) + g(q) + f(\dot{q}) = \Gamma \tag{V.19}$$

Où H, h, g et f sont définis dans (IV.12) et sont pratiquement connues.

La boucle de linéarisation est réalisée par le choix d'un couple Γ à appliquer au robot de la forme suivante [CRA 86] :

$$\Gamma = \alpha\Gamma' + \beta \tag{V.20}$$

avec $\alpha = H_0(q)$ et $\beta = h_0(q, \dot{q}) + g_0(q) + f_0(q)$, Où H_0, h_0, g_0, f_0 sont les estimés de H, h, g, f .

La combinaison des équations (V.19) et (V.20) donne :

$$H(q)\ddot{q} + h(q, \dot{q}) + g(q) + f(q) = H_0(q)\Gamma' + h_0(q, \dot{q}) + g_0(q) + f_0(q) \quad (V.21)$$

Dans le cas où le modèle du robot est connu avec une précision suffisante, on aura:

$$H(q) \approx H_0(q) ; g(q) \approx g_0(q)$$

$$h(q, \dot{q}) \approx h_0(q, \dot{q}) ; f(q) \approx f_0(q)$$

L'équation (V.21) devient alors:

$$\ddot{q} = \Gamma' \quad (V.22)$$

Car la matrice H^{-1} est non singulière.

Ainsi, le problème de commande du robot se trouve transformé en celui de n doublets d'intégrateurs découplés. L'entrée Γ' peut être interprétée comme une commande de la boucle externe qui devra asservir le système. Cette commande est spécifiée en terme d'accélération, elle peut être obtenue par compensation PD.

$$\Gamma' = \ddot{q}_d - k_v \dot{e}(t) - k_p e(t) \quad (V.23)$$

$k_v, k_p, e(t)$ et $\dot{e}(t)$ sont définis dans (V.18).

Les matrices des gains de retour k_p et k_v sont diagonales définies positives et peuvent être choisies de manière à imposer une dynamique de rejection de l'erreur de suivi des trajectoires désirées. On peut imposer au système de se comporter en boucle fermée comme un système multivariable de second ordre avec un amortissement ξ et une pulsation ω_0 . Ceci peut être réalisé par:

$$k_{vi} = 2\xi_i \omega_{0i} \quad \text{et} \quad k_{pi} = \omega_{0i}^2$$

En remplaçant l'expression de Γ' (V.23) de l'équation (V.22), on obtient l'équation de l'erreur en boucle fermée suivante :

$$\ddot{e}(t) + k_v \dot{e}(t) + k_p e(t) = 0 \quad (V.24)$$

L'équation (V.24) a pour solution un signal $e(t)$ qui tend exponentiellement vers zéro. On peut donc conclure que le système en boucle fermée, avec cette commande et dans le cas où les paramètres dynamiques du système sont bien connus, est exponentiellement stable.

L'architecture d'une telle loi de commande est représentée sur la figure (V.12).

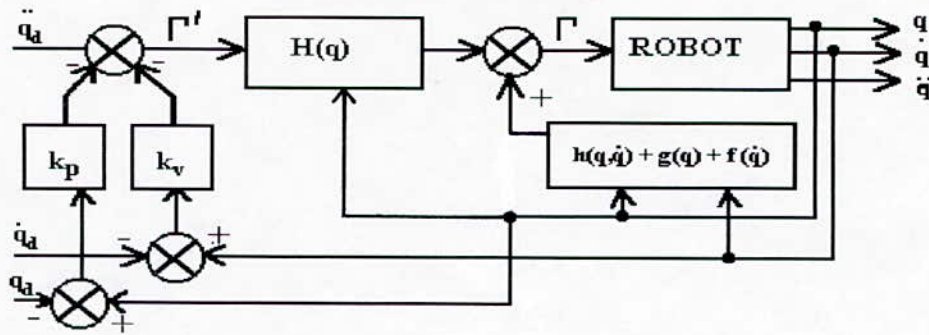
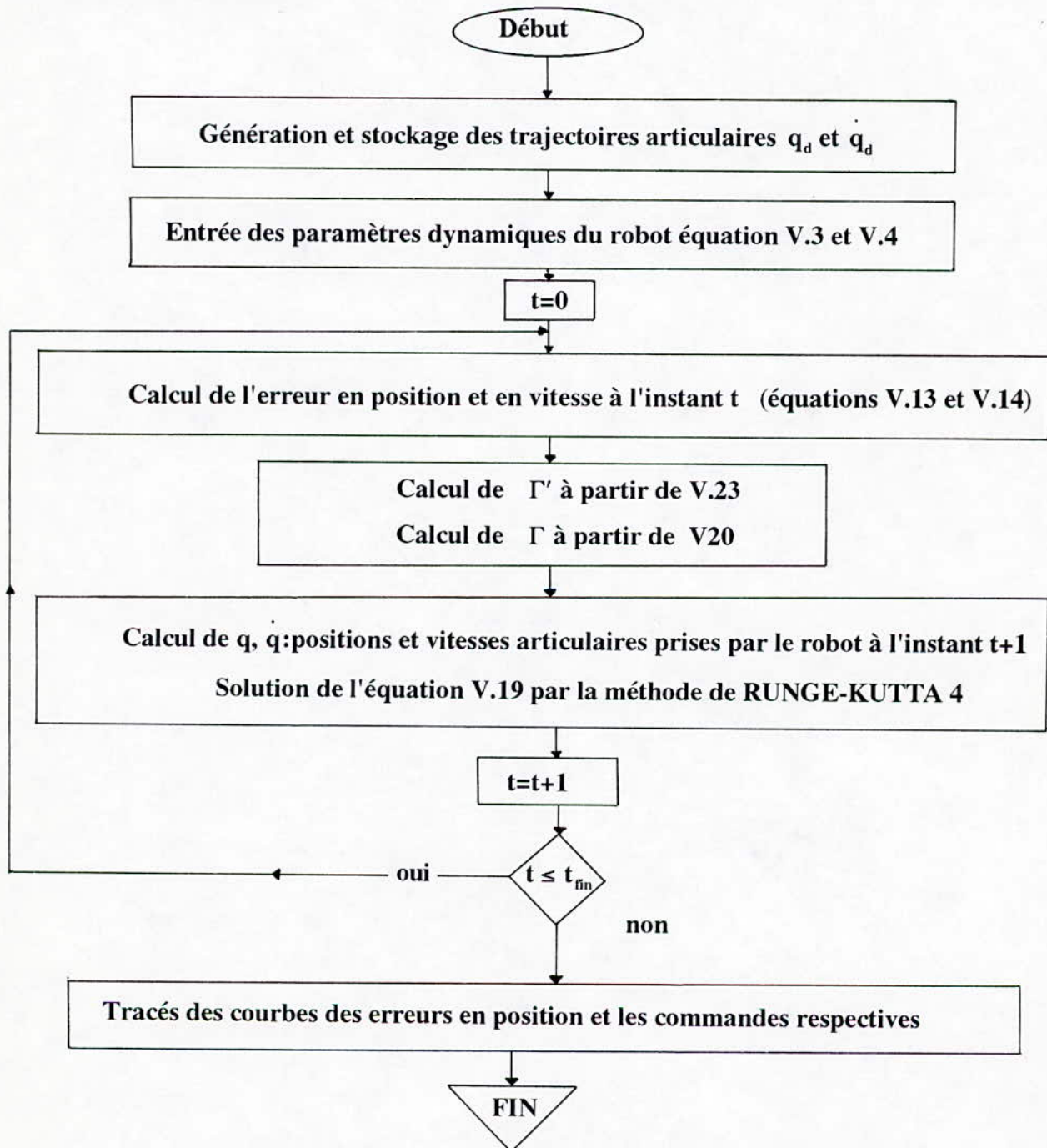


Figure V.12 Architecture d'une loi de commande PD linéarisante

Algorithme de commande :



Résultats et interprétations

pour une consigne indicielle, on vérifie que l'erreur satisfait bien l'équation d'erreur (V.24) que nous écrivons :

$$\ddot{e}(t) + 2\xi\omega_n \dot{e}(t) + \omega_n^2 e(t) = 0 \text{ avec } \omega_n = \sqrt{k_p} \text{ et } \xi = \frac{k_v}{2\sqrt{k_p}}.$$

Notons que du fait que le système est découplé (chose que nous avons vérifiée en simulation), on peut présenter les résultats en une seule simulation en prenant des valeurs différentes de ξ et de ω_n pour les trois articulations:

nous avons fait le choix suivant :

1- $\xi=1$ et $\omega_n=10$ c'est à dire $k_{p1}=100$ et $k_{v1}=20$ pour la première articulation.

2- $\xi=0.5$ et $\omega_n=10$ c'est à dire $k_{p2}=100$ et $k_{v2}=10$ pour la deuxième articulation.

3- $\xi=1.5$ et $\omega_n=10$ c'est à dire $k_{p3}=100$ et $k_{v3}=30$ pour la troisième articulation.

Les résultats sont donnés par la figure suivante:

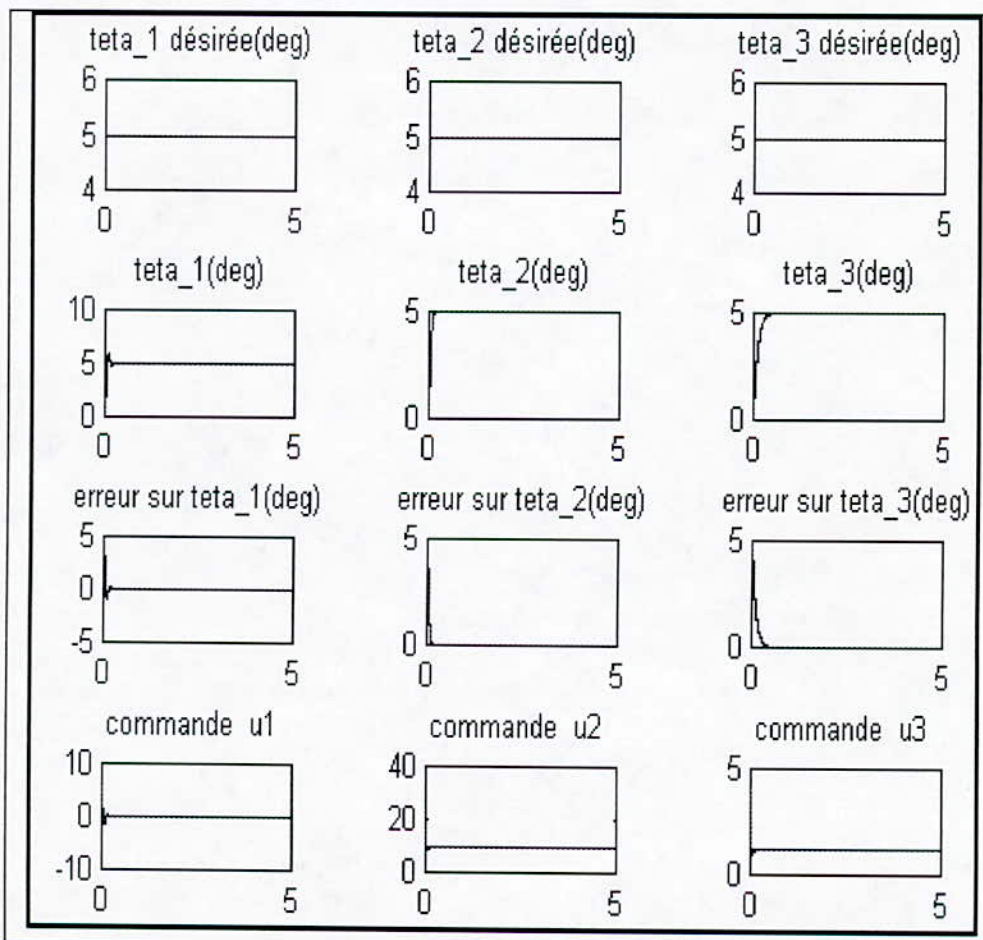


Figure V.13. Réponse indicielle d'un bras régi par une commande linéarisante

On voit bien que la réponse de la première articulation est oscillatoire($\xi=0.5$), celle de la deuxième est non oscillatoire et très rapide($\xi=1$) alors que celle de la troisième est très amortie($\xi=2$). Ceci est en parfait accord avec la théorie.

En pratique, on ne dispose que d'une estimation plus au moins précise des paramètres du robot tels que les termes d'inertie, de Coriolis, centrifuges et de frottement. On utilise dans ce cas le modèle

estimé, la commande est [GUE 95]: $\Gamma = \hat{H}(q)\Gamma' + \hat{h}(q, \dot{q}) + \hat{g}(q) + \hat{f}(\dot{q})$

L'application de cette commande, au robot défini par (V.19) donne :

$$\hat{H}(q)\Gamma' + \hat{h}(q, \dot{q}) + \hat{g}(q) + \hat{f}(\dot{q}) = H(q)\ddot{q} + h(q, \dot{q}) + g(q) + f(\dot{q}) + \hat{H}(q)\ddot{q} - \hat{H}(q)\ddot{q}$$

D'où en utilisant (V.13 et V.14), on obtient :

$$\ddot{e}(t) + k_v \dot{e}(t) + k_p e(t) = -\hat{H}^{-1}(q) \left(\Delta H \ddot{q} + \Delta h + \Delta g + \Delta f \right)$$

$$\text{Avec : } \begin{cases} \Delta H = H(q) - \hat{H}(q) \\ \Delta h = h(q, \dot{q}) - \hat{h}(q, \dot{q}) \\ \Delta g = g(q) - \hat{g}(q) \\ \Delta f = f(\dot{q}) - \hat{f}(\dot{q}) \end{cases}$$

Si les solutions sont précises alors l'erreur de suivi s'annulera. Par contre, dans le cas d'une erreur de modélisation, une excitation interne non linéaire(dynamique générée par l'erreur paramétrique), se propage dans le système en boucle fermée en propageant une erreur de suivi de trajectoire.

