

Ministère de L'enseignement Supérieur et de La Recherche Scientifique

Ecole Nationale Polytechnique

Laboratoire de Commande des Processus



## *Mémoire*

Présentée par :

*HAOUARI FOUAD*

Ingénieur d'état en Automatique de l'UMBB

Pour obtenir le titre de

**Magister en Automatique**

*Option : Systèmes Intelligents de Commande et Robotique*

Thème

## **Commande Robuste d'un Robot Manipulateur PUMA Basée sur la Méthode du Diagramme des Coefficients et de la GPC**

Soutenus publiquement le 04.07.2006 devant le jury composé de

Président :	M.O MAHMOUDI	Maître de Conférence ENP
Rapporteur :	M.S BOUCHERIT	Professeur ENP
	N. BALI	Chargé de cours, USTHB
Examineurs :	M. TADJINE	Maître de conférence ENP
	B. HEMICI	Chargé de cours ENP
	A. TALHA	Docteur, USTHB

Année Universitaire 2006

# Dédicace

Je dédie ce modeste travail à ma chère mère, à mon cher père, qui m'ont soutenus et encouragés durant toutes mes études et nulle chose ne récompensera leurs sacrifices ainsi qu'à tous mes amis les plus proches.

# Remerciements

Nous remercions dieu pour nous avoir donné la force morale et physique pour accomplir ce modeste travail.

Qu'il nous soit permis d'adresser notre sincère reconnaissance à notre encadreur Mr M.S BOUCHERIT pour la confiance qu'il nous a témoignée tout au long de ce travail, pour son soutien et son souci de voir ce travail terminé, qu'il trouve ici l'expression de notre profonde gratitude.

Nous remercions vivement Mr N.Bali pour toutes les suggestions et les discussions enrichissantes dont il nous a fait bénéficier au cours de notre travail.

Nous adressons également nos sincères remerciements à Mr. M.TADJINE pour l'aide qu'il nous a apporté.

Nous tenons à présenter nos remerciements aussi à toutes les personnes qui ont participé de près ou de loin à l'élaboration du présent mémoire.

Ainsi au président de jury et les membres de jury ayant accepté de le juger.

# *Sommaire*

## SOMMAIRE

<b>Introduction générale</b> .....	1
------------------------------------	---

### **Chapitre 1 : Modélisation de Robot Manipulateur**

1.1 Introduction .....	3
1.2 présentation de robot.....	3
1.3 Modélisation cinématique.....	3
1.3.1 problème cinématique direct.....	4
1.3.2 Application au PUMA à 2 d d l.....	6
1.3.3 Modélisation géométrique.....	7
1.4 Génération de trajectoire .....	8
1.5 Modélisation dynamique .....	10
1.5.1 Approche de LAGRANGE-EULER .....	10
1.5.1.1 Energie cinétique du robot manipulateur.....	11
1.5.1.2 Energie Potentiel du Bras manipulateur .....	12
1.5.1.3 Equation dynamique du bras manipulateur .....	13
1.5.1.4 Application au PUMA à 2 d d l.....	14
1.5.2 Approche de NEWTON-EULER .....	15
1.5.2.1 Application au repère liées aux liaisons d'un robot.....	16
1.5.2.2 Equation récursives du mouvement du robot.....	17
1.5.2.3 Equations récursives modifiées.....	18
1.5.2.4 Algorithme de NEWTON-EULER .....	19
1.6 Résultat du simulation .....	19
1.7 Validation Des modèles dynamiques .....	21
1.8 Linéarisation.....	23
1.8.1 Linéarisation entrée/sortie.....	23
1.8.2 Linéarisation exacte entrées/sorties.....	24
1.8.3 Application au robot manipulateur PUMA.....	26
1.9 Conclusion .....	27

### **Chapitre 2 : Etude des performances de la commande CDM**

2.1 Introduction.....	28
2.2 Développement de la commande : méthode de diagramme des coefficients .....	28
2.3 Le polynôme caractéristique de système en boucle fermé .....	28
2.4 Le diagramme de coefficients.....	30
2.5 Les conditions de stabilité et de robustesse.....	30
2.6 La forme standard de CDM .....	31
2.7 Algorithme de commande CDM .....	31

2.8 Résultats de simulation.....	31
2.8.1 Etude d'un système stable .....	32
2.8.2 Etude d'un système stable à phase non minimale .....	36
2.8.3 Etude d'un système instable .....	39
2.8.4 Etude d'un système instable à phase non minimale .....	43
2.9 Influence de la sortie à une perturbation, en régime établi.....	46
2.10 Synthèse et discussion des résultats obtenus.....	48
2.11 Conclusion.....	49

### **Chapitre 3 : Application de la commande CDM au bras manipulateur PUMA**

3.1 Introduction.....	50
3.2 Application de la commande CDM au robot manipulateur PUMA.....	50
3.3 Influence des paramètre de CDM sur la commande de robot.....	52
3.4 Résultats de simulation.....	56
3.5 Conclusion.....	59

### **Chapitre 4 : Commande Prédictive généralisée**

4.1 Introduction.....	60
4.2 Développement de la commande prédictive généralisée.....	60
4.2.1 Modélisation.....	61
4.2.2 Prédiction de la sortie.....	61
4.3 Résolution itérative de l'équation de DIOPHANTINE.....	63
4.4 Loi de Commande Prédictive.....	64
4.5 Critère Quadratique .....	64
4.6 Calcul du prédiction a j-pas .....	65
4.7 Minimisation du critère.....	66
4.8 Introduction des horizons $N_1, N_2, N_u, \lambda$ .....	67
4.9 Conclusion.....	68

### **Chapitre 5 : Application de la GPC au Robot manipulateur PUMA**

Introduction.....	69
5.1 Modèle de représentation .....	69
5.2 Résultats de simulation.....	70
5.3 Synthèse et discussion.....	74
5.4 Comparaison entre la commande CDM et GPC.....	75
5.5 Conclusion.....	75

<b>Conclusion générale.....</b>	<b>76</b>
---------------------------------	-----------

*INTRODUCTION*  
*INTRODUCTION*  
*GENERALE*  
*GENERALE*

## Introduction générale

Les préoccupations de l'automatique issue de la régulation de processus comportant quelque entrées et quelque sorties ont évolué, vers la conduite de vastes ensembles technologiques appelée grand systèmes, dont les réseaux de distribution, eau, gaz, télécommunication, réseaux d'ordinateurs, ou les ateliers flexibles sont des exemples typiques.