Par curiosité, nous avons simulé l'effet d'une mauvaise connaissance du système par un changement de 20% des paramètres du système lors du calcul de Γ . Le résultat est donné par la figure suivante pour les paramètres suivants:

$$k_{p_i} = 100, k_{v_i} = 20 \quad i=1,2,3.$$

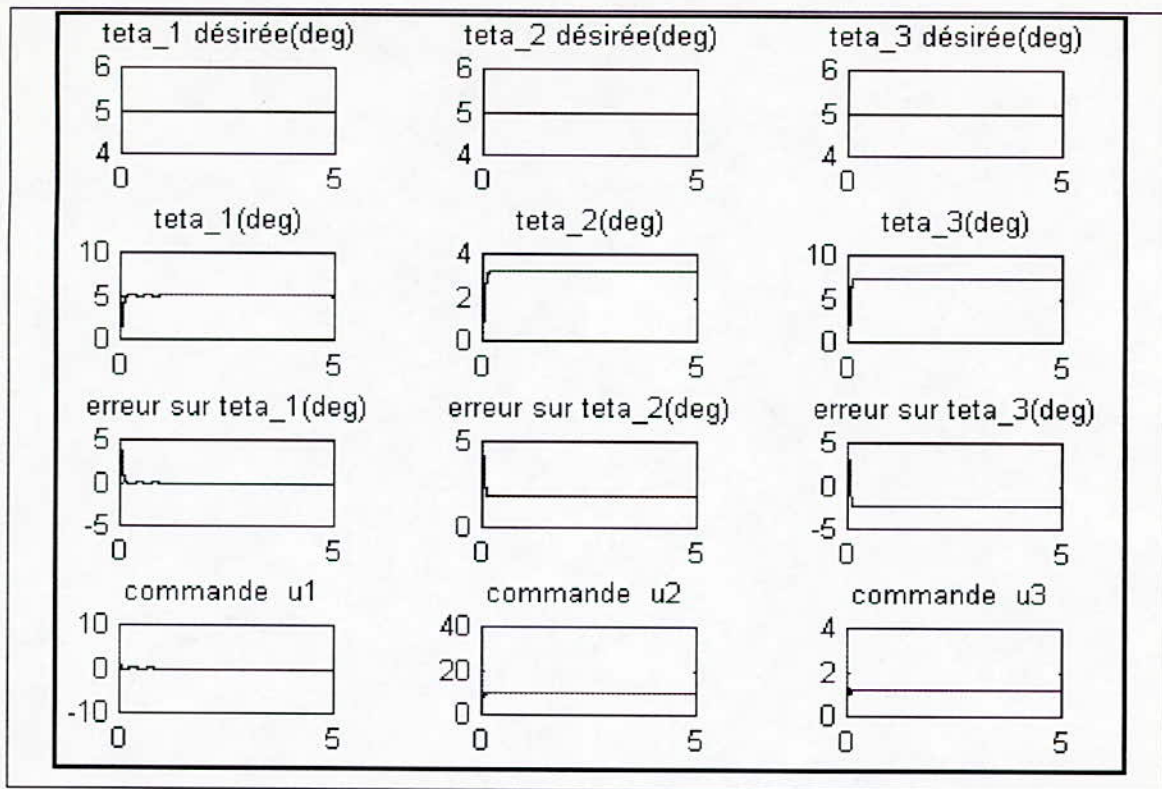


Figure V.13 : Réponse indicielle du bras RRR pour une modélisation à 20% près de ces paramètres.

On remarque que la réponse possède une erreur finale et ce pour la deuxième et la troisième articulation. On voit ainsi que cette méthode exige une connaissance presque parfaite du système, chose qui n'est pas souvent vraie en réalité.

On a ajouté une petite action intégrale avec une constante $k_i=1$ pour toutes articulations.

Le résultat est que les erreurs finales ont disparu (figure V.14).

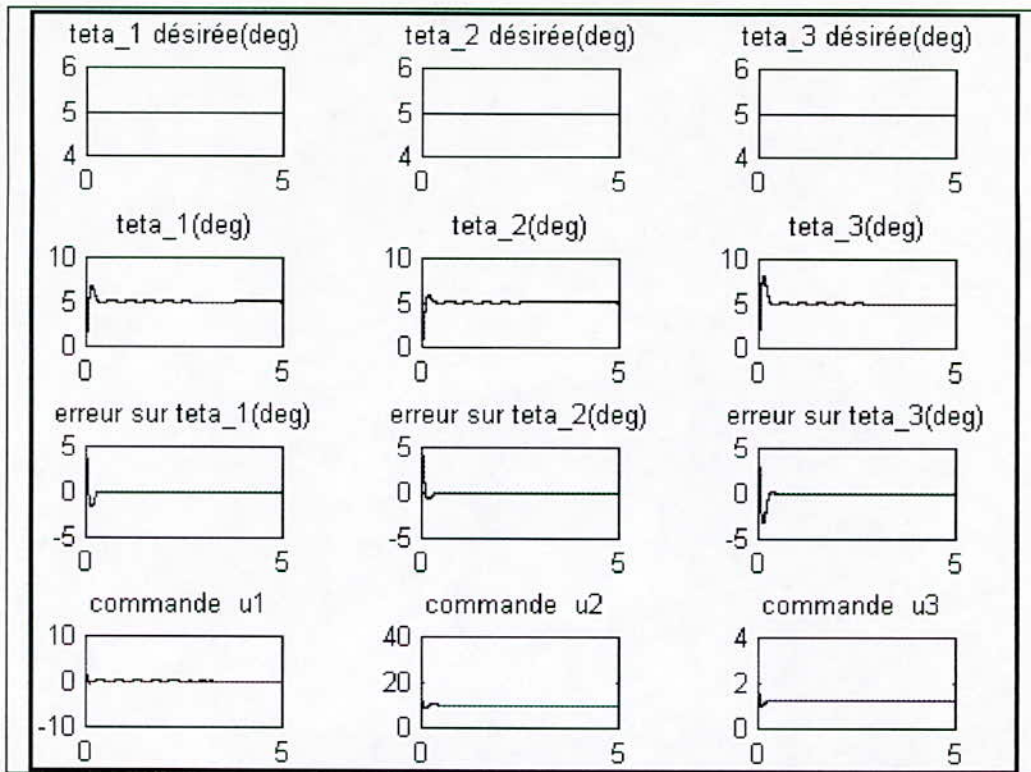


Figure V.14 : Réponse du bras RRR modélisé à 20% avec une action intégrale.

La figure V.15 représente la réponse du bras pour une consigne cycloïdale entre les configurations $[0, 0, 0]$ et $[\pi/4, \pi/4, \pi/4]$ pour $k_{vi}=20$ et $k_{pi}=100$, $i=1,2,3$.

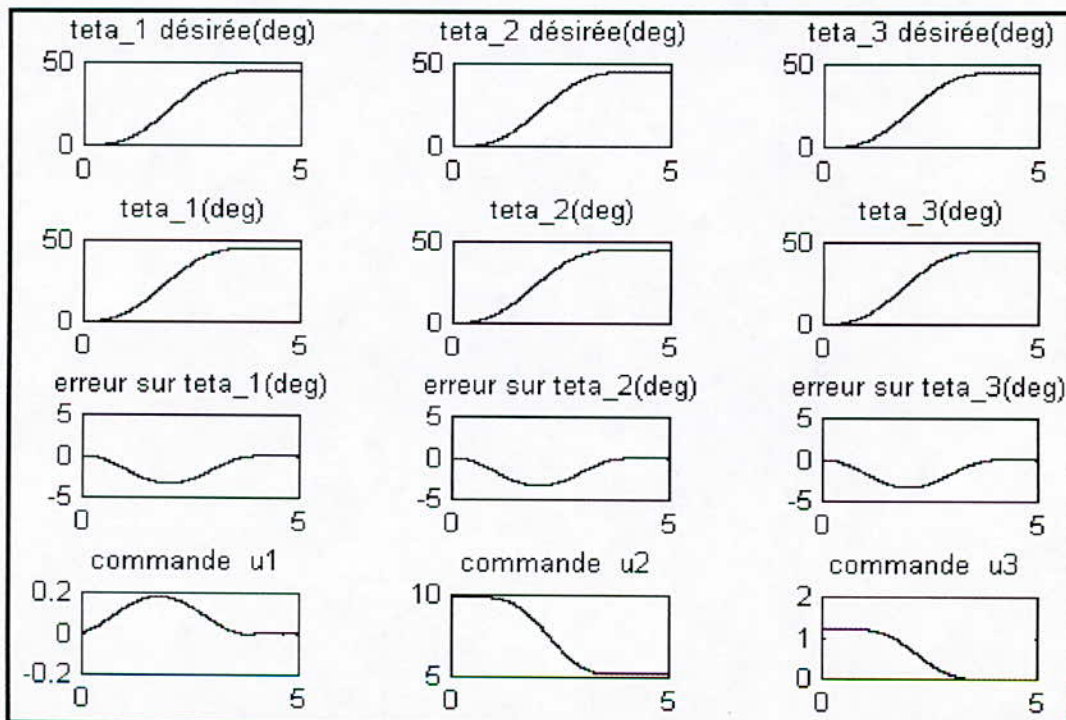


Figure V.15 : Réponse d'un bras RRR pour une trajectoire cycloïdale (commande linéarisante)

V.5 Commande linéarisante prédictive [CRA 86][GUE 95][LALL 94]

L'implémentation pratique de la commande linéarisante nécessite beaucoup de temps de calcul pour les termes non linéaire qui sont fonctions des positions et vitesses mesurées à chaque instant. Pour réduire cette complexité de calcul, un autre schéma de calcul peut être utilisé.

La commande prédictive peut être obtenue à partir de la commande linéarisante en remplaçant les variables $q(t)$ et $\dot{q}(t)$ par celles des trajectoires désirées $q_d(t)$ et $\dot{q}_d(t)$.

Ces dernières peuvent être connues à l'avance, par conséquent tous les calculs leurs correspondants peuvent être fait *hors ligne*, ce qui réduit la complexité des calculs nécessaires en ligne ou en temps réel. Les termes calculés hors ligne sont $H(q_d)$ et $h(q, \dot{q}_d) + g(q_d) + f(\dot{q}_d)$.

L'expression de la commande devient [CRA 86]:

$$\Gamma = H(q_d)\Gamma' + h(q, \dot{q}_d) + g(q_d) + f(\dot{q}_d) \quad (\text{V.25})$$

$$\text{avec } \Gamma' = \ddot{q}_d - k_v \dot{e}(t) - k_p e(t)$$

Ces données sont stockées en mémoire pour être utilisées lors de la commande. Le schéma de la commande ainsi obtenue est donné par la figure suivante :

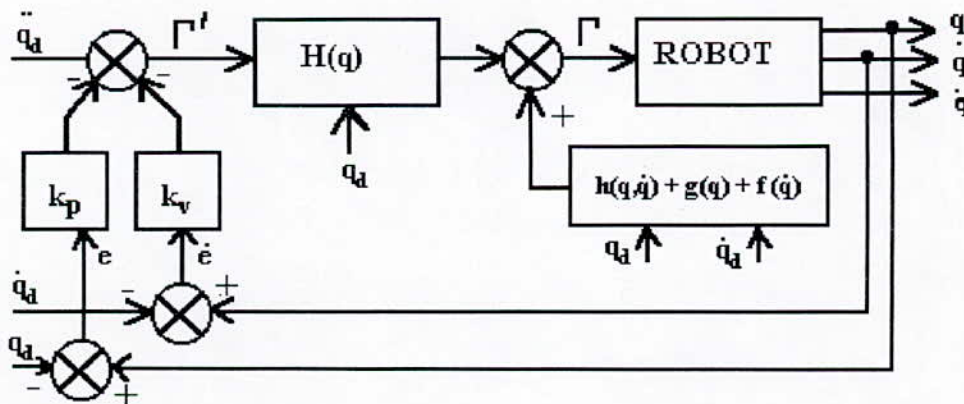
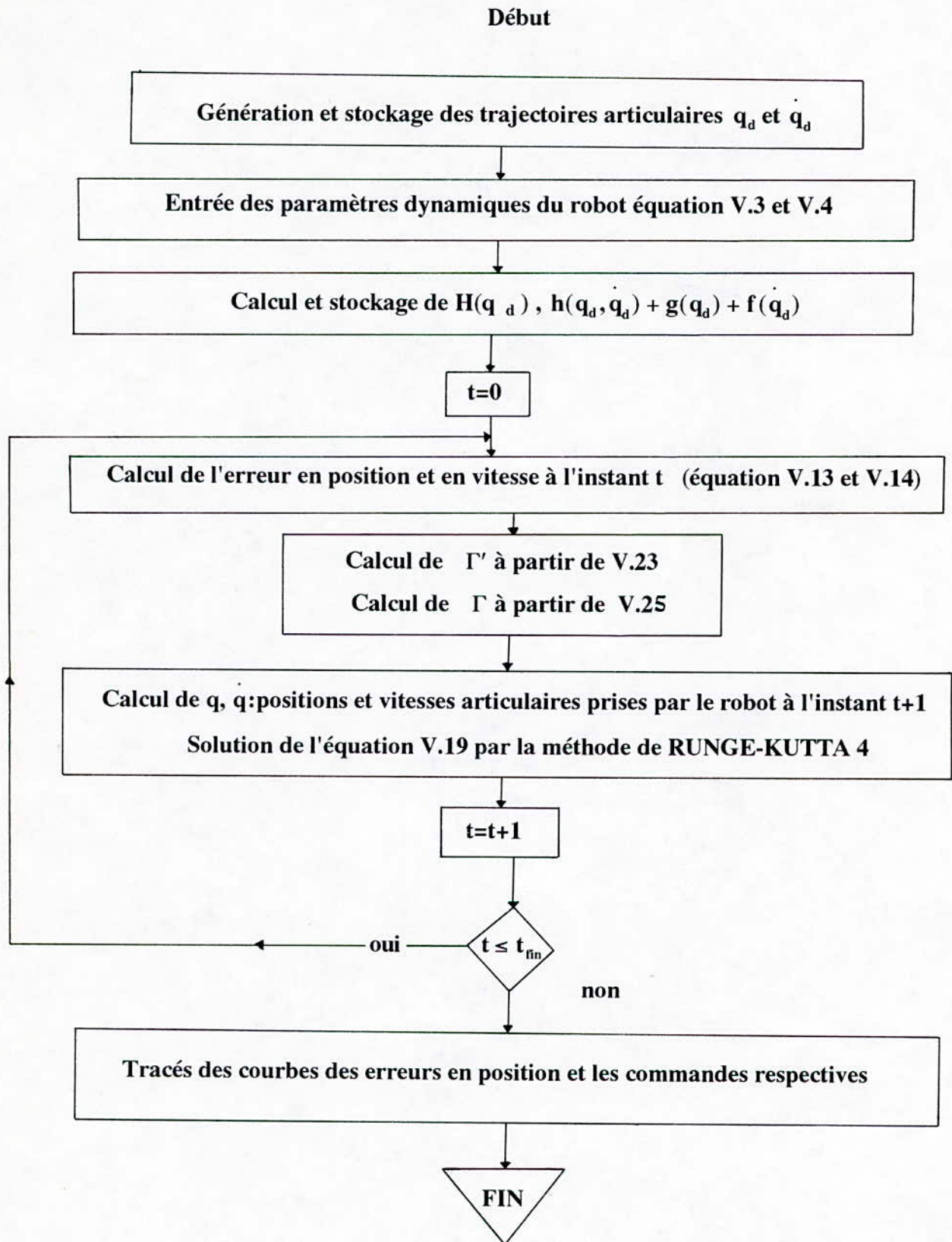


Figure V.16 : Schéma d'une commande linéarisante prédictive

Algorithme de commande



Résultats :

Les résultats d'une commande linéarisante prédictive pour les mêmes données utilisées pour la commande linéarisante simple c'est à dire:

- $\xi=1$ et $\omega_n=10$ c'est à dire $k_{p1}=100$ et $k_{v1}=20$ pour la première articulation.

- $\xi=0.5$ et $\omega_n=10$ c'est à dire $k_{p2}=100$ et $k_{v2}=10$ pour la deuxième articulation.

- $\xi=1.5$ et $\omega_n=10$ c'est à dire $k_{p3}=100$ et $k_{v3}=30$ pour la troisième articulation.

sont montrés par la figure suivante :

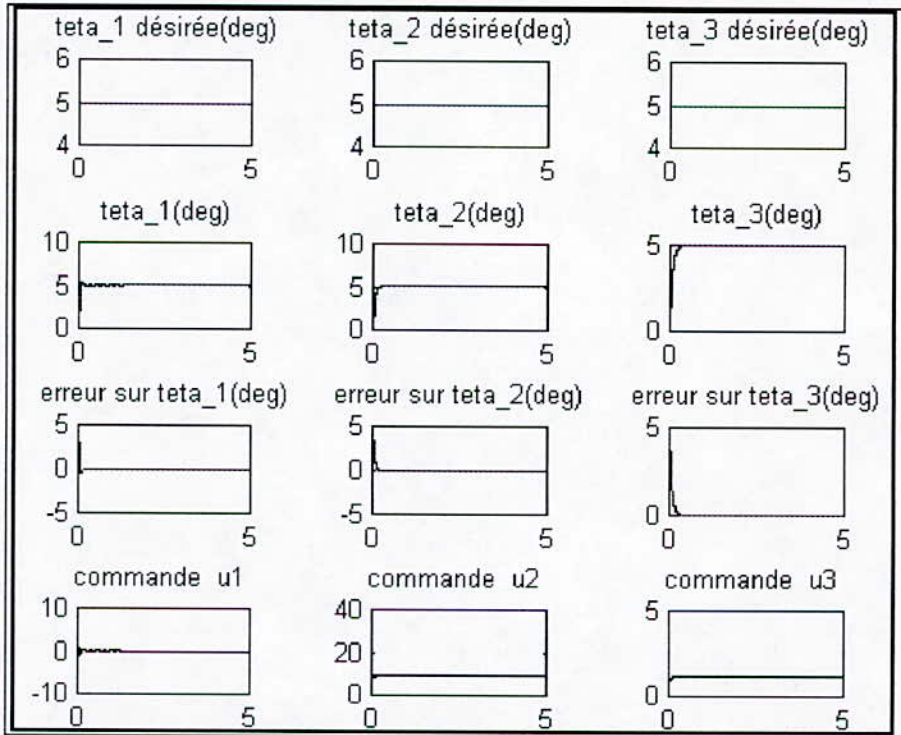


Figure V.17 : Réponse du bras pour une commande linéarisante prédictive .

On remarque qu'il n'y a pas de différences notables avec la commande linéarisante simple.

La figure (IV.18) représente la réponse du bras pour une consigne cycloïdale entre les configurations $[0, 0, 0]$ et $[\pi/4, \pi/4, \pi/4]$ pour $k_{vi}=20$ et $k_{pi}=100$, $i=1,2,3$.

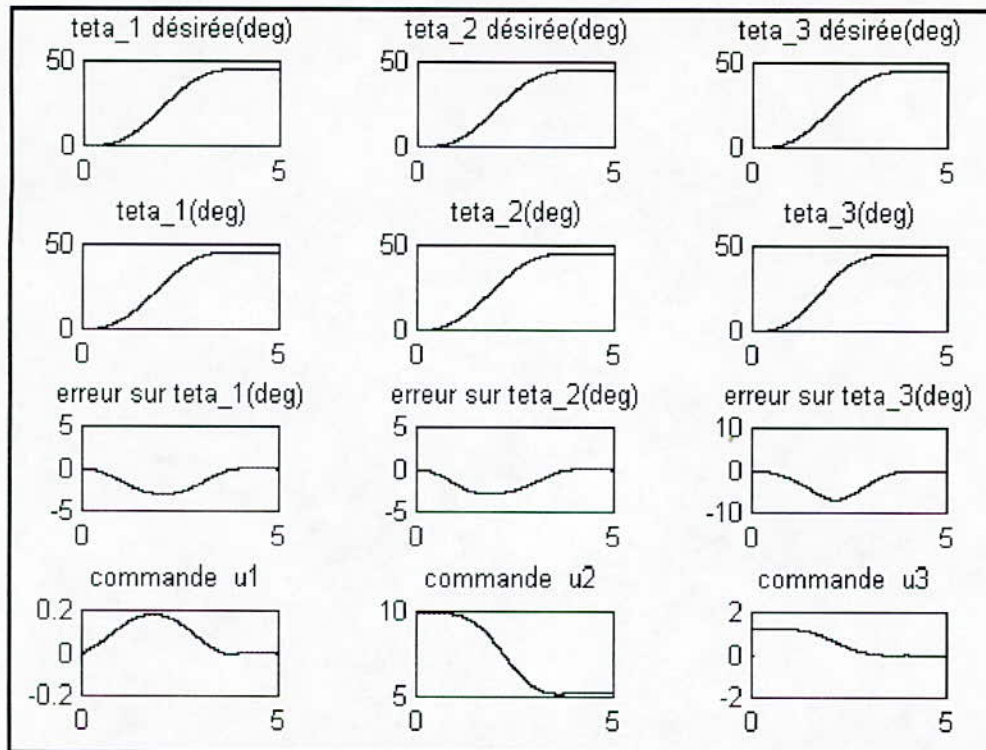


Figure V.18: Réponse du bras pour une commande linéarisante prédictive (consigne cycloïdale).

Là encore, on remarque qu'il n'y a pas de différences notables avec la commande linéarisante simple. C'est pour cela qu'on utilise la commande linéarisante prédictive en temps réel. En effet, ces performances ne s'éloignent pas de celles de la commande linéarisante simple, alors qu'elle présente un gain en temps énorme.

Considérations pratiques [CRA 86] :

Lors de l'élaboration de la commande linéarisante, on a pris en compte quelques suppositions qui sont rarement vraies en pratique.

On va discuter dans ce paragraphe les problèmes pratiques liés à ce type de commande et présenter les solutions actuelles.

La difficulté majeure dans l'utilisation de la commande linéarisante est la connaissance non exacte du modèle du robot d'une part, du fait que certaines composantes du modèle sont extrêmement difficiles à déterminer telles que les termes de frottement visqueux qui sont en général des fonctions non linéaires de la position et de la vitesse articulaire $f(\theta, \dot{\theta})$ [CRA 86].

L'usure de quelques articulations entraîne aussi un changement du modèle du robot [CRA 86].

D'autre part, de sa nature même, le robot manipulateur est contraint à soulever des masses très variées (du simple outil de peinture de masse faible aux charges de manutention);

l'effet de l'outil terminal va sans nulle doute changer le modèle dynamique du robot d'une manière difficile à maîtriser.

Une des méthodes pour l'analyse des effets non modélisés est de les considérer comme un vecteur de perturbation Γ_p agissant sur les articulations dans le modèle du robot; Ce terme Γ_p inclut aussi d'autres effets comme les résonances des segments et des bruits actionneurs[CRA 86].

Le modèle dynamique du robot devient alors:

$$\Gamma_p + \Gamma = H(q)\ddot{q} + h(q, \dot{q}) + g(q) + f(\dot{q}) \quad (\text{V.27})$$

et le vecteur erreur articulaire s'écrit alors à partir de (V.20) et de (V.27) sous la forme suivante:

$$\ddot{e} + k_v \dot{e} + k_p e = H^{-1}(q)\Gamma_p \quad (\text{V.28})$$

La perturbation est modulée par l'inverse de la matrice d'inertie et on remarque que le membre de droite n'est pas découplé car $H(q)$ n'est pas forcément diagonale, donc une perturbation sur une articulation sera répercutée sur les autres.

V.6 Conclusion

Après avoir présenté les problèmes de commande d'un système robotique à travers quelques commandes classiques, nous pouvons conclure que :

- Le choix des gains K_p , K_v , K_i est empirique (Le correcteur ne prend pas en charge les évolutions temporelles du système à commander), et il n'est valable que pour une consigne particulière.
- La forme de l'erreur de suivi de la trajectoire n'est pas prévisible, d'où le risque de perdre le contrôle et la stabilité du robot.

Une commande plus performante, qui est la commande linéarisante, permet de maîtriser la forme de l'erreur de suivi de n'importe quelle trajectoire et ce pour un bras manipulateur quelconque. Néanmoins, elle présente les inconvénients pratiques suivants :

- Nécessité de la connaissance exacte du modèle du robot à commander.
- Elle nécessite une masse importante de calcul en temps réel ("en ligne").
- Cette commande est non robuste vis à vis des erreurs de modélisation.

L'application d'une loi de commande de type **adaptative** s'impose donc pour maîtriser les évolutions temporelles du robot, et permet de le contrôler dans le cas où son modèle est mal connu. En général, ce type de commande utilise un processus adaptatif, qui, basé sur l'observation de la configuration et des erreurs de suivi, ajuste les paramètres non linéaires du modèle jusqu'à l'annulation des erreurs de poursuite. La conception de tels systèmes est un domaine de recherche actuel.

Chapitre VII

PRESENTATION DU LOGICIEL

Chapitre VI

Graphismes et Animations

VI.1 Introduction [DJE 96]

C'est sans doute le brillant essor de l'informatique qui a permis le développement de systèmes de conception et de fabrication assistées par ordinateur.

Ces systèmes facilitent la conception et la fabrication d'objets 3D de forme géométrique complexe en liaison avec la commande des machines automatiques.

L'introduction des robots industriels dans des cellules de travail automatisées nécessite qu'un robot soit capable de coopérer avec d'autres machines et varier son mouvement en fonction de la disposition des autres machines.

Afin de prévoir le comportement d'un robot lorsqu'on lui consigne une tâche donnée, on fait appel aux méthodes de C.A.O (Conception Assistée par Ordinateur). Il s'agit d'une simulation complète sur ordinateur nous permettant de visualiser le comportement du robot sur écran. L'utilisation du graphisme en programmation robotique présente les intérêts suivants:

- 1-Mise au point accélérée du programme de commande d'où un gain de temps appréciable.
- 2-Programmation expérimentale du robot possible avant même que celui-ci ne soit acheté, ce qui permet notamment de simuler sur écran différents robots avant d'investir dans un système robotique.

Néanmoins, ce mode de programmation présente quelques inconvénients:

- 1-Le prix de base d'un système parfait capable de supporter une telle application est très élevé.
- 2-Il n'existe pas de standard commun aux équipements de conception assistée par ordinateur et aux matériels robotique, l'utilisateur est donc obligé d'acquérir des connaissances sur des systèmes très divers, ce qui demeure coûteux.

Dans le présent chapitre, on va présenter quelques algorithmes graphiques simples, que nous avons appliqué à quelques bras manipulateurs. Nous allons présenter en premier lieu une technique de représentation plane de l'espace. Ensuite, on va présenter quelques transformations géométriques,

puis on va présenter un algorithme de traitement des surfaces cachées. On présentera enfin, les routines de bases utilisées dans nos programmes.

VI.2 Représentation plane de l'espace

La représentation d'un espace tridimensionnel dans un espace bidimensionnel (écran), nécessite un outil permettant de projeter l'espace dans un plan pour donner l'illusion d'une forme à trois dimensions. On trouve une variété de méthodes dans la littérature [SCH 87] , nous irons au plus simple et utiliserons une perspective cavalière[LEV 92].

Soit OXY le repère de l'écran et Oxyz le repère tridimensionnel.

La figure (VI-1) montre la représentation plane de (o xyz) dans OXY

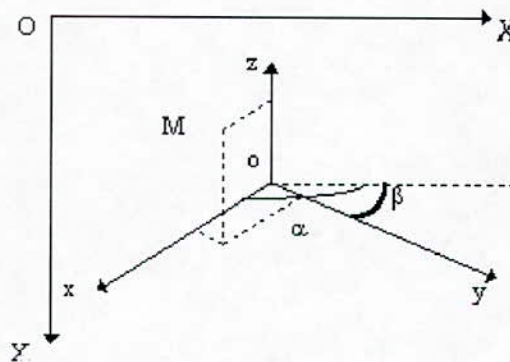


Figure VI.1 : Représentation plane de l'espace.

L'angle entre l'axe OX et l'axe ox sera α , l'angle entre l'axe OX et oy sera β et le point o aura (X_0, Y_0) pour coordonnées dans l'écran:

$$\vec{Oo} = X_0\vec{I} + Y_0\vec{J} \quad (\text{VI.1})$$

Soit M un point de l'espace, ses coordonnées dans repère (o x y z) seront trois réels(x,y,z):

$$\vec{oM} = x\vec{i} + y\vec{j} + z\vec{k} \quad (\text{VI.2})$$

Sa projection dans le plan (l'écran) aura pour coordonnées (X,Y) dans le repère (O X Y) :

$$\vec{OM} = X\vec{I} + Y\vec{J} \quad (\text{VI.3})$$

Les vecteurs unitaires \vec{i} , \vec{j} et \vec{k} seront représentés dans le repère (O X Y) par les formules suivantes:

$$\begin{aligned}
 \bar{i} &= \cos(\alpha)\bar{I} + \sin(\alpha)\bar{J} \\
 \bar{j} &= \cos(\beta)\bar{I} + \sin(\beta)\bar{J} \\
 \bar{k} &= -\bar{J}
 \end{aligned}
 \tag{VI.4}$$

Ce qui donne les formules pour passer des coordonnées de l'espace aux coordonnées de l'écran:

$$\begin{cases}
 X = X_0 + x.\cos(\alpha) + y.\cos(\beta) \\
 Y = Y_0 + x.\sin(\alpha) + y.\sin(\beta) - z
 \end{cases}
 \tag{VI.5}$$

Pour une bonne application de cette méthode, on aura besoin des positions intermédiaires et de la position finale de l'effecteur présentée par la quatrième colonne des matrices de transformation. Donc l'utilisation de ces matrices, présentées dans le chapitre II de ce mémoire, facilitera davantage l'application de cette technique.

VI.3 Transformations géométriques

Pour animer un objet, il est nécessaire de pouvoir le translater, le tourner, l'agrandir etc...; Dans nos programmes, nous avons besoin des positions intermédiaires de l'extrémité de chaque segment. Comme il s'agit de transformations linéaires, le calcul des transformées de ces points suffit pour construire le bras manipulateur. Notons que ces positions intermédiaires sont données par la quatrième colonne des matrices de transformation.

VI.3.1 Translation

Pour un point P représenté par un vecteur \vec{P} , la translation \vec{T} de ce point correspond à la somme des vecteurs : $\vec{P}' = \vec{P} + \vec{T} \Rightarrow$

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \Rightarrow \begin{cases} X' = X + T_x \\ Y' = Y + T_y \\ Z' = Z + T_z \end{cases}
 \tag{VI.6}$$

VI.3.3 Rotation

Pour la rotation, il est important de préciser son **centre** dans l'espace et son **axe**, en plus de l'**angle de rotation**. Examinons le cas simple d'une rotation dans le plan (OXY) autour de l'axe OZ. L'angle de rotation est θ .

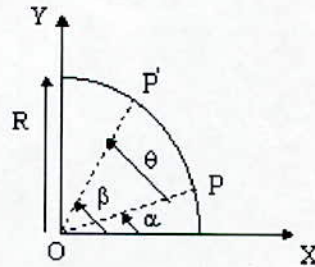


Figure VI.2 : Rotation d'un point dans le plan OXY.