Un modèle de connaissance basé sur les lois physiques régissant l'ensemble de la structure de robot est élaboré, ce modèle est obtenu par l'utilisation des formalismes de LAGRANGE-EULER (L-E) et de NEWTON-EULER (N-E) [1] aboutissant à un modèle dynamique du manipulateur fortement non linéaire [2]. Des hypothèses simplificatrices sont introduites pour obtenir un modèle réduit capable de reproduire le comportement physique du système considéré [3]. Pour pouvoir appliqué les lois de commande CDM et GPC qui sont développées sous la base d'un modèle linéaire, l'introduction de la technique de linéarisation exacte de robot est nécessaire [4][5].

En 1998 la technique de commande méthode de diagramme des coefficients (CDM) a été introduite par MANABE [6], figiolée en 2002 [7], elle se base sur des concepts connus depuis 1960 et utilisés dans la méthode de NASLIN et KESSLER [8]. Cette méthode de commande est classée entre les méthodes classiques et les méthodes modernes [9], elle permet la synthèse d'un correcteur robuste, qui répond aux spécifications imposées simplement par un choix judicieux de polynôme caractéristique [10].

La synthèse de la loi de commande est basée sur l'utilisation d'un outil appelé diagramme des coefficients [11]. Il permet de choisir judicieusement le polynôme caractéristique en boucle fermée, Ce polynôme assure de bonnes performances statiques et dynamiques, robustesse vis-à-vis de variation paramétrique et l'incertitude paramétrique [12]. La démarche de synthèse se résume dans la détermination de la constante de temps équivalente du système comme paramètre de réglage à partir de temps de réponse en boucle fermée puis on définit les indices de stabilité qui permettent de définir le polynôme caractéristique assurant ainsi les performances désirées [13][14]. En choisissant une structure pour le correcteur CDM suivant les objectif désirées et par identification avec le polynôme caractéristique obtenu, on détermine les paramètres de correcteur CDM. L'application de CDM au robot manipulateur PUMA a montré que malgré la dynamique imposée par le modèle du robot en boucle ouverte de très bons résultats ont été obtenu et ceci par un choix spécifique des paramètres de synthèse de CDM [15].

La deuxième technique de commande étudiée est la commande prédictive généralisée (GPC) [16], L'objectif de cette loi de commande prédictive consiste à prendre en compte, a l'instant présent le comportement futur [17], en utilisons explicitement un modèle numérique du système afin de prédire dans le futur sur un horizon fini, cependant il n'existe pas une stratégie unique, mais plutôt tout un ensemble de méthodes de commande prédictive, assez similaire, bâti autour de principe connus, mais présentant néanmoins quelque différences dans l'interprétation des concepts clés [18][19], La détermination de la loi de commande est basée sur la minimisation d'un critère quadratique multipas au sens de moindre carrés [20]. L'application de la GPC au robot manipulateur PUMA a donné de bonnes performances grâce aux choix des paramètres de synthèse [21].



Ce travail est réparti sur cinq chapitres consacrés à l'étude des performances de la CDM et de la GPC ainsi que leurs application au robot PUMA, on trouve dans chaque chapitre une introduction, une conclusion et des résultats de simulation validons le développement théorique.

Le chapitre un est consacré à la modélisation de robot manipulateur PUMA. La méthode d'E-L nous permet d'établir le modèle dynamique de robot, dans le but de valider ce modèle, la méthode de N-E est étudiée. Les réponses en boucle ouverte sont obtenues pour le suivi d'une trajectoire circulaire.

Le chapitre deux traite la technique de commande CDM. On présente le diagramme des coefficients qui est un outil d'analyse des performances et permet d'apporter des modifications nécessaires pour améliorer davantage les résultats. L'efficacité de l'algorithme est tester sur quatre types de systèmes.

Le chapitre trois est consacré à l'application de la commande CDM au robot manipulateur PUMA et l'influence des paramètres de cette technique de commande sur le robot manipulateur.

Le chapitre quatre traite le développement de l'algorithme de base de la GPC. La commande est élaborée à partir de la minimisation d'un critère quadratique multipas basé sur l'écart entre la sortie prédite et la référence.

Le chapitre cinq traite l'application de la commande prédictive généralisée au robot manipulateur PUMA et l'influence des paramètres de cette technique sur le modèle de robot.

Enfinement nous terminons cette étude par une conclusion générale.

# CHAPITRE I

# CHAPITRE I

Modélisation de robot manipulateur PUMA

## 1.1 Introduction

En automatique la modélisation constitue une étape indispensable dans la commande des systèmes dynamiques. En effet, il existe deux façons d'élaborer les relations d'entrées/sorties du système : le modèle de connaissance et le modèle de représentation, le premier est obtenu en utilisant les lois physiques [1], tandis que le deuxième est obtenu en utilisant les techniques d'identification en se basant sur des observations faites sur les entrées et les sorties du système.

Dans le cas de la robotique les concepts fondamentaux de la mécanique, tels que les formalismes de N-E et L-E [1] peuvent être utilisés indifféremment pour obtenir les équations différentielles régissant le mouvement dynamique du robot.

La robotique est une technique fondamentalement pluridisciplinaire car elle utilise de nombreuses technologies : mécanique, pneumatique, hydraulique, électronique, informatique etc., elle est donc une notion vaste, dont une définition universelle semble impossible à donner bien que de nombreuses tentatives ont été faites. La définition la plus précise à notre égard est celle de l'AFNOR, qui définit ce qu'est un manipulateur : « mécanisme multi-fonctionnel à plusieurs degrés de libertés commandé directement par un opérateur humain ou par un système logique ».

Le robot industriel est présenté comme un manipulateur automatique asservi en position, reprogrammables, polyvalent et capable de positionner des matériaux, pièces, outils ou dispositifs spécialisés. Il est aussi présent dans les environnements hostiles, où il est destiné à accomplir des tâches dont l'homme est incapable de les effectuer « manipulation des matériaux radioactifs dans les cellules chaudes des réacteurs nucléaires, etc.. »

## 1.2 Présentation de robot

Un robot manipulateur est caractérisé par une structure arborescente articulée simple ou multiple dont les segments sont mobiles les uns par rapport aux autres, cet ensemble à pour objectif de mener l'organe terminal vers un lieu géométrique imposé par la tâche.