Avant la rotation, les coordonnées du point P sont :

$$\begin{cases} X = R \cos(\alpha) \\ Y = R \sin(\alpha) \end{cases}$$

Après la rotation :

$$\begin{cases} X' = R \cos(\beta) \\ Y' = R \sin(\beta) \end{cases}$$

Par construction $\beta = \alpha + \theta$.

$$\text{Or on a : } \begin{cases} \sin(\beta) = \sin(\alpha + \theta) = \sin(\theta)\cos(\alpha) + \cos(\theta)\sin(\alpha) \\ \cos(\beta) = \cos(\alpha + \theta) = \cos(\alpha)\cos(\theta) - \sin(\alpha)\sin(\theta) \end{cases}$$

$$\text{Ainsi : } \begin{cases} X' = X\cos(\theta) - Y\sin(\theta) \\ Y' = X\sin(\theta) + Y\cos(\theta) \\ Z' = Z \end{cases}$$

Où bien sous forme plus élégante:

$$\vec{P}' = R_z \vec{P} \text{ Avec } R_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{VI.7})$$

De la même manière, on tire la matrice de rotation autour de l'axe OY:

$$R_y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (\text{VI.8})$$

Et celle autour de l'axe OX :

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (\text{VI.9})$$

VI.4 Elimination des parties cachées

L'élimination des parties cachées permet de conférer aux objets leur opacité naturelle, leur donnant l'aspect de volumes pleins au lieu de squelettes filiformes. En effet, un simple cube dessiné avec toutes ses arêtes est ambigu; seule l'élimination des parties cachées permet de déterminer sous quel angle il est vu comme l'illustre la figure suivante:

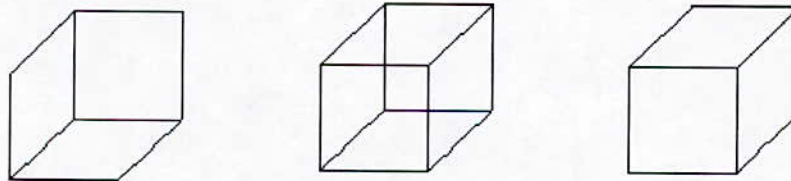


Figure VI.3 : Cube sans élimination des lignes cachées et interprétations possibles.

Cette image peut être interprétée comme un conduit de section carrée ouvert aux deux extrémités. On trouve divers algorithmes de parties cachées dans la littérature [SCH 87]; Nous avons opté pour la méthode du **tri par éloignement** [Depth Sort]. Elle porte le nom de **l'algorithme du peintre**.

Si une priorité doit être associée à chacune des facettes, l'image est simplement obtenue en les affichant selon l'ordre de leur priorité. La facette de plus faible priorité, c'est à dire la plus éloignée de l'observateur, est tracée en premier. Les facettes les plus proches, qui sont peintes en dernier, peuvent ainsi recouvrir les facettes éloignées déjà tracées. Cette méthode est inspirée de la manière habituelle de réaliser une peinture, en remplissant d'abord toute la toile par les couleurs constituant l'arrière plan; puis en mettant en place les divers objets par plans successifs en **superposant des couches de peinture**. Nous avons appliqué cette méthode à la base du bras RRR qui est représentée par un cube. Celui-ci est représenté par un tableau de structure de taille huit (*Poly*). Chaque structure est composée des coordonnées spatiales des huit sommets du cube. On effectue le calcul de la profondeur moyenne (x_{moyen}) de chaque face du cube (6 au total) et effectuons leur tri par ordre décroissant de profondeur moyenne dans le tableau *triface*. On effectuera l'affichage des faces selon cet ordre (la face la plus profonde est la première à être dessinée). Notons que la face la plus profonde correspond à un x_{moyen} le plus faible du fait de la direction de l'axe ox' (voir figure VI.1 pour $\alpha=90^\circ$, $\beta=0^\circ$).

Nous présentons ci-dessous la sous-routine *SurfaceCache* effectuant le traitement des surfaces cachées.

```
void TMaFenetre::SurfaceCache()
```

```
{
```

```
int ij;
```

```
float temp, triface[6];
```

```
DefPoly change;
```

```
for(i=0;i<(6);i++)
```

```
{
```

```
triface[i]=0.0;
```

```
for(j=0;j<4;j++)
```

```
triface[i]=triface[i]+som[Poly[i].dsom[j]].x;
```

```
asm {  
    CLI  
    MOV DX, 43h  
    MOV AL, 36h  
    OUT DX, AL  
    SUB DX, 3  
    XOR AX, AX  
    OUT DX, AL  
    OUT DX, AL  
    STI  
}
```

VI.6 Conclusion

Nous avons présenté dans ce chapitre quelques algorithmes graphiques de base. Ceux-ci ne constituent qu'une initiation au problème de la représentation et de l'animation d'objets 3D. En effet, il reste à donner une forme plus proche de la réalité au robot; Le rendu réaliste n'est possible qu'en utilisant des méthodes de traitement de surfaces cachées plus élaborées, ainsi qu'un algorithme d'éclairage.

Néanmoins, il faut toujours garder un compromis entre la qualité de l'image et la rapidité de l'animation. En effet, une représentation plus proche de la réalité du robot nécessite l'emploi d'algorithmes plus élaborés qui eux nécessitent un temps de calcul plus long.

On conclut qu'une animation plus élaborée nécessite l'utilisation de techniques graphiques plus performantes, ainsi qu'un matériel puissant (carte graphique puissante).

Chapitre VI

GRAPHISME ET ANIMATION

Chapitre VII

Présentation du logiciel

VII.1 Introduction

Le logiciel que nous proposons a été conçu dans le but de servir à tout intéressé au domaine de la ROBOTIQUE. Il est particulièrement destiné à être un support du cours de ROBOTIQUE enseigné en 5^{ème} année ELECTRONIQUE à l'E.N.P.

Il a été réalisé en partie avec le langage C++(version 3.1 du Borlandc++)[DEL 94], et en autre par MATLAB 4.2 [MAT 93]. Nous l'avons nommé ROBSICOM (ROBots : SIMulation & COMmande).

Il est composé d'un menu principal dans lequel on propose à l'utilisateur différents articles correspondant chacun à une option.

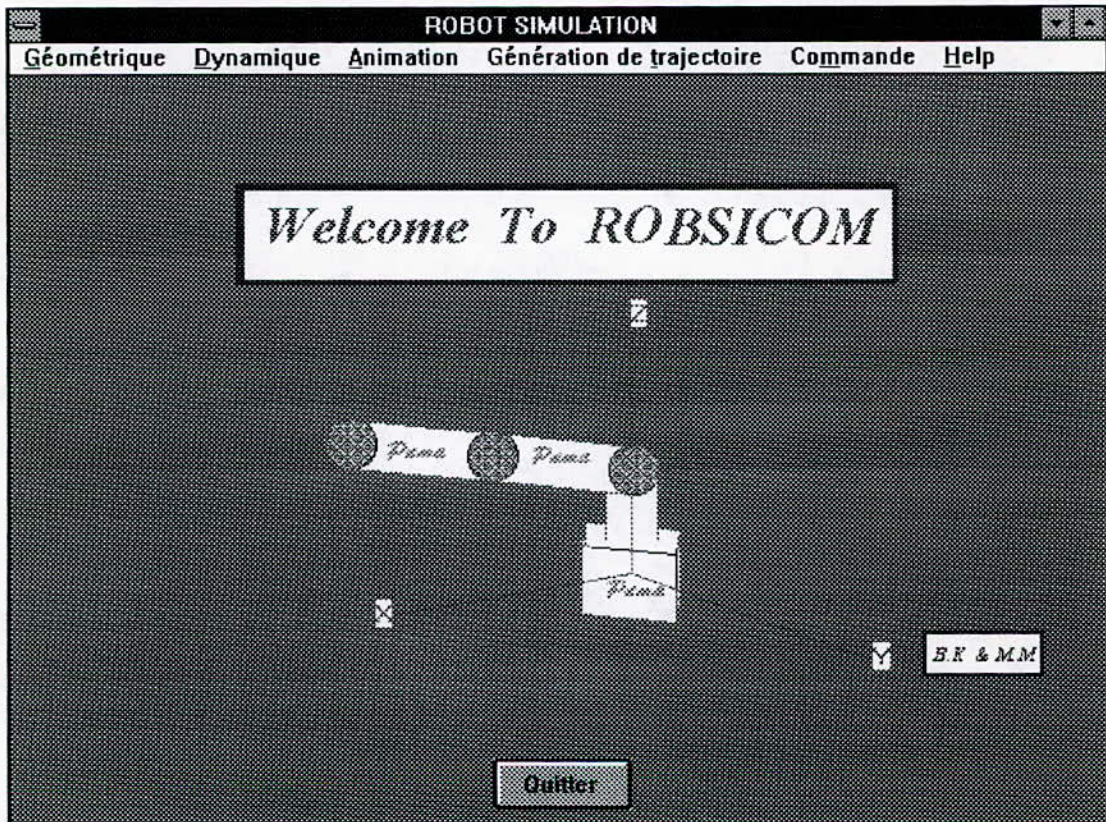


Figure VII.1: Menu principal de ROBSICOM.

Il s'agit des cinq options suivantes:

VII. 2 Modélisation géométrique

Dans laquelle on trouve 2 options:

- géométrie directe.
- géométrie inverse.

Pour chaque option on propose la simulation géométrique (directe ou inverse) des 5 bras manipulateurs suivants:

- Bras articulé à trois degrés de liberté rotoïdes (RRR).
- Bras cylindrique (TRT).
- Bras sphérique (RRT).
- Bras articulé à deux degrés de liberté (PUMA).
- Bras cartésien (TTT).

En sélectionnant un de ces articles, une boîte de message apparaîtra à l'écran indiquant à l'utilisateur la commande qu'il doit taper à l'invite du MATLAB.

En tapant cette commande, l'utilisateur aura un menu dans lequel il devra remplir une boîte de saisie contenant des informations sur : le bras choisi (dimensions), la position désirée (modélisation géométrique directe) ou la configuration du bras (modélisation géométrique inverse). La figure (VII.2) indique le menu présenté sous MATLAB lors de la modélisation géométrique inverse d'un bras RRR.

En cliquant sur le bouton OK, il aura l'information requise à savoir la configuration du bras (géométrie inverse) ou la position de l'extrémité terminale du bras (géométrie directe).

La figure (VII.3) indique la réponse du programme pour des dimensions données et une position désirée particulière d'un bras RRR. On y trouve toutes les configurations possibles.

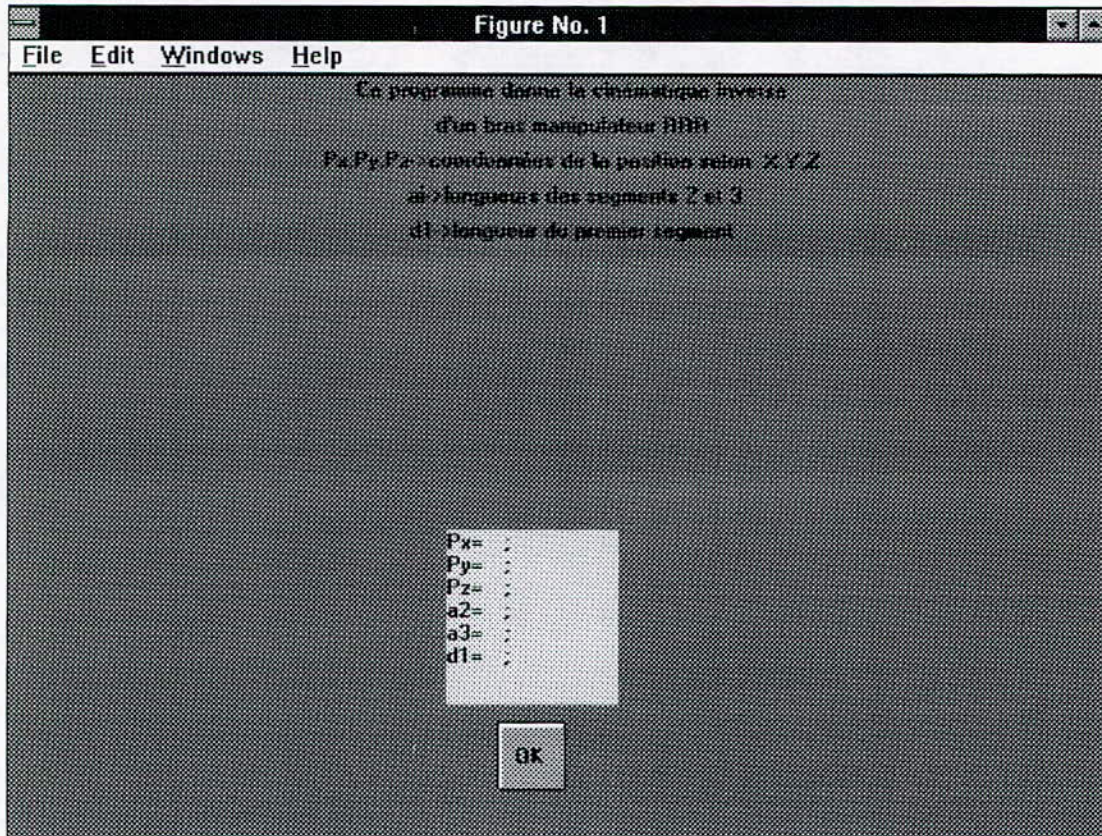
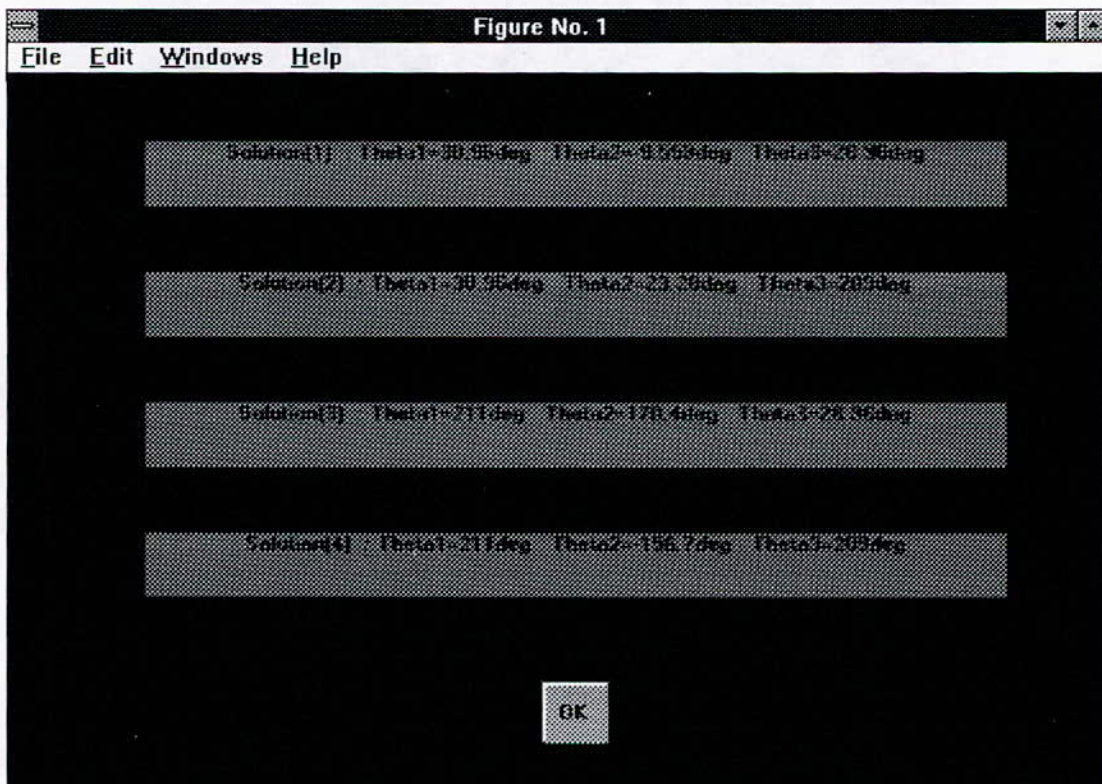


Figure VII.2 : menu proposé par MATLAB en tapant 'modirrr'.



FigureVII.3: Réponse du programme 'modirrr'.

VII.3 Modélisation dynamique:

Dans cette partie, on propose à l'utilisateur un outil lui permettant de simuler le comportement en boucle ouverte de 3 bras manipulateurs:

- Bras articulé à trois degrés de liberté (RRR).
- Bras cylindrique (TRT).
- Bras articulé à deux degrés de liberté (PUMA).

En sélectionnant un article correspondant à un de ces trois bras, une boîte de message surgit indiquant à l'utilisateur la commande qu'il doit taper à l'invite du MATLAB. En tapant cette commande, l'utilisateur aura un menu dans lequel il pourra agir sur certains paramètres physiques du bras sélectionné (masse, dimensions). La figure (VII.4) indique le menu présenté sous MATLAB lors de la modélisation dynamique d'un bras RRR. En cliquant sur OK, on aura l'information requise à savoir la réponse indicielle de chaque articulation en position et en vitesse. La figure (VII.5) montre le résultat d'une simulation d'un bras RRR.

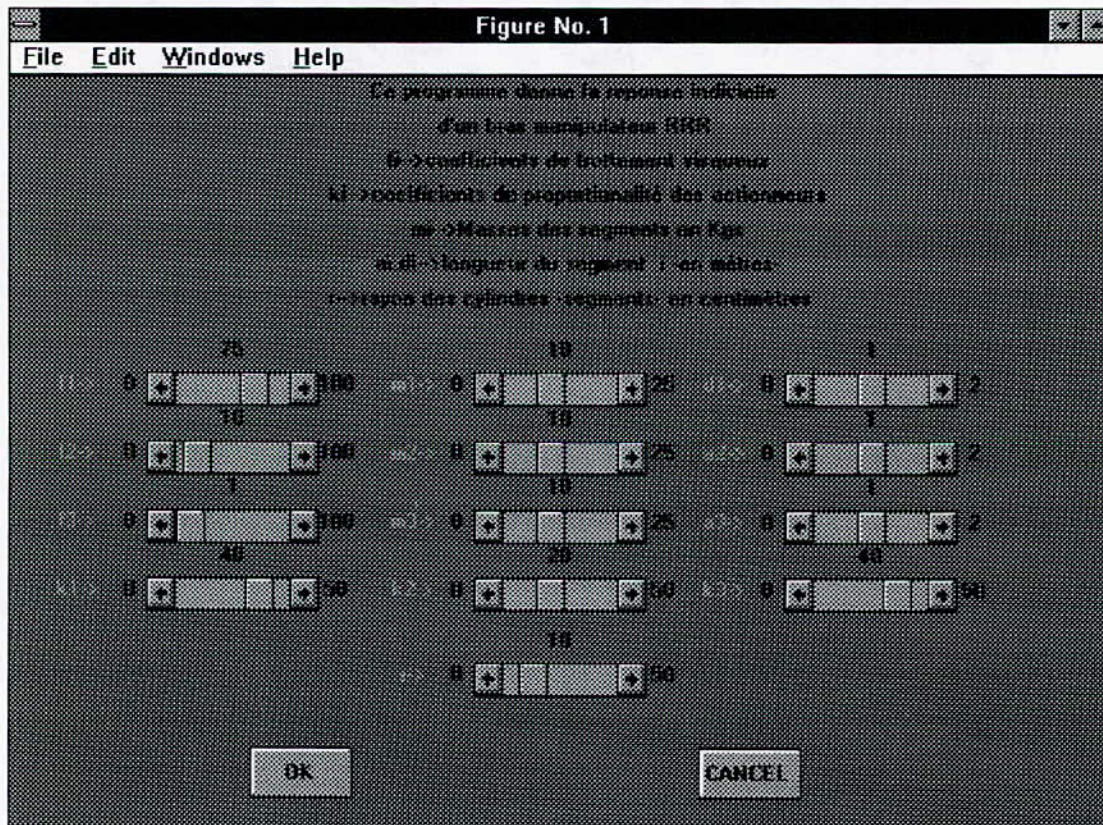


Figure VII.4 : menu proposé par MATLAB en tapant 'modd_rrr'.

VII.4 Visualisation graphique

On propose dans cette partie la visualisation graphique en mouvement de trois bras manipulateurs: bras RRR, bras PUMA, bras cartésien (TTT).

En sélectionnant un des trois articles correspondants, un sous-menu apparaîtra. La figure (VII.6) montre le menu relatif au bras RRR. Il contient les articles suivants:

-Dimensions: En le sélectionnant, on pourra donner les dimensions du bras à l'aide d'une boîte de dialogue.

-Position désirée: En le sélectionnant, on donnera les coordonnées cartésiennes du point cible. La position initiale correspond à celle où toutes les variables articulaires sont nulles.

-Géométrie inverse: permettant à l'utilisateur d'avoir la configuration du bras correspondante à la position finale.

-Visualisation: En le sélectionnant, on pourra visualiser le mouvement du bras sélectionné.

Chaque variable articulaire varie linéairement dans le temps.

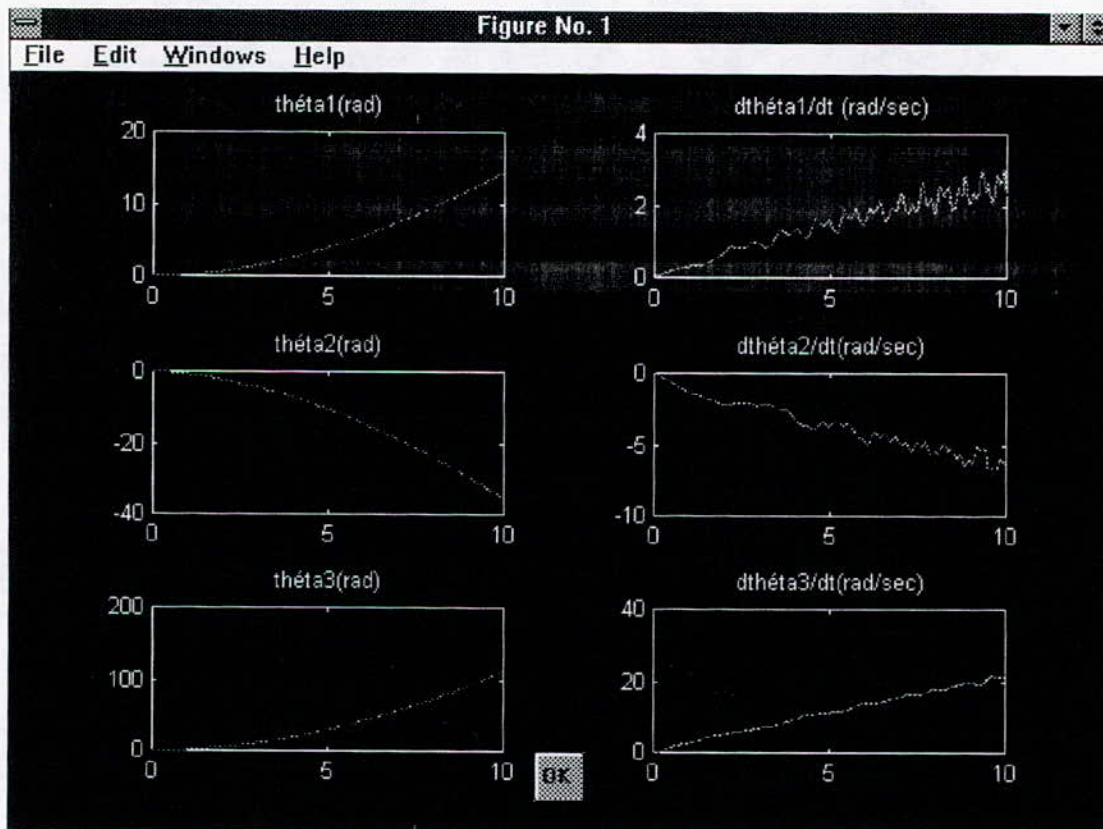


Figure VII.5. Réponse indicielle d'un bras RRR

VII.5 Génération de trajectoires

On propose dans cette partie un outil de génération de trajectoire entre deux points de l'espace pour un bras RRR.

Quatre articles sont proposés dont trois correspondent à la génération de trajectoires dans l'espace des variables électriques ('Polynomiale', 'cycloïdale' et 'linéaire') et un générant une trajectoire linéaire dans l'espace cartésien ('Poursuite Linéaire').

En sélectionnant un des quatre articles, un sous-menu apparaîtra. La figure (VII.7) montre le menu relatif à l'article 'Cycloïdale'. Celui-ci contient les articles suivants:

-Données: En le sélectionnant, on pourra spécifier les coordonnées cartésiennes du point de départ et de celui d'arrivée.

-Visualisation: contenant deux sous articles:

-Graphiques: permettant la visualisation du mouvement du manipulateur.

-Accélération-Vitesses: En sélectionnant cet article, une boîte de message apparaîtra indiquant la commande qu'on doit taper à l'invite du MATLAB.

En tapant cette commande à l'invite du MATLAB, on pourra visualiser les courbes de la position, de la vitesse et de l'accélération en fonction du temps et ce pour chaque articulation.

La figure (VII.8) montre un tel résultat présenté sous MATLAB pour une trajectoire cycloïdale.

Remarque: Dans le cas de la génération de trajectoires linéaires avec des limites paraboliques, un article supplémentaire est proposé. Il est composé de deux sous articles : un correspondant aux vitesses des articulations durant la partie linéaire du mouvement et l'autre correspondant aux accélérations durant les parties paraboliques.

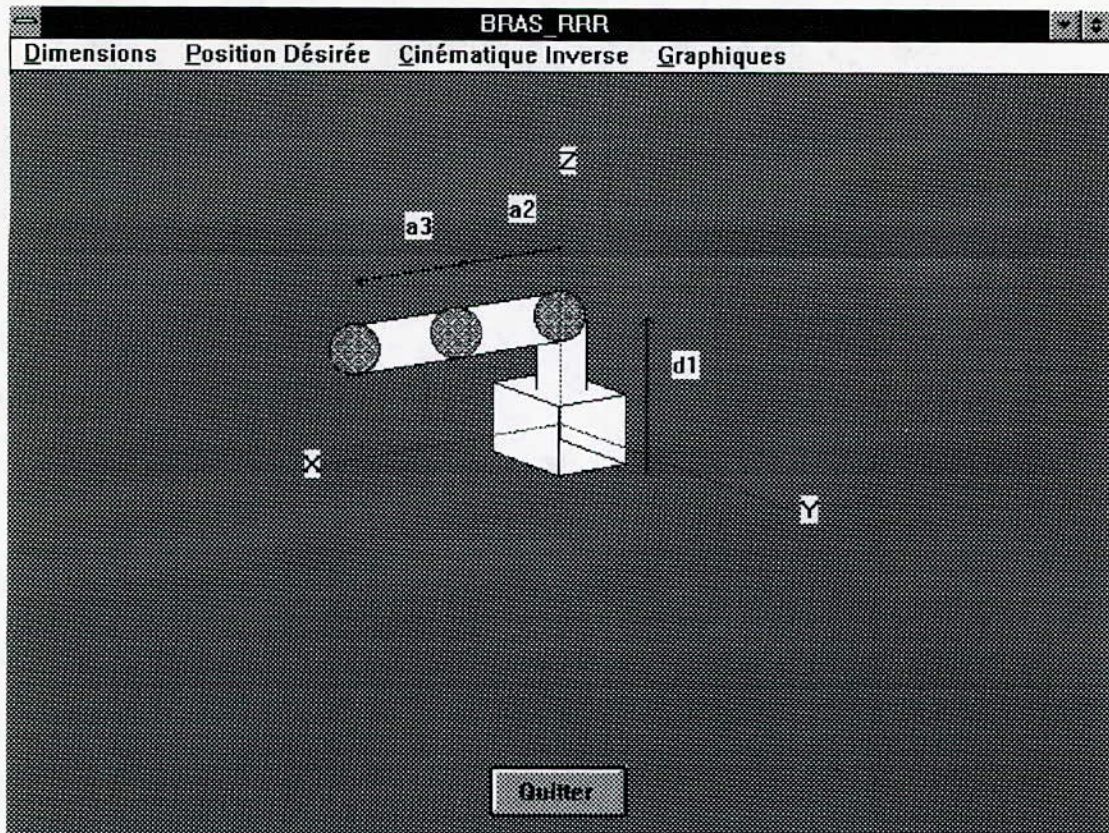


Figure VII.6 : Menu correspondant à la visualisation graphique d'un bras RRR.

VII.6 Commande

Dans cette partie, on propose un outil de simulation pour la commande d'un bras RRR ayant les caractéristiques données au chapitre (V).

On y trouve les trois articles:

-Classique : permet une commande basée sur la linéarisation du système en négligeant tout terme non linéaire. En sélectionnant cet article, une boîte de message apparaît contenant la commande que l'utilisateur devra taper à l'invite du MATLAB. En tapant cette commande, Le menu présenté par la figure (VII.9) apparaît sous MATLAB. On peut agir sur la constante de temps du système d'ordre 2 classique que représente chaque articulation (Tau_i $i=1,2,3$). La valeur du coefficient d'amortissement est prise égale à un.

En cliquant sur OK, on pourra visualiser les courbes d'erreurs pour chaque articulation. La figure (VII.10) indique un tel résultat. En cliquant de nouveau sur OK, on pourra avoir les valeurs des constantes Kp , Kv et Ki pour chaque articulation.

-PID classique :

-PID classique :

En sélectionnant cet article, une boîte de message surgit comme pour l'article précédent. En tapant la commande qu'elle contient à l'invite du MATLAB, le menu présenté par la figure (VII.11) apparaît.