Le robot manipulateur que nous nous proposons de modéliser est le PUMA qui est caractérisé par deux articulations rotationnelles  $\theta_1$  et  $\theta_2$ , pour pouvoir établir les différents modèles, plusieurs hypothèses doivent être prises en considération :

- Les frottements sont de nature visqueuse, et linéaire par rapport à la vitesse généralisée.
- Les différentes liaisons sont rigides.
- Les actionneurs sont idéaux, c'est à dire que la force généralisée est directement proportionnelle au signal de commande.
- Les capteurs ont des gains unitaires, et de dynamique négligeable.

## 1.3 Modélisation cinématique

Un manipulateur est considéré comme étant une chaîne de liaisons connectées entre elles par des articulations rotationnelles ou prismatiques. Une extrémité de cette chaîne est attachée à la base, tandis que l'autre extrémité est libre et reliée à l'outil terminal, en vue de manipuler les objets ou effectuer des tâches d'assemblages. Le mouvement relatif de chaque articulation

entraîne celui des liaisons, ce qui permet de positionner l'extrémité libre et de donner à l'outil terminal l'orientation désirée. En effet, dans la plupart des applications en robotique, on s'intéresse à la trajectoire de l'outil terminal par rapport à un référentiel fixe et arbitraire. La cinématique d'un robot se veut l'étude analytique de cette trajectoire sans celle des forces et moments qui causent le mouvement.

Dans ce qui suit on va considérer un manipulateur dont les paramètres géométriques sont supposés connus et on va essayer de répondre aux deux questions suivantes :

1. pour un manipulateur dont les paramètres géométriques sont supposés connus, pour chaque vecteur des variables d'articulations  $[q_1 \ q_2 \ q_3 \dots \dots \dots q_n]^T$ ,  $n$  étant le nombre de degrés de liberté (d d l) du manipulateur, quelle est la position et l'orientation de l'outil terminal correspondante, par rapport au référentiel fixe (la base) ?
2. soit une position et orientation désirées de l'outil terminal, le manipulateur peut il atteindre celle-ci ? si oui, quelles sont les différentes configurations du manipulateur, c'est à dire l'ensemble des vecteurs  $q$ , qui peuvent nous le garantir ?

la première de ces questions est appelée problème cinématique direct, et la seconde problème appelée cinématique inverse, seul l'approche directe est traitée du fait qu'elle intervient directement dans la modélisation dynamique des robots.

### 1.3.1 Problème cinématique direct [2]

Pour décrire les relations de translations et de rotations entre les liaisons adjacents d'un robot manipulateur DENAVIT & HARTENBERG (D-H) (1955) [2] Proposèrent une méthode matricielle pour établir systématiquement les repères attachés à chaque liaison dans une chaîne articulée. La représentation de D-H consiste en l'utilisation d'une transformation homogène  $4 \times 4$  qui représente le système de coordonnées de chaque liaison par rapport au système de coordonnées de liaison précédente et cela, à travers d'une séquence de transformations on peut ramener le système de coordonnées attaché à l'élément terminale (la tache) au système de coordonnées attaché à la base (repère fixe), figure 1.1.

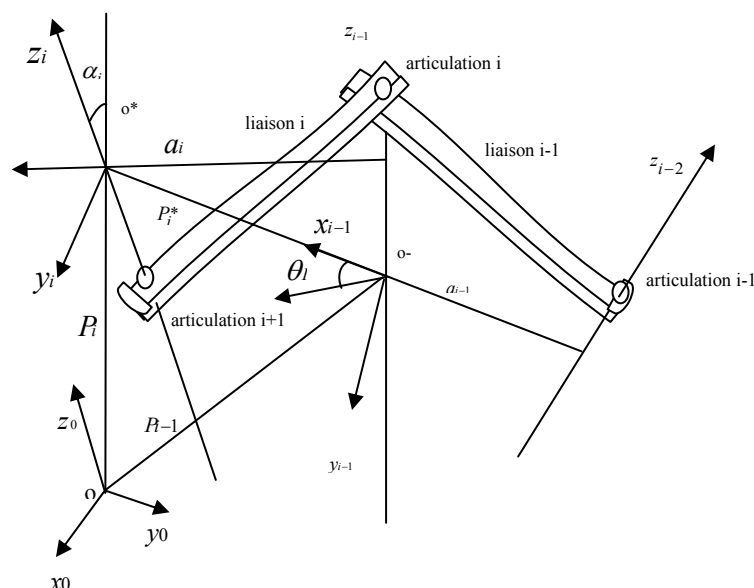


figure 1.1 relation entre divers repères d'une chaîne articulée

Les repères sont déterminés selon les règles suivantes :

- L'axe  $Z_{i-1}$  est le long de l'axe de l'articulation  $i$
- L'axe  $X_i$  est normal à l'axe  $Z_{i-1}$  :  $X_i = Z_{i-1} \wedge Z_i$ .
- L'axe  $Y_i$  est déterminé de telle manière à constituer un trièdre direct.

Il est à noter que le choix du référentiel de base est arbitraire, pour autant que l'axe  $Z_0$  soit le long de l'axe de mouvement de l'articulation 1, de même le référentiel de l'outil terminal pourra être placé n'importe où, tant que l'axe  $X_n$  est normal à  $Z_{n-1}$ . Il est maintenant possible de définir les paramètres cinématique, de la manière suivante :

1.  $\theta_i$  est l'angle de l'axe  $X_{i-1}$  vers l'axe  $X_i$  autour de l'axe  $Z_{i-1}$
2.  $\alpha_i$  est l'angle de l'axe  $Z_{i-1}$  vers l'axe  $Z_i$  autour de l'axe  $X_i$
3.  $a_i$  est la distance comprise entre l'intersection de l'axe  $Z_{i-1}$  avec l'axe  $X_i$  et l'origine du référentiel  $i$  le long de l'axe  $X_i$
4.  $d_i$  est la distance comprise entre l'origine du référentiel  $i-1$  et l'intersection de l'axe  $Z_{i-1}$  avec l'axe  $X_i$  le long de l'axe  $Z_{i-1}$