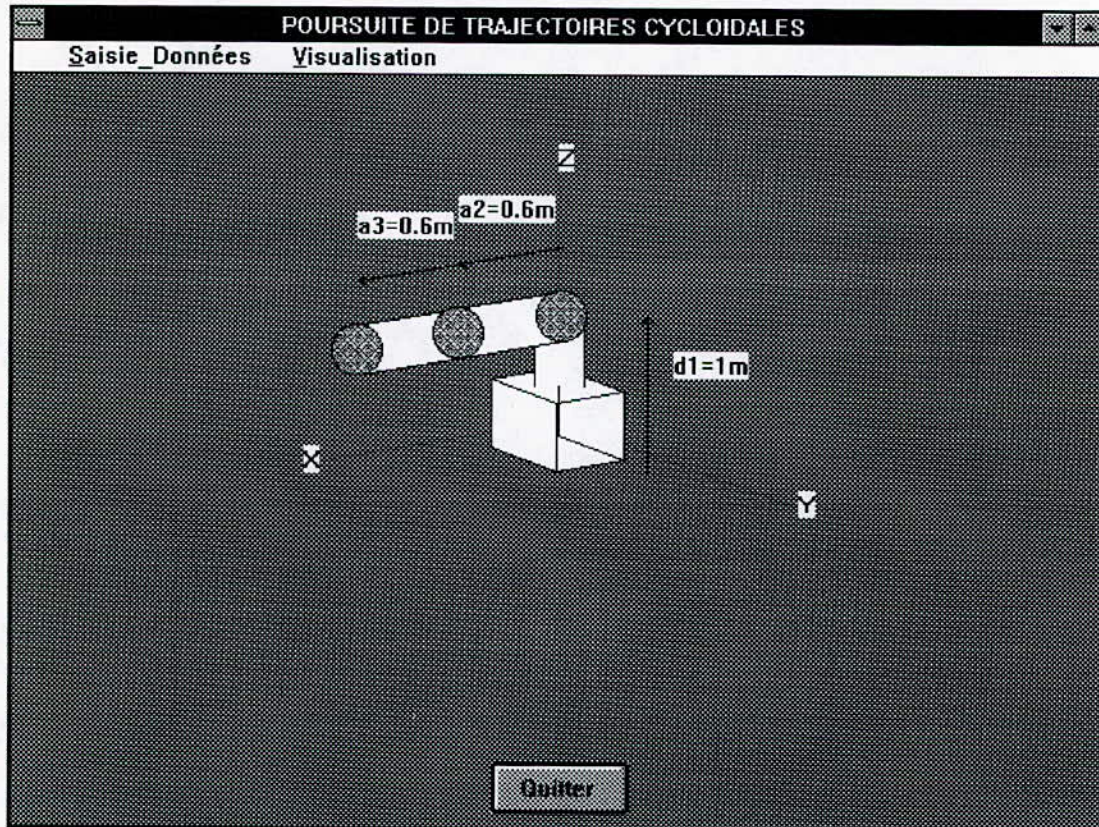


Figure VII.7 : Sous- menu relatif à la génération de trajectoires cycloïdales.

On peut agir sur les constantes K_p , K_v et K_i pour chaque articulation. En cliquant sur OK, on aura les courbes d'erreurs pour chaque articulation. La figure (VII.12) montre un tel résultat.

-PID linéarisante: permet la simulation de la commande linéarisante du bras RRR. En tapant la commande proposée par une boîte de message à l'invite du MATLAB, on aura le menu présenté par la figure (VII.13). On peut agir sur les constantes K_p , K_v et K_i pour chaque articulation.

En cliquant sur OK, on pourra visualiser les courbes d'erreurs pour chaque articulation. La figure (VII.14) indique un tel résultat.

- **Remarque:** Pour les programmes concernant la commande, l'utilisateur pourra changer les paramètres du bras dans le fichier "**Données.m**". Ce fichier contient tous les paramètres du bras (masses, dimensions, constantes, coefficients de proportionnalité du moteur etc...).
En outre, on a prévu l'ajout d'autres types de commande au logiciel. Ceci est garantie par une programmation modulaire. Ainsi, la trajectoire figure dans le fichier "**Trajet.m**" quant à la commande, on la trouve dans un fichier à part. Il suffit donc de dérouter l'appel de ce fichier pour ajouter un autre type de commande.

VII.7 Help

Cette partie donne à l'utilisateur une aide quant à l'utilisation du logiciel. Elle contient les 6 articles suivants:

- Géométrique:** Aide sur la partie '**Modélisation géométrique**' du logiciel.
- Dynamique :** Aide sur la partie '**Modélisation dynamique**' du logiciel.
- Graphique:** Aide sur la partie '**Graphique**' du logiciel.
- Trajectoire:** Aide sur la partie '**Génération de trajectoires**' du logiciel.
- Commande:** Aide sur la partie '**Commande**' du logiciel.
- A propos:** Aide sur le logiciel en sa globalité.

La figure (VII.15) montre l'aide proposée suite au choix de l'article '**A propos**'.

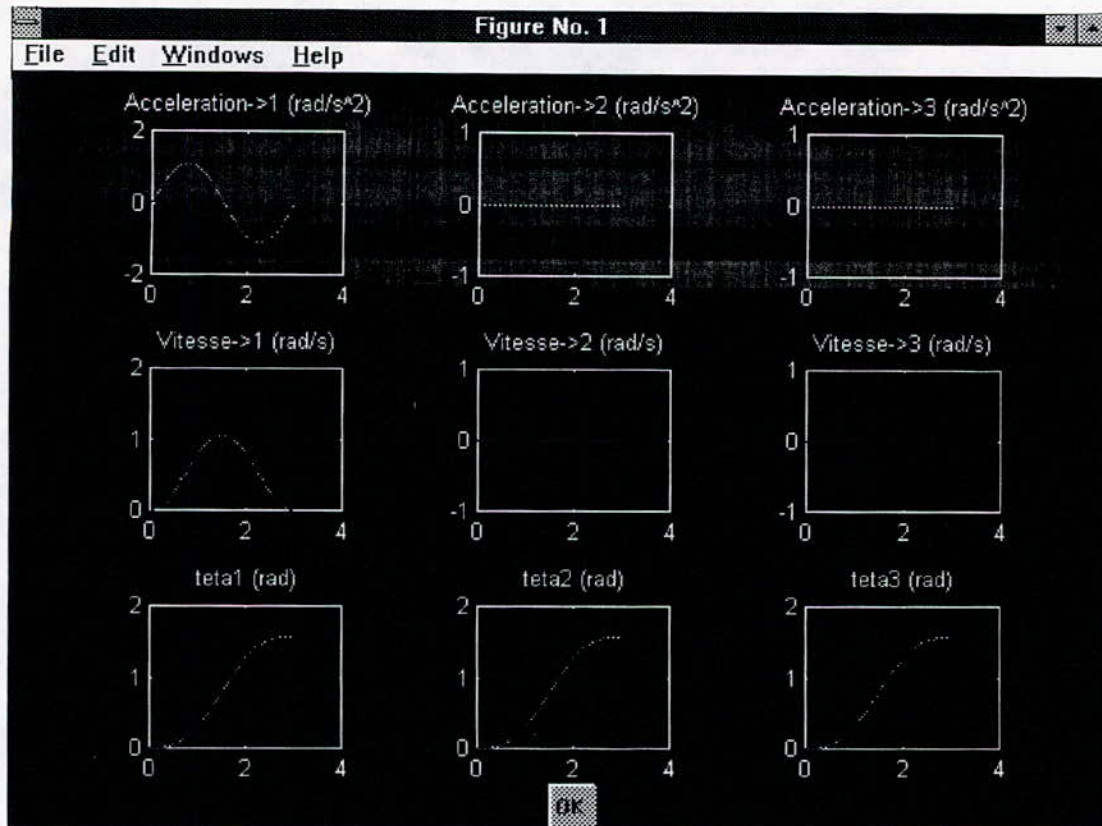


Figure VII.8. Visualisation des positions, vitesses et accélérations des articulations

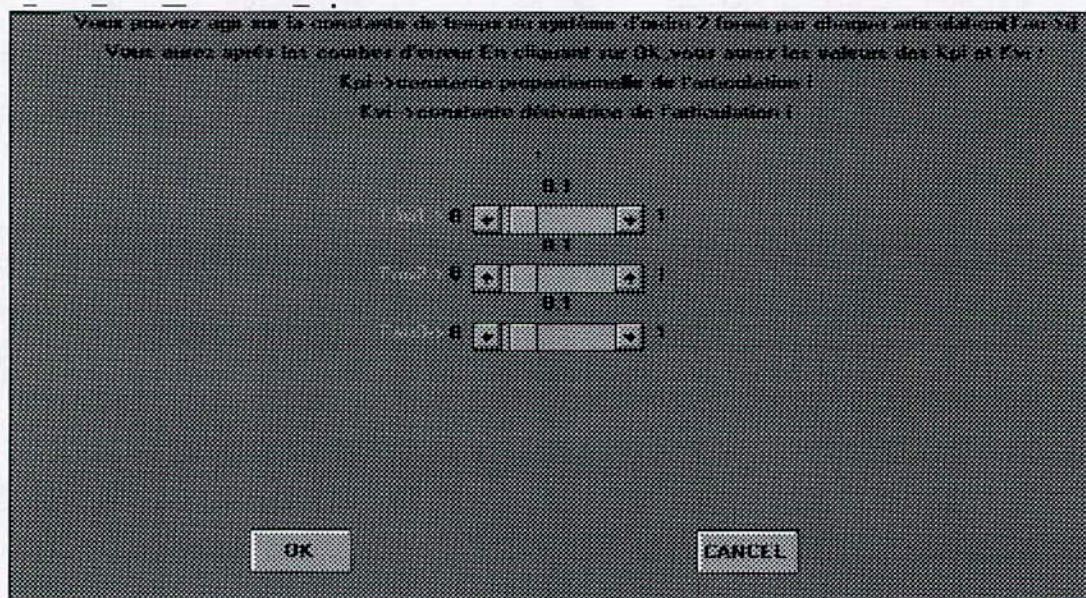


Figure VII.9. Menu relatif à la simulation classique d'un bras RRR (Linéarisation).

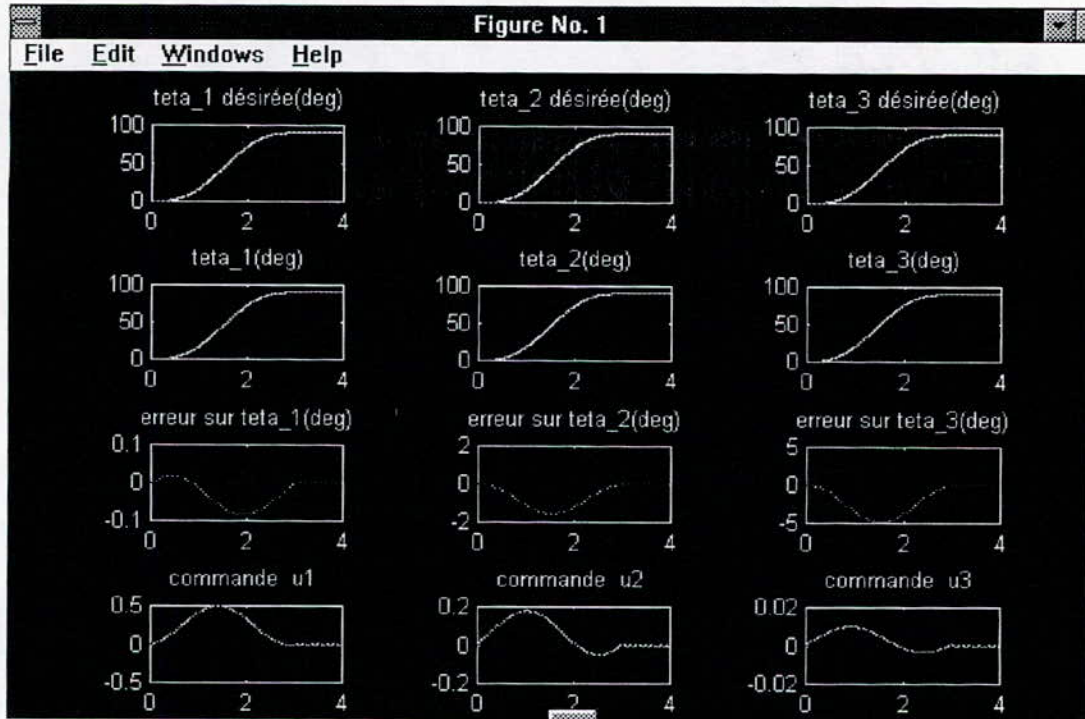


Figure VII.10 : Courbes d'erreur pour une commande classique avec $T_{aui}=0.1\text{sec}$ $i=1,2,3$.

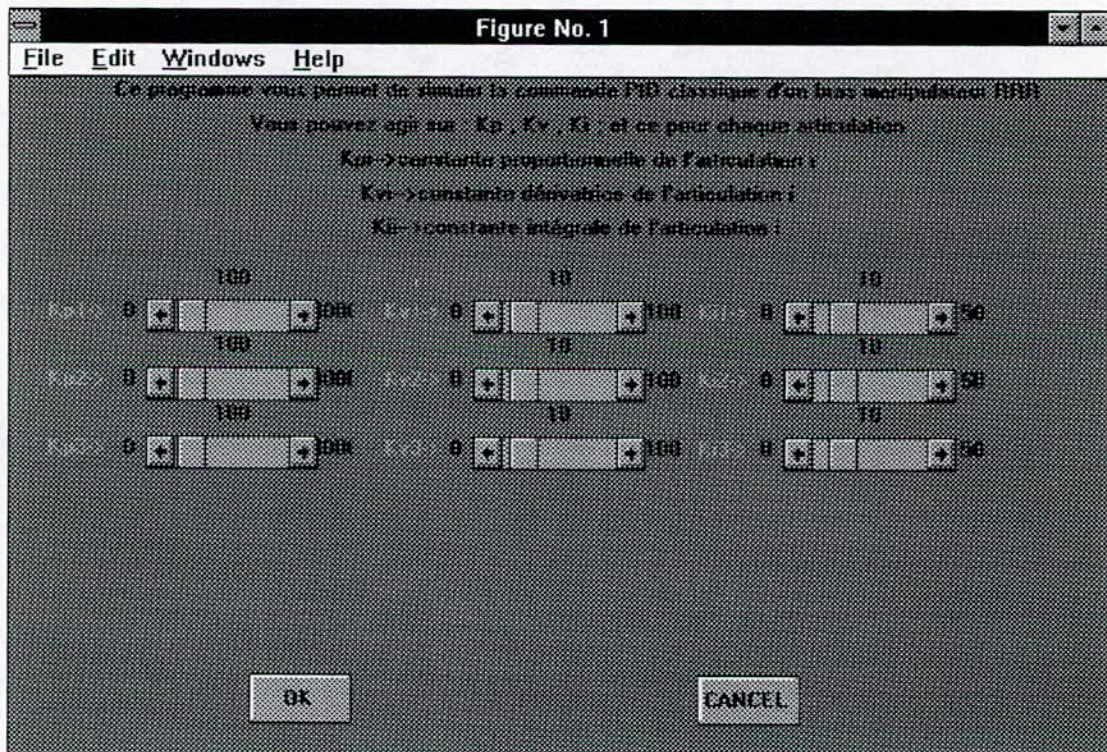


Figure VII.11. Menu relatif à la simulation PID classique d'un bras RRR.

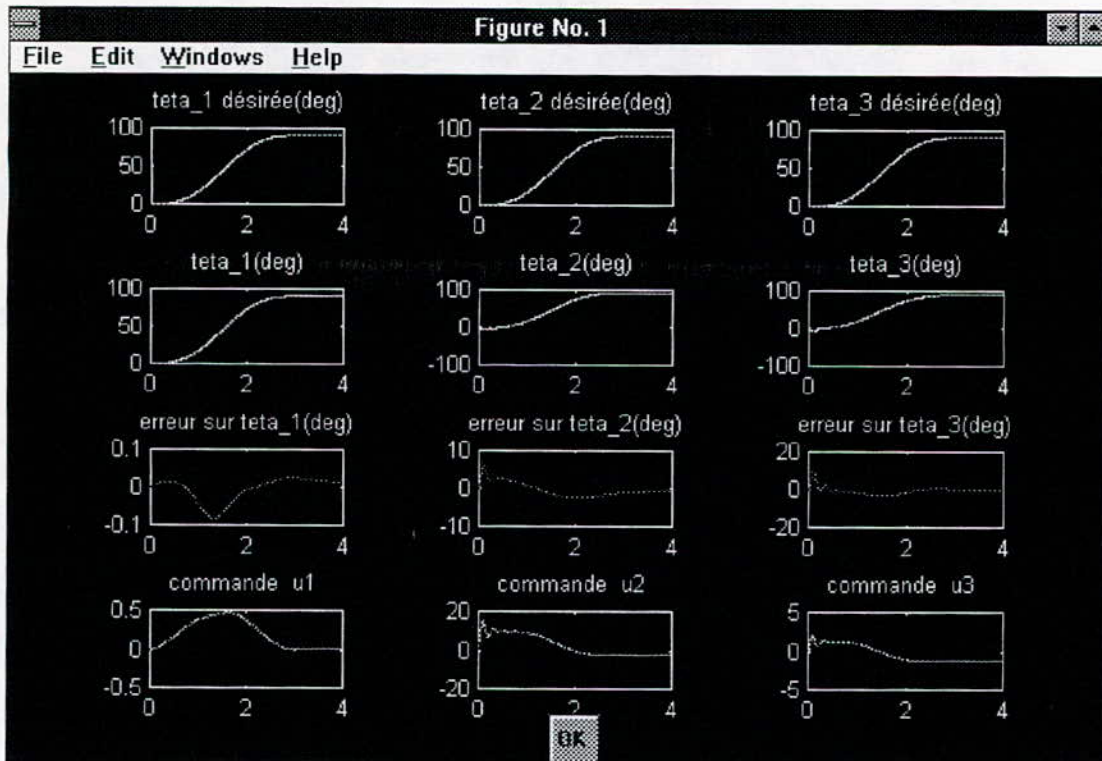


Figure VII.12 : Résultat de simulation de la commande PID classique d'un bras RRR.

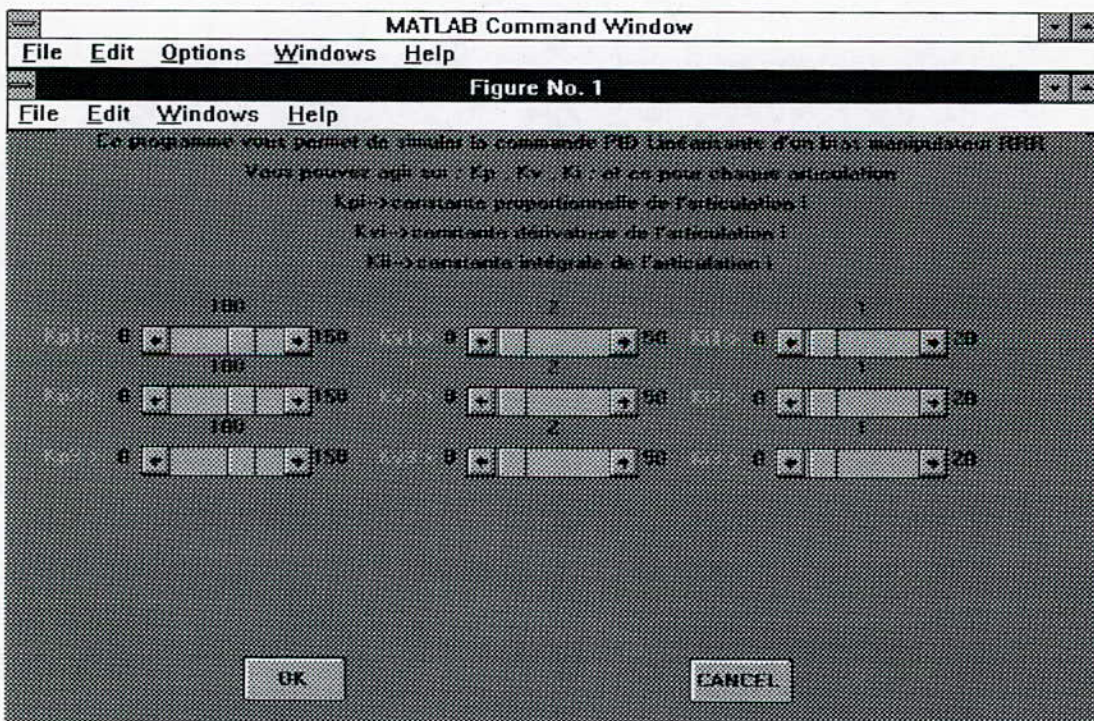


Figure VII.13 : Menu relatif à la simulation de la commande linéarisante d'un bras RRR

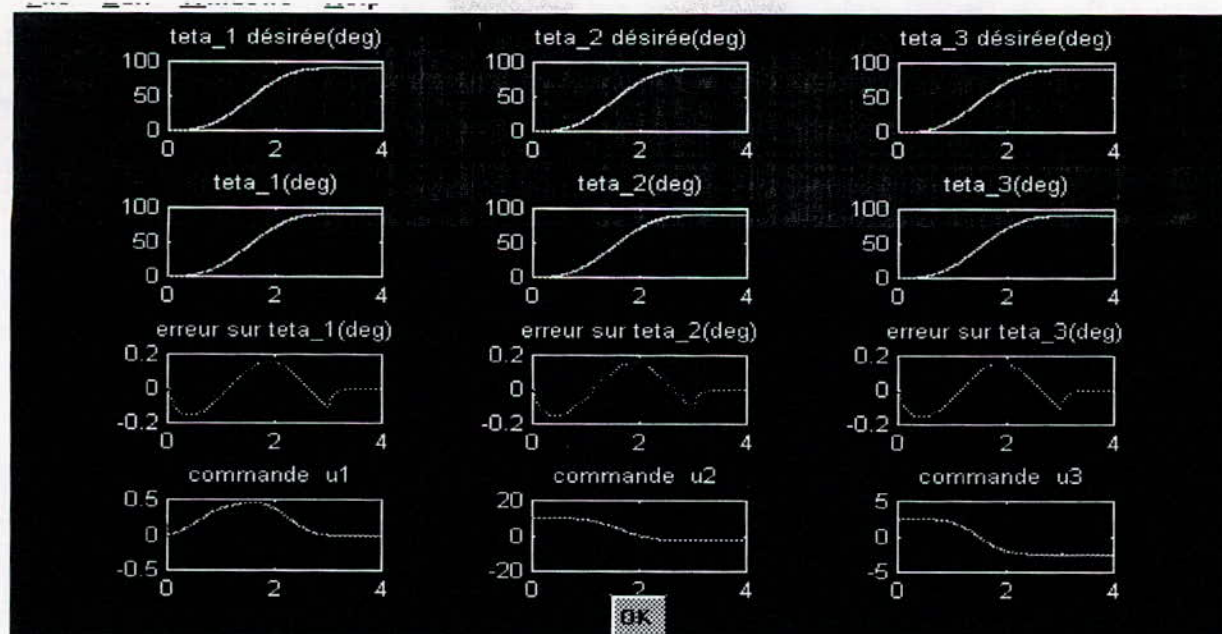


Figure VII.14 : Résultats de simulation de la commande linéarisante d'un bras RRR.



Figure VII.15 : Boîte d'aide surgissant lors de la sélection de l'article "A propos...".

CONCLUSION GENERALE

Conclusion générale

Nous avons présenté à travers ce travail, un logiciel sous WINDOWS, permettant la simulation graphique et dynamique de bras manipulateurs. Cette simulation s'avère nécessaire avant toute conception et réalisation de robots.

Notre étude a concerné principalement un bras articulé à trois degrés de liberté, ainsi que d'autres robots classiques (T.R.T, PUMA à 2 d.d.l, T.T.T); L'extension pour n'importe quelle autre structure de robot est possible.

Le logiciel englobe dans sa principale partie la poursuite de trajectoires dans l'espace opérationnel et dans l'espace articulaire (sous un aspect cinématique), ainsi que quelques lois de commande appliquées au robot articulé.

En perspective, ce travail reste ouvert sur les aspects suivants :

- L'élaboration de lois de commande performantes, en position et en force, pour le contrôle du bras.
- Poursuite de trajectoires quelconques, en tenant compte du modèle dynamique du robot, ainsi que de son environnement.
- Amélioration de la visualisation et de l'animation 3D du robot.

Annexe

A.1 Modélisation du bras articulé par la méthode d'Euler-Lagrange :

On rappelle les matrices de transformations déjà calculées dans le deuxième chapitre :

$$A_1 = {}^0A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; A_2 = {}^1A_2 = \begin{bmatrix} C_2 & S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = {}^2A_3 = \begin{bmatrix} C_3 & S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^1T_3 = A_2A_3 = \begin{bmatrix} C_{23} & -S_{23} & 0 & a_3C_{23} + a_2C_2 \\ S_{23} & C_{23} & 0 & a_3S_{23} + a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_2 = A_1A_2 = \begin{bmatrix} C_1C_2 & -C_1S_2 & S_1 & a_2C_1C_2 \\ S_1C_2 & -S_1S_2 & -C_1 & a_2S_1C_2 \\ S_2 & C_2 & 0 & a_2S_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_3 = A_1A_2A_3 = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & S_1 & C_1(a_3C_{23} + a_2C_2) \\ S_1C_{23} & -S_1S_{23} & -C_1 & S_1(a_3C_{23} + a_2C_2) \\ S_{23} & C_{23} & 0 & a_3S_{23} + a_2S_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Type des segments utilisés

On a choisit des segments cylindriques (figure A.1). Chaque segment (S_i) est lié à son repère de transformation en son sommet : $R_i(x_i, y_i, z_i)$.

Ces différents moments d'inertie sont:

Sa masse m_i .

$\bar{x}_i = \bar{z}_i = 0, \bar{y}_i = -\frac{L}{2}$ (coordonnées du centre de masse dans R_i).

$$I_{ixx} = \int_{(S_i)} (y^2 + z^2) dm = m \left(\frac{L^2}{3} + \frac{R^2}{4} \right).$$

$$I_{izz} = I_{ixx}.$$

$$I_{yy} = \int_{(S_i)} (x^2 + z^2) dm = \frac{mR^2}{2}.$$

$$I_{ixz} = I_{iyz} = I_{ixy} = 0.$$

La matrice d'inertie de chaque segment (S_i) sera notée:

$$J_i = \begin{bmatrix} \alpha_i & 0 & 0 & \bar{x}_i m_i \\ 0 & \beta_i & 0 & \bar{y}_i m_i \\ 0 & 0 & \gamma_i & \bar{z}_i m_i \\ \bar{x}_i m_i & \bar{y}_i m_i & \bar{z}_i m_i & m_i \end{bmatrix} \quad \text{Avec :} \quad \begin{cases} \alpha_i = \frac{-I_{ixx} + I_{iyy} + I_{izz}}{2} \\ \beta_i = \frac{I_{ixx} - I_{iyy} + I_{izz}}{2} \\ \gamma_i = \frac{I_{ixx} + I_{iyy} - I_{izz}}{2} \end{cases} \quad i=1,2,3$$

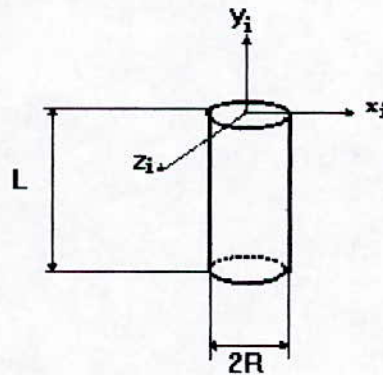


Figure A-1 Segment "i" du bras

L'établissement du modèle revient au calcul des différents termes D_{ij} , D_i , D_{ijk} ; et l'injection de ces termes dans l'équation (IV.11) nous donne le modèle dynamique du bras articulé.

A.1.1 Calcul des éléments D_{ii}

Par l'application directe de (6.84), on obtient :

i=3

$$D_{33} = I_{3zz}$$

i=2

$$D_{22} = I_{2zz} + I_{3zz} + (a_2^2 + a_2 a_3 S_3) m_3.$$

i=1

$$D_{11} = I_{1yy} + S_2^2 I_{2xx} + C_2^2 I_{2yy} + m_3 (a_2 C_2)^2 + S_{23}^2 I_{3xx} + C_{23}^2 I_{3yy} + m_3 a_2 a_3 C_{23} C_2.$$

A.1.2 Calcul des termes D_{ij} ($i \neq j$)

Compte tenu de la remarque (a), on a $D_{ij} = D_{ji}$. Il nous reste à calculer trois termes à savoir: D_{12}, D_{13}, D_{23} et ce à partir de (a); On trouve:

$$D_{12} = D_{21} = 0.$$

$$D_{13} = D_{31} = 0.$$

$$D_{32} = D_{23} = \frac{a_2 a_3 m_3 C_3}{2} + I_{3zz}.$$

A.1.3 Calcul des termes de gravitation D_i ($i=1,2,3$)

A partir de (IV.25), on a:

$$D_1 = 0$$

$$D_2 = \frac{a_2}{2} m_2 g C_2 + m_3 g \left(\frac{a_3}{2} C_{23} + a_2 C_2 \right)$$

$$D_3 = \frac{a_3}{2} m_3 g C_{23}$$

A.1.4 Calcul des termes D_{ijk}

Pour $i=1,2,3$; $j=1,2,3$; $k=1,2,3$; les termes D_{ijk} sont au nombre de 27. Cependant, en exploitant la remarque b) ($D_{ijk} = D_{ikj}$), le nombre de termes se réduit à 18.

$$D_{111} = \sum_{p=1}^3 \text{Trace} \left\{ \left(\frac{\partial^2 T_p}{\partial q_1^2} \right) J_p \left(\frac{\partial T_p}{\partial q_1} \right)^T \right\}$$

On trouve que $D_{111} = 0$

$$D_{112} = \sum_{p=2}^3 \text{Trace} \left\{ \left(\frac{\partial^2 T_p}{\partial q_2 \partial q_1} \right) J_p \left(\frac{\partial T_p}{\partial q_1} \right)^T \right\}$$

$$= (\beta_3 - \alpha_3) C_{23} S_{23} + (\beta_2 - \alpha_2) S_2 C_2 + (m_2 - m_3) a_2^2 C_2 S_2 - m_3 \frac{a_2 a_3}{2} S_{233}.$$

mais $\beta_i - \alpha_i = (I_{ixx} - I_{iyy})$, $i=1,2,3$.