Enfin, on va présenter dans ce qui suit l'algorithme qui permet de traiter n'importe quelle configuration d'un robot manipulateur de  $n$  d d l

1. Etablir le système de coordonnées lié à la base, en s'assurant que l'axe  $Z_0$  est bien selon l'axe de rotation de l'articulation 1, et qu'il pointe vers « l'épaule » du bras manipulateur.
2. pour chaque articulation  $i=1, \dots, n-1$  faire les étapes 3 à 6.
3. aligner l'axe  $Z_i$  avec l'axe de l'articulation  $i+1$ .
4. Etablir l'origine de chaque système de coordonnées, celui-ci sera donnée par l'intersection de l'axe  $Z_{i-1}$  et  $Z_i$  ou bien l'intersection de leurs normale avec l'axe  $Z_i$ .
5. Etablir l'axe  $X_i$  selon la règle suivante :  $X_i = \pm (Z_{i-1} \wedge Z_i) / \| Z_{i-1} \wedge Z_i \|$  ou bien le long de la normale commune aux axes  $Z_i$  et  $Z_{i-1}$ .
6. Etablir l'axe  $Y_i$ , selon la règle suivante :  $Y_i = \pm (Z_i \wedge X_i) / \| Z_i \wedge X_i \|$ , afin d'établir le trièdre direct  $(X_i, Y_i, Z_i)$ .
7. Etablir l'axe  $Z_n$  selon la direction de l'axe  $Z_{n-1}$ , et en sortant du corps de robot
8. déterminer les paramètres de chaque articulation :  $i=1, n-1$ .

Une fois ces étapes effectuées, on détermine les matrices de transitions relatives à chaque articulation, elle sont données en fonction des paramètres décrits ci-dessus et en remarquant qu'un point  $n$  exprimé dans le référentiel  $i$ , peut être exprimé dans le référentiel  $i+1$  en opérant les transformations successives suivantes :

1. Rotation autour de l'axe  $Z_{i-1}$  d'un angle  $\theta_i$  afin d'aligner l'axe  $X_{i-1}$  avec l'axe  $X_i$ .
2. Translation le long de l'axe  $Z_{i-1}$  d'une distance  $d_i$ , afin de faire coïncider  $X_i$  et  $X_{i-1}$ .
3. Translation le long de l'axe  $X_i$  d'une distance  $a_i$  afin de faire coïncider les origine des axes  $X_i$  et  $X_{i-1}$
4. Rotation autour de l'axe  $Z_{i-1}$  d'un angle  $\theta_i$  afin de faire coïncider les deux systèmes de coordonnées  $i$  et  $i-1$ .

Ce qui donne la matrice de transformation homogène suivante :

$$A_i^{i-1} = T_{Z,d} \times R_{Z,\theta} \times T_{X,a} \times R_{X,\alpha} \quad (1.1)$$

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & -\sin\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

On va appliquer l'algorithme donné ci-dessus sur le PUMA à 2 d d l

### 1.3.2 Application au PUMA à 2 d d l

Dans ce travail, nous avons opté pour un robot planaire PUMA à 2 d d l [3] (seule les articulation 1 et 2 du robot industriel PUMA 500 sont prise en compte). Le schéma descriptif de ce manipulateur est donné à la figure 1.2

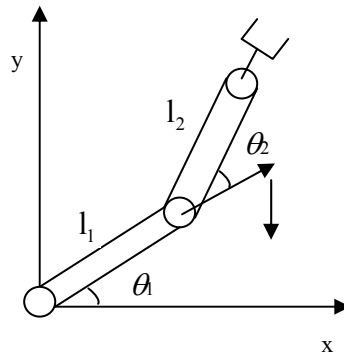


figure 1.2 Présentation du PUMA à 2 d d l

Les paramètres de la transformation D-H du robot, sont regroupés dans le tableau suivant :

articulation	$\theta_i$	$a_i$	$d_i$	$\alpha_i$
1	$\theta_1$	$l_1$	0	0
2	$\theta_2$	$l_2$	0	0

Tableau 1.1 Paramètre de D-H pour le PUMA à 2 d d l

L'implémentation des repères pour chaque liaison, en suivant les étapes de l'algorithme cité ci-dessus, et illustrée sur la figure 1.3

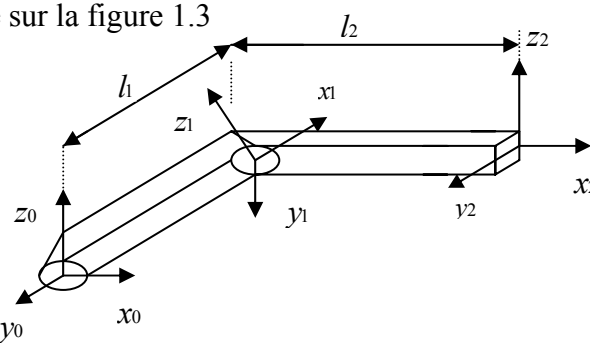


figure 1.3 Implémentation de la méthode de D-H

Ainsi en utilisant l'algorithme de la représentation de D-H nous aboutissons aux matrices de transformations homogènes suivantes :

$$A_1^0 = \begin{bmatrix} c1 & -s1 & 0 & l1c1 \\ s1 & c1 & 0 & l1s1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2^1 = \begin{bmatrix} c2 & -s2 & 0 & l2c2 \\ s2 & c2 & 0 & l2s2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2^0 = \begin{bmatrix} c12 & -s12 & 0 & l(c1+c12) \\ s12 & c12 & 0 & l(s1+s12) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

avec :  $c_i = \cos\theta_i$ ,  $s_i = \sin\theta_i$ ,  $c_{12} = \cos(\theta_1 + \theta_2)$ ,  $s_{12} = \sin(\theta_1 + \theta_2)$ .

### 1.3.3 Modélisation géométrique

Les transformations précédentes peuvent être utilisées pour établir les relations entre les coordonnées généralisées. En effet, le modèle géométrique consiste à résoudre l'équation de transformation afin de trouver la position de l'élément terminale en terme d'angles et de déplacements entre les articulations. Les angles et déplacements entre les articulations sont appelés : coordonnées articulaires alors que les positions de l'élément terminale « outil de travail » dans l'espace sont décrites comme étant des coordonnées opérationnelles ou cartésiennes.

Il y'a deux types de modèle géométrique le modèle géométrique direct et inverse :

Le modèle géométrique direct qui s'obtient on égalisant la matrice de transformation homogène qui relie l'élément terminale à la base ( $A_0^2$  dans notre cas) et la matrice de transformation générale.

$$A_2^0 = \begin{bmatrix} X_x & Y_x & Z_x & P_x \\ X_y & Y_y & Z_y & P_y \\ X_z & Y_z & Z_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$

$(X_x \ X_y \ X_z)^T$  est l'orientation de l'élément terminale suivant l'axe  $X$  de la base.

$(Y_x \ Y_y \ Y_z)^T$  est l'orientation de l'élément terminale suivant l'axe  $Y$  de la base.

$(Z_x \ Z_y \ Z_z)^T$  est l'orientation de l'élément terminale suivant l'axe  $Z$  de la base.

$(P_x \ P_y \ P_z)^T$  est la position de l'élément terminale.

On trouve donc :

$$\begin{cases} x = P_x = l(c_1 + c_{12}) \\ y = P_y = l(s_1 + s_{12}) \\ z = P_z = 0 \end{cases} \quad (1.5)$$

Cette équation est appelé modèle géométrique direct de robot manipulateur. Elle peut être obtenu facilement en utilisons les relations trigonométrique et en se référons à la figure 1.2. Le modèle géométrique inverse est obtenu en inversons l'équation (1.5), Il donne la configuration géométrique du robot en fonction de la position de l'élément terminal et ne conduit pas toujours à une solution unique. En partant de (1.5), on obtient :

$$x^2 + y^2 = 2l^2(1 + c_2) \quad (1.6)$$

donc

$$\theta_2 = \arctg\left(\frac{\sqrt{1 - A^2}}{A}\right) \quad (1.7)$$

avec

$$A = \frac{x^2 + y^2}{2l^2} - 1 \quad (1.8)$$

d'autre part, on a :

$$x = l(1+c_2)c_1 - ls_1s_2 \quad (1.9)$$

$$y = l(1+c_2)s_1 + lc_1s_1 \quad (1.10)$$

$$\frac{y}{x} = \frac{\left(\frac{s_1}{c_1}\right) + \frac{s_2}{1+c_2}}{1 - \frac{s_1s_2}{c_1(1+c_2)}} \quad (1.11)$$

En utilisons la relation suivante :  $\arctg(a) + \arctg(b) = \arctg\left(\frac{a+b}{1-ab}\right)$

on obtient

$$\theta = \arctg\left(\frac{x}{y}\right) - \arctg\left(\frac{s_2}{1+c_2}\right) \quad (1.12)$$

avec

$$a = \frac{s_1}{c_1} = \text{tg}(\theta) \quad \text{et} \quad b = \frac{s_2}{1+c_2} \quad (1.13)$$

Les équations (1.7) et (1.12) représentent le modèle géométrique inverse de robot manipulateur

## 1.4 Génération de trajectoire

En se basant sur la définition de robots donnée précédemment, à savoir que l'organe terminal de robot PUMA accomplit un certain mouvement dans l'espace établi au préalable par l'opérateur. Le mouvement ainsi réalisé est appelé trajectoire. En effet, à chaque point de coordonnées généralisées correspondantes suivant la trajectoire décrite par l'élément terminal, on peut trouver une relation entre les coordonnées généralisées ( $q_i$ ) et les paramètres introduits pour décrire cette même trajectoire.

La trajectoire désirée pour que le robot l'exécute est un cercle dans le plan  $(O, X, Y)$  situé à une distance  $(C_{X_0}, C_{Y_0})$  de repère  $R_0$  du robot et de rayon R.

les équations du cercle dans le repère  $(O, X, Y)$  sont :

$$\begin{cases} X = C_{X_0} + R \cos(\omega t) \\ Y = C_{Y_0} + R \sin(\omega t) \end{cases} \quad (1.13)$$

donc

$$\begin{cases} C_{X_0} = (2l - R) \cos\left(\frac{\pi}{4}\right) \\ C_{Y_0} = (2l - R) \cos\left(\frac{\pi}{4}\right) \end{cases} \quad (1.14)$$

A partir des coordonnées généralisées établies pour un point quelconque dans l'espace, on détermine leurs évolutions en fonction du temps pour une trajectoire circulaire, ce qui revient à calculer  $\theta_1$  et  $\theta_2$  pour chaque point de cercle.

Etant donné que nous avons  $\theta_1$  et  $\theta_2$ , équations (1.7) (1.12), proposons nous de déterminer les vitesses et accélérations de trajectoire circulaire à partir des coordonnées généralisées du modèle géométrique inverse.



$$\begin{cases} \theta_2 = q_{d2} = \arccos \left[ \frac{1}{2} \left( \frac{r}{l} \right)^2 - 1 \right] \\ \theta_1 = q_{d1} = \varphi - \arcsin \left[ \frac{l}{r} \sin(\pi - q_{d2}) \right] \end{cases} \quad (1.15)$$

où l'indice d veut dire " désiré " .

sachant que

$$\begin{cases} \dot{x} = -R\omega \sin(\omega t) \\ \dot{y} = R\omega \cos(\omega t) \end{cases} \quad (1.16)$$

On détermine après quelques calculs la dérivée simple et double de vitesse et d'accélération généralisés  $\dot{q}_{d2}$  et  $\ddot{q}_{d2}$

avec

$$\begin{cases} q_{d2} = \arccos(\mu) \\ \mu = \frac{1}{2} \left( \frac{l}{r} \right)^2 - 1 \\ \dot{\mu} = \frac{r\dot{r}}{l^2} \end{cases} \quad (1.17)$$

$$\begin{cases} \dot{q}_{d2} = -\frac{\mu}{\sqrt{1-\mu^2}} \\ \ddot{q}_{d2} = -\frac{(\dot{r}^2 + \ddot{r}r)(1-\mu^2) + \mu\dot{\mu}r\dot{r}}{(1-\mu^2)\sqrt{1-\mu^2}} \end{cases} \quad (1.18)$$