d'où:

$$\left\{ \begin{array}{l} D_{122} = (I_{3xx} - I_{3yy}) C_{23} + (I_{2xx} - I_{2yy}) C_2 S_2 + (m_2 - m_3) a_2^2 C_2 S_2 - m_3 \frac{a_2 a_3}{2} S_{223} \\ D_{121} = D_{112} \end{array} \right.$$

$$D_{113} = \text{Trace} \left\{ \left(\frac{\partial^2 T_3}{\partial q_1 \partial q_3} \right) J_3 \left(\frac{\partial T_3}{\partial q_1} \right)^T \right\} = (\beta_3 - \alpha_3) C_{23} S_{23} - m_3 \frac{a_2 a_3}{2} S_{23} C_2$$

On trouve $D_{223} = D_{232} = -\frac{a_2 a_3}{2} m_3 S_{223}$

$$D_{233} = \text{Trace} \left\{ \left(\frac{\partial^2 T_3}{\partial q_3^2} \right) J_3 \left(\frac{\partial T_3}{\partial q_2} \right)^T \right\}$$

On trouve $D_{233} = -m_3 (a_3^2 C_{23} S_{23} + \frac{a_3}{2} S_{223})$

$$D_{311} = \text{Trace} \left\{ \left(\frac{\partial^2 T_3}{\partial q_1^2} \right) J_3 \left(\frac{\partial T_3}{\partial q_3} \right)^T \right\}$$

$$= (\alpha_3 - \beta_3) C_{23} S_{23} + m_3 \frac{a_2 a_3}{2} S_{23} C_2$$

Ainsi $D_{311} = (I_{3yy} - I_{3xx}) C_{23} S_{23} + m_3 \frac{a_2 a_3}{2} S_{23} C_2$

$$D_{312} = \text{Trace} \left\{ \left(\frac{\partial^2 T_3}{\partial q_1 \partial q_2} \right) J_3 \left(\frac{\partial T_3}{\partial q_3} \right)^T \right\}$$

On trouve $D_{312} = D_{321} = 0$

$$D_{322} = \text{Trace} \left\{ \left(\frac{\partial^2 T_3}{\partial q_2^2} \right) J_3 \left(\frac{\partial T_3}{\partial q_3} \right)^T \right\}$$

On trouve $D_{322} = \frac{a_2 a_3}{2} m_3 S_2$

$$D_{313} = \text{Trace} \left\{ \left(\frac{\partial^2 T_3}{\partial q_1 \partial q_3} \right) J_3 \left(\frac{\partial T_3}{\partial q_3} \right)^T \right\}$$

On trouve $D_{313} = D_{331} = 0$

$$D_{323} = \text{Trace} \left\{ \left(\frac{\partial^2 T_3}{\partial q_2 \partial q_3} \right) J_3 \left(\frac{\partial T_3}{\partial q_3} \right)^T \right\}$$

On trouve $D_{323} = D_{332} = 0$

$$D_{333} = \text{Trace} \left\{ \left(\frac{\partial^2 T_3}{\partial q_3^2} \right) J_3 \left(\frac{\partial T_3}{\partial q_3} \right)^T \right\}$$

On trouve $D_{333} = 0$

A.2 Modélisation du bras PUMA à 2 D.D.L par la méthode d'Euler-Lagrange :

Les matrices de transformations sont déjà calculées dans la partie cinématique (chapitre II).

$$T_1 = A_1$$

$$T_2 = A_1.A_2$$

En appliquant la formule générale de l'équation dynamique trouvée dans la partie précédente on trouve les forces généralisées suivantes:

$$\begin{aligned} \Gamma_1 = & \left(\frac{1}{3} m_1 l^2 + \frac{4}{3} m_2 l^2 + m_2 l^2 C_2 \right) \ddot{q}_1 + \left(\frac{1}{3} m_2 l^2 + \frac{1}{2} m_2 l^2 C_2 \right) \ddot{q}_2 \\ & - m_2 l^2 S_2 \dot{q}_1 \dot{q}_2 - \frac{1}{2} m_2 l^2 S_2 \dot{q}_2^2 + \left(m_1 g \frac{1}{2} + m_2 g l \right) C_1 \\ & + \frac{1}{2} m_2 g l C_{12} + b_1 \dot{q}_1 \end{aligned}$$

$$\begin{aligned} \Gamma_2 = & \left(\frac{1}{3} m_2 l^2 + \frac{1}{2} m_2 l^2 C_2 \right) \ddot{q}_1 + \frac{1}{3} m_2 l^2 \ddot{q}_2 + \frac{1}{2} m_2 l^2 S_2 \dot{q}_1^2 \\ & + \frac{1}{2} m_2 g l C_{121} + b_2 \dot{q}_2 \end{aligned}$$

Avec $l_1 = l_2 = l$.

Dans notre configuration on a $g^t = (0, -g, 0, 1)$ dans le repère $R_0 (X_0, Y_0, Z_0)$

b_i : coefficients de frottement visqueux

Γ_i : coordonnées du point i dans le repère R_i

i_i : dynamique des actionneurs négligeable pour la matrice j_i

Les produits d'inertie sont nuls car $R_i (x_i, y_i, z_i)$ est parallèle au repère principal de la liaison i , donc les matrices d'inertie sont données par :

$$j_1 = \begin{bmatrix} \frac{1}{3}m_1l^2 & 0 & 0 & -\frac{1}{2}m_1l \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_1l & 0 & 0 & m_1 \end{bmatrix} ; \quad j_2 = \begin{bmatrix} \frac{1}{3}m_2l^2 & 0 & 0 & -\frac{1}{2}m_2l \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_2l & 0 & 0 & m_2 \end{bmatrix}$$

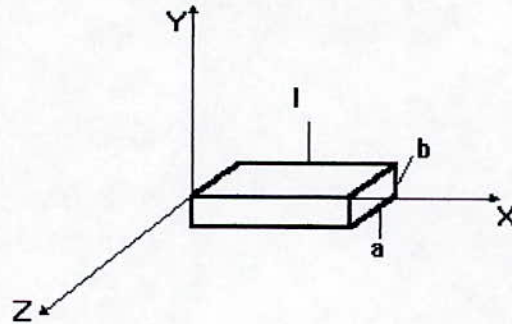


Figure A.2 Segment sous forme de parallélépipède du bras PUMA

Suivant le schéma ci-dessus on a :

$$\int x^2 dm = m l^2/3$$

$$\int y^2 dm = mb^2/3$$

avec $b \ll l$ et $a \ll l$

On suppose que les actionneurs sont idéaux, c'est à dire que la force généralisée est directement proportionnelle au signal de commande :

$$\Gamma_1 = K_1 U_1$$

$$\Gamma_2 = K_2 U_2$$

K_1 , K_2 : les coefficients de proportionnalités des actionneurs

La mise sous forme d'état des équations différentielles non linéaires trouvées nous permet d'utiliser la méthode de RUNGE- KUTTA d'ordre 4 sous les conditions initiales suivantes:

Le vecteur d'état X s'écrit:

$$X^T = [X_1 \quad X_2 \quad X_3 \quad X_4] = [q_1 \quad \dot{q}_1 \quad q_2 \quad \dot{q}_2]$$

Tel que initialement on a:

$$X^T = [0 \quad 0 \quad 0 \quad 0]$$

On détermine alors la forme d'état suivante :

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = \frac{\left[\frac{(K_1 U_1 - f_1 - g_1 - b_1 \dot{q}_1)}{\alpha} + (-K_2 U_2 + f_2 + g_2 + b_2 \dot{q}_2) \frac{\beta}{\alpha \gamma} \right]}{\left[1 - \frac{\beta}{\alpha \gamma} \beta \right]}$$

$$\dot{X}_3 = X_4$$

$$\dot{X}_4 = \frac{\left[\frac{(K_2 U_2 - f_2 - g_2 - b_2 \dot{q}_2)}{\gamma} + (-K_1 U_1 + f_1 + g_1 + b_1 \dot{q}_1) \frac{\beta}{\alpha \gamma} \right]}{\left[1 - \frac{\beta}{\alpha \gamma} \beta \right]}$$

Avec:

$$\alpha = 1/3 l^2 (m_1 + 4 m_2 + 3 m_2 \cos(q_2))$$

$$\beta = m_2 l^2 (1/3 + 1/2 \cos(q_2))$$

$$\gamma = 1/3 m_2 l^2$$

$$g_1 = 1/2 m_1 g l \cos(q_1) + m_2 g l (1/2 \cos(q_1 + q_2) + \cos(q_1))$$

$$g_2 = 1/2 m_2 g l \cos(q_1 + q_2)$$

$$f_1 = -m_2 l^2 \dot{q}_2 \sin(q_2) \left(\frac{1}{2} \dot{q}_2 + \dot{q}_1 \right)$$

$$f_2 = \frac{1}{2} m_2 \sin(q_2) l^2 \dot{q}_1^2$$

A.3 Modélisation du bras T.R.T par la méthode d'Euler-Lagrange :

D'après les matrices de transformation calculées dans la partie cinématique on a

$$T_1 = A_1$$

$$T_2 = A_1.A_2$$

$$T_3 = A_1.A_2.A_3$$

En utilisant l'équation générale de la dynamique on peut calculer les forces généralisées, donc la méthode est la même que celle utilisée pour le robot PUMA.

La matrice J_i est donnée par :

$$j_i = \begin{bmatrix} \alpha_i & 0 & 0 & m_i \bar{x}_i \\ 0 & \beta_i & 0 & m_i \bar{y}_i \\ 0 & 0 & \gamma_i & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix}$$

Suivant la structure suivante de notre bras on a :

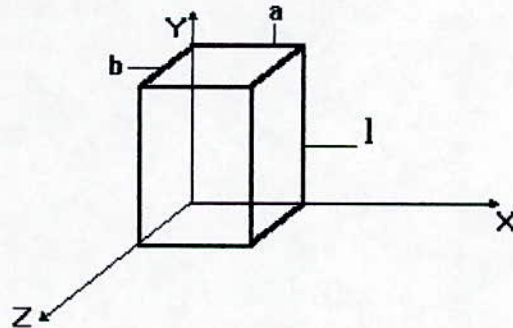


Figure A.3 Segment "i" du bras TRT

$$m_2 \approx 0 \quad ; \quad \bar{x}_i = \bar{y}_i = 0 \quad ; \quad \alpha_2 = \gamma_2 = 0 \quad ; \quad \bar{z}_2 = 0 \quad ; \quad \bar{z}_1 = -\frac{l_1}{2} \quad ; \quad \bar{z}_3 = -\frac{l_2}{2}$$

$$\alpha_3 = \int x^2 dm_3 = m_3 b^2/12$$

$$\gamma_3 = \int z^2 dm_3 = m_3 l_2^2/3$$

Les forces généralisées sont:

$$\Gamma_1 = (m_1 + m_3) \times (\ddot{q}_1 + g) + f_1 \dot{q}_1$$

$$\Gamma_2 = (\alpha_3 + \gamma_3 + d_3 m_3 \bar{z}_3 + m_3 (d_3 \bar{z}_3 + d_3^2)) \ddot{q}_2 \\ + 2m_3 (\bar{z}_3 + d_3) \dot{q}_3 \dot{q}_2 + f_2 \dot{q}_2$$

$$\Gamma_3 = m_3 \ddot{q}_3 - m_3 (\bar{z}_3 + d_3) \dot{q}_2^2 + f_3 \dot{q}_3$$

f_i : représente les coefficients de frottement visqueux

du fait que $b \ll l$ on a : $\alpha_3 \approx 0$

La force généralisée est directement proportionnelle au signal de commande, en prenant:

$$\Gamma_1 = K_1 U_1$$

$$\Gamma_2 = K_2 U_2$$

$$\Gamma_3 = K_3 U_3$$

L'équation d'état des équations différentielles non linéaires est:

$$\dot{X} = AX + BU + D$$

$$Y = CX$$

$$\dot{X} = [X_1 \ X_2 \ X_3 \ X_4 \ X_5 \ X_6] = [q_1 \ \dot{q}_1 \ q_2 \ \dot{q}_2 \ q_3 \ \dot{q}_3]$$

avec :

$$Y^T = [y_1 \ y_2 \ y_3] = [q_1 \ q_2 \ q_3] \quad : \quad \text{vecteur de sorties}$$

$$U^T = [U_1 \ U_2 \ U_3] \quad : \quad \text{vecteur de commande}$$

et

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{f_1}{m_1 + m_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{f_2}{m_3 j^*} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{-f_3}{m_3} \end{bmatrix} ; \quad B = \begin{bmatrix} 0 & 0 & 0 \\ \frac{K_1}{m_1 + m_3} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{K_2}{m_3 j^*} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{K_3}{m_3} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$d^t = [0 \ -g \ 0 \ -2.(X_5 - l_2/2).X_4.X_6/j^* \ 0 \ (X_5 - l_2/2).X_4^2 \]$$

$$j^* = (l_2)^2/3 - l_2.X_5 + X_5^2$$

Bibliographie

Références bibliographiques

- [COI 95] COIFFET.C, "*Robots : définitions et classement* ", *Techniques de l'ingénieur (Traité Mesure et Contrôle)*, Doc R7700, Paris, 1995.
- [CRA 86] CRAIG.J, "*Introduction to robotics : Mechanics and control* ", Ed.Addison Wesley, 1986.
- [DEL 95] DELANNOY.C, "*Programmer WINDOWS avec TURBO C++* ", Editions CHIHAB, Alger, 1995.
- [DJE 96] N.DJEDDI et D.DRIBINE, "*Modélisation cinématique et dynamique de bras manipulateurs* ", Thèse d'ingénieur, ENP, 1996.
- [FU 87] FU.S.K, GONZALES R.C, LEE C.S.G, "*Robotics :Control, Sensing, Vision and intelligence* ", Mac Graw Hill Company, 1987.
- [GOR 84] GORLA.P et RENAUD.M, "*Modèle des robots manipulateurs, application à la commande* ", Cepadues-Editions, Paris, 1984.
- [GUE 95] GUENFAF.L, "*Etude comparative de différentes stratégies de commande adaptative : Application à un robot manipulateur* ", Thèse de Magister, ENP, 1995.
- [KOR 84] KOREN.Y, "*Robotics for Engineers* ", Mac Graw Hill, 1984.
- [LALL 94] LALLEMAND.J.P et ZEGHLOUL.S, "*Robotique : aspects fondamentaux, Modélisation mécanique, C.A.O Robotique et commande*", Editions MASSON, 1994.
- [LEV 92] LEVINE.A, "*TURBO C : INITIATION* ", BERTI Editions, Alger, 1992.
- [MAT 93] "*MATLAB User's Guide : High-Performance Numeric Computation and Visualisation Software* ", The MATH WORKS Inc, February 1993.
- [PAU 81] PAUL.R.P, "*Robot manipulators : Mathematics, Programming and Control* ", M.I.T Press, Cambridge, Massachussets and London, 1981.
- [SCH 87] SCHWEIZER.P, "*Infographie II* " , Collection infographique, Presses Polytechniques Romandes,1987.
- [STO 89] STOTEN.D.P, "*Generalized manipulator dynamics with regard to model reference adaptive control* ", INT.J.Control, Vol.50, N°6, pp.2245-2268, 1989.
- [VIB 87] VIBET.C, "*Robots : Principes et contrôle* " , Ellipses, 1987.