La même méthode est appliquée pour  $\theta_1$  , on obtient alors

$$\begin{cases} \dot{q}_{d1} = \dot{\varphi} - \dot{\phi}_1 \\ \ddot{q}_{d1} = \ddot{\varphi} - \ddot{\phi}_1 \end{cases} \quad (1.19)$$

avec

$$\begin{cases} \varphi = \arctan\left(\frac{x}{y}\right) \\ \phi_1 = \arcsin\left[\frac{l}{r} \sin(\pi - q_{d2})\right] \\ \ddot{\varphi} = \frac{(\dot{y}x - \ddot{y}y)r - 2\dot{r}(\dot{y}x - \dot{x}y)}{r} \\ \ddot{\phi}_1 = \frac{\ddot{\theta}\sqrt{1-\theta^2} + \dot{\theta}\phi_1\theta}{1-\theta^2} \\ \theta = \frac{l}{r} \sin(\pi - q_{d2}) \end{cases} \quad (1.20)$$

### Remarque

Le comportement dynamique du robot exige d'imposer des trajectoires réalisables, Le choix d'une trajectoire est relié à l'évolution de la position, la vitesse et l'accélération. En effet,

la continuité en position, vitesse et accélération offre au robot la possibilité de poursuivre cette trajectoire avec de commandes réalisables. La décomposition de la tâche en plusieurs points intermédiaires, nécessite une continuité de premier et second type.

La connaissance des conditions aux limites de différents types permet le calcul direct des paramètres des polynômes du troisième degré permettant une continuité en position et en vitesse. Cette tâche se traduit par le passage d'un état d'équilibre à un autre état d'équilibre.

## 1.5 Modélisation dynamique

Les équations dynamiques du mouvement d'un bras manipulateur, sont l'ensemble des équations mathématiques décrivant le comportement dynamique du manipulateur, de telles équations sont utilisées pour la simulation par ordinateur, du mouvement du bras manipulateur, ainsi que pour la conception d'une commande en vue de satisfaire les performances imposées.

Le modèle du bras manipulateur, peut être déterminé à partir des approches conventionnelles, l'approche de L-E ou celle de N-E, ainsi peuvent être systématiquement appliquées en vue de déterminer le modèle dynamique de robot manipulateur, cependant, il est à noter qu'il existe plusieurs variantes de ces deux approches telles que « UICKER'S-L-E », « HOLLERBACH'S RECURSIVE LAGRANGE », « LEE'S GENERALIZED D'ALEMBERT G.D »...etc. Ces approches peuvent aboutir à des équations dont la structure est différente, car elles ont été conçues pour différentes tâches. Dans ce qui suit, on va uniquement présenter les approches de L-E et N-E.

### 1.5.1 Approche de LAGRANGE-EULER

Les équations générales du mouvement de bras manipulateur peuvent être convenablement exprimées par l'application directe du formalisme de L-E, celui-ci associé à la représentation de D-H, peut constituer un algorithme efficace et compact afin d'aboutir aux équations décrivant le mouvement de bras manipulateur.

L'algorithme est exprimé sous forme matricielle, ce qui facilite l'analyse et l'implémentation sur l'ordinateur. La détermination des équations dynamiques d'un bras manipulateur à n d d l est basée sur le fait que la matrice homogène de transformation relie le référentiel de la liaison i et celui de la liaison i-1.

L'équation de L-E est donnée par :

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial (\dot{q}_i)} \right] - \frac{\partial L}{\partial q_i} = \tau_i \quad (1.21)$$

où

L : fonction de lagrange =  $E_c - E_p$ .

$E_c$  : Energie cinétique totale du bras manipulateur.

$E_p$  : Energie potentielle totale du bras manipulateur.

$q_i$  :  $i^{ieme}$  coordonnées généralisée.

$\frac{dq_i}{dt}$  : dérivée première par rapport au temps  $q_i$ .

$\tau_i$  : force généralisées appliquées à la  $i^{eme}$  articulation.

### 1.5.1.1 Energie cinétique de robot manipulateur

Le formalisme de L-E exige la connaissance de l'énergie cinétique du système physique, qui à son tour exige la connaissance de la vitesse de chaque liaison.

Soient  ${}^i r_i$  les coordonnées homogènes d'un point fixe appartenant à la liaison  $i$ , par rapport au référentiel de la liaison  $i$ ,  ${}^0 r_i$  ces coordonnées par rapport au référentiel fixe de la base.

On a donc :

$${}^0 r_i = A_i^0 \times {}^i r_i \quad (1.22)$$

avec :

$$A_i^0 = A_1^0 \times A_2^0 \times \dots \times A_i^0 \quad (1.23)$$

Puisqu'il s'agit d'une liaison rigide, il s'ensuit que :  $\frac{d{}^i r_i}{dt} = 0$

d'où

$${}^0 v_i = \frac{d{}^0 r_i}{dt} = {}^i n \left( \frac{d{}^0 r_i}{dt} \right) = {}^i r_i \left( \frac{d{}^0 A_i}{dt} \right) \quad (1.24)$$

d'une manière générale, on trouve :

$${}^0 v_i = \sum_{j=1}^i \frac{\partial A_i^0}{\partial q_j} \cdot \frac{dq_j}{dt} \cdot {}^i r_i \quad (1.25)$$

En remarquant que la dérivée précédente de la matrices  $A_i^0$  peut être obtenu aisément à l'aide de la matrice  $Q_i$  définie pour une structure à articulations rotationnelles, comme suit :

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.26)$$

et pour une structure à articulations prismatiques, comme suit :

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.27)$$

On obtient :

$$U_{ij} = \frac{\partial A_i^0}{\partial q_j} = \begin{cases} 0 & \text{si } j \geq i \\ A_{j-1}^0 Q_j A_i^{j-1} & \text{si } j \leq i \end{cases} \quad (1.28)$$

d'où en utilisant les notations précédentes, on aura :

$${}^0 v_i = \sum U_{ij} \frac{dq_j}{dt} \cdot {}^i r_i \quad (1.29)$$

Si on désigne par  $E_{ci}$  l'énergie cinétique de la liaison  $i$ , et par  $dE_{ci}$  l'énergie cinétique d'une particule de masse  $dm$  de l'articulation  $i$ , on a :

$$dE_{ci} = \frac{1}{2} \left( \left( \frac{dx_i}{dt} \right)^2 + \left( \frac{dy_i}{dt} \right)^2 + \left( \frac{dz_i}{dt} \right)^2 \right) dm \quad (1.30)$$

$$dE_{ci} = \frac{1}{2} \text{trace}({}^0v_i {}^0v_i^T) . dm \quad (1.31)$$

En substituons, l'équation (1.25) dans (1.31), on trouve :

$$dE_{ci} = \frac{1}{2} \text{trace} \left[ \sum_{p=1}^i \sum_{r=1}^i U_{ip} ({}^i n \ dm \ {}^i r_i^T) U_{ir}^T \dot{q}_p \dot{q}_r \right] \quad (1.32)$$

L'énergie cinétique totale de l'articulation  $i$  est donc donnée par :

$$E_{ci} = \frac{1}{2} \text{trace} \left[ \sum_{p=1}^i \sum_{r=1}^i U_{ip} \left( \int {}^i n \ {}^i r_i^T \right) U_{ir}^T \dot{q}_p \dot{q}_r \right] \quad (1.33)$$

Le terme entre parenthèse représente l'inertie de l'articulation et peut s'écrire sous forme matricielle comme suit :

$$\begin{bmatrix} \int x_i^2 . dm & \int x_i y_i . dm & \int x_i z_i . dm & \int x_i . dm \\ \int y_i x_i . dm & \int y_i^2 . dm & \int y_i z_i . dm & \int y_i . dm \\ \int x_i z_i . dm & \int z_i y_i . dm & \int z_i^2 . dm & \int z_i . dm \\ \int x_i . dm & \int y_i . dm & \int z_i . dm & \int dm \end{bmatrix} \quad (1.34)$$

Cette matrice d'inertie peut être exprimée à l'aide des moments d'inertie de la liaison  $i$  par rapport aux principaux axes du référentiel  $(x_i, y_i, z_i)$ .

$$\begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m \end{bmatrix} \quad (1.35)$$

Il s'ensuit que l'énergie totale du bras manipulateur sera la suivante :

$$E_{c\text{Totale}} = \frac{1}{2} \sum_{i=1}^n \left( \sum_{p=1}^i \sum_{r=1}^i \text{trace}(U_{ip} J_i U_{ir}^T) \dot{q}_p \dot{q}_r \right) \quad (1.36)$$

### 1.5.1.2 Energie potentielle du bras manipulateur

L'énergie potentielle de l'articulation  $i$   $E_{pi}$  est donnée par :

$$E_{ci} = - m_i . g . (A_i^0 \cdot i n) \quad (1.37)$$

$$E_c = - m_i . g . (A_i^0 \cdot i n) \quad (1.38)$$

tel que :  $g$  est le vecteur gravitation.  $g = (0 \ 0 \ -|g| \ 0)^T$

### 1.5.1.3 Equation dynamique du bras manipulateur

La fonction de lagrange sera donnée

$$L = \frac{1}{2} \sum_{i=1}^n \left( \sum_{p=1}^i \sum_{r=1}^i \text{trace}(U_{ip} J_i U_{ir}^T) \dot{q}_p \dot{q}_r \right) + \sum_{i=1}^n m_i g (A_i^0 r_i^i) \quad (1.39)$$

En appliquant l'équation (1.8) à la fonction lagrangienne donnée ci-dessus.

on trouve :

$$\tau_i = \sum_{j=i}^n \sum_{k=1}^i \text{tr}(U_{ik} J_j U_{jk}^T) \frac{d^2 q_k}{dt^2} + \sum_{j=i}^n \sum_{k=1}^i \sum_{m=1}^j \text{tr}(U_{jkm} J_j U_{ji}^T) \dot{q}_k \dot{q}_m - \sum_{j=i}^n m_i g U_{ji} r_j^j \quad (1.40)$$

$$U_{ijk} = \begin{cases} A_0^{k-1} Q_k A_{k-1}^{j-1} Q_j A_{j-1}^i & ; k \leq j \leq i \\ A_0^{j-1} Q_j A_{j-1}^{k-1} Q_k A_{k-1}^i & ; j \leq k \leq i \\ 0 & ; i < j \text{ ou } i < k \end{cases} \quad (1.41)$$

$\tau_i$  est le couple généralisé appliqué à la  $i^{\text{eme}}$  articulation.

L'écriture matricielle de l'équation dynamique de mouvement du bras manipulateur est la suivante :

$$\tau = D(q) \cdot \frac{d^2 q}{dt^2} + h(q, \frac{dq}{dt}) + c(q) \quad (1.42)$$

avec:

- $D_{ik} = \sum_{j=\max(i,k)}^n \text{tr}(U_{jk} J_j U_{ji}^T) \quad i, k = 1, n$
- $h_{ikm} = \sum_{j=\max(i,k,m)}^n \text{tr}(U_{jkm} J_j U_{ji}^T) \quad i, k, m = 1, n$
- $h_i = \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \cdot \dot{q}_k \cdot \dot{q}_m \quad i=1, n$
- $c_i = \sum_{j=1}^n -m_j \cdot g \cdot U_{ji} \cdot r_j^j \quad i=1, n$

#### Remarques

1. les coefficients  $c_i$ ,  $D_{ik}$ ,  $h_{ikm}$  sont en fonction des variables d'articulations  $\theta_i$  et des paramètres d'inertie du robot et sont appelés les coefficients dynamiques de robot manipulateur. à partir des équations données ci-dessus il est aisé de donner à chacun d'eux le sens physique correspondant.
2. les coefficients  $D_{ik}$  représentent l'accélération des variables d'articulation. pour  $i=k$ ,  $D_{ii}$  est relie à l'accélération de l'articulation  $i$  quand le couple  $T_i$  est appliqué. Pour  $i \neq k$ ,  $D_{ik}$  est relie au couple de réaction induit par l'articulation  $k$  et appliqué à l'articulation  $i$  ou vice versa.
3. les coefficients  $c_i$  représentent l'effet de la charge dû à la gravité de liaison  $i$ .
4.  $h_{ikm}$  est relié aux vitesses angulaires des articulations  $k$  et  $m$ , il s'agira de la force centrifuge générée par la vitesse angulaire  $\frac{d\theta_k}{dt}$  et exercée sur l'articulation  $i$ , autrement, il s'agira de la

force coriolis générée par les vitesses angulaires des articulations  $k$  et  $m$ . Enfin, le bon sens physique nous montre que :  $h_{ikm}=h_{imk}$ .

#### 1.5.1.4 Application au PUMA à 2 d d l

Pour établir le modèle dynamique du robot, nous avons besoin de calculer les matrices  $U_j$  et  $U_{ijk}$ , pour cela en utilise les équations (1.26),(1.27) et (1.41).

$$Q_1=Q_2=\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; U_{11}=A_0^1 Q_1 A_1^0=\begin{bmatrix} -s_1 & -c_1 & 0 & -ls_1 \\ -c_1 & -s_1 & 0 & lc_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.43)$$

$$U_{21}=Q_1 A_0^2=\begin{bmatrix} -s_{12} & -c_{12} & 0 & -l(s_{12}-s_1) \\ -c_{12} & -s_{12} & 0 & l(c_{12}+c_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; U_{22}=A_0^2 Q_2 A_0^2=\begin{bmatrix} -s_{12} & -c_{12} & 0 & -ls_{12} \\ -c_{12} & -s_{12} & 0 & lc_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.44)$$

$$U_{212}=U_{221}=Q_1 A_0^1 Q_2 A_1^2=\begin{bmatrix} -c_{12} & s_{12} & 0 & -lc_{12} \\ -s_{12} & -c_{12} & 0 & -ls_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; U_{211}=Q_1 A_0^2=\begin{bmatrix} -c_{12} & s_{12} & 0 & -l(c_{12}+c_1) \\ -s_{12} & -c_{12} & 0 & -l(s_{12}+s_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.45)$$

$$U_{222}=A_0^2 Q_2 A_1^2 Q_2 A_1^2=U_{212} \quad (1.46)$$

Les matrices d'inertie sont données par :

$$J_i=\begin{bmatrix} I_i & -1 & 0 & \bar{x}_i m_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \bar{x}_i m_i & 0 & 0 & m_i \end{bmatrix}; \quad i=1,2 \quad (1.47)$$

avec :  $I_i=\frac{1}{3}m_i l^2$  et  $\bar{x}_1=\bar{x}_2=-\frac{1}{2}$

On obtient finalement et après quelque calculs le modèle dynamique du robot manipulateur :

$$\tau(t)=D(\theta).\ddot{\theta}+h(\theta,\dot{\theta})+c(\theta) \quad (1.48)$$

$$\begin{bmatrix} \tau_1(\theta) \\ \tau_2(\theta) \end{bmatrix}=\begin{bmatrix} \frac{1}{3}m_1 l^2+\frac{4}{3}m_2 l^2+m_2 l^2 c_2 & \frac{1}{3}m_2 l^2+\frac{1}{2}m_2 l^2 c_2 \\ \frac{1}{3}m_2 l^2+\frac{1}{2}m_2 l^2 c_2 & \frac{1}{3}m_2 l^2 \end{bmatrix}\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \\ +\begin{bmatrix} -\frac{1}{2}m_2 l^2 s_2 \dot{\theta}_2^2-m_2 l^2 s_2 \dot{\theta}_1 \dot{\theta}_2 \\ \frac{1}{2}m_2 l^2 s_2 \dot{\theta}_1^2 \end{bmatrix}+\begin{bmatrix} \frac{1}{2}m_1 g l c_1+m_2 g l(\frac{1}{2}c_{12}+c_1) \\ \frac{1}{2}m_2 g l c_{12} \end{bmatrix} \quad (1.49)$$

### 1.5.2 Approche de NEWTON-EULER

Dans la partie précédente, on a élaboré les équations différentielles non linéaires du second ordre obtenues par le formalisme d'E-L. L'utilisation de ces équations pour déterminer les force généralisées à partir des positions, vitesses, accélérations généralisées pour une trajectoire donnée devient une tâche ardente et fastidieuse dès que le nombre de degrés de liberté est supérieur à deux, en plus elle nécessite un temps de calcul énorme. Dans le but d'améliorer la commande en temps réel du robot, un modèle dynamique simplifié a été proposé. Ce modèle néglige les force coriolis et centrifuges, et réduit remarquablement le temps de calcul de la commande, Néanmoins, L'effet coriolis et centrifuge est très important quant le robot est amené à suivre des trajectoires rapides. Cela dit que le modèle dynamique simplifié restreint le mouvement de robot aux plages des faibles vitesses qui ne sont pas désirables dans un environnement industriel.

La méthode de N-E est actuellement considérée comme l'approche la plus favorable et la plus exploitable. En effet, sa structure récursive la rend facilement implémentable sur des calculateurs numériques. Le formalisme de N-E peut être considéré comme l'inverse du modèle dynamique obtenu par l'approche d'E-L et donc utilisé dans la commande inverse des robots manipulateurs. La méthode de N-E utilise deux étapes pour la dérivation des forces généralisées à partir des positions vitesses et accélérations désirées. Ces deux étapes vont être déterminées dans les sections qui suivent.

#### Rappel théorique

Soit le repère fixe  $(O, X, Y, Z)$  et le repère mobile  $(O, X^*, Y^*, Z^*)$  représenté sur la figure 1.4 tournant autour du repère fixe à la vitesse angulaire  $\omega$ .

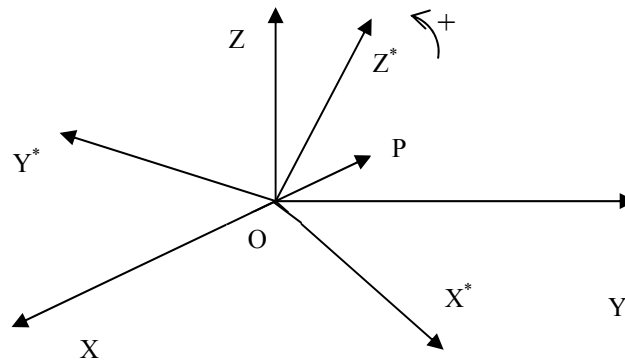


figure 1.4 Rotation d'un repère par rapport à un autre

La vitesse et l'accélération du point mobile P repère fixe  $(O, X, Y, Z)$  sont données par :

$$\frac{dr}{dt} = \frac{dr^*}{dt} + \omega \times r \quad (1.50)$$

$$\frac{d^2r}{dt^2} = \frac{d^2r^*}{dt^2} + 2\omega \times \frac{dr^*}{dt} + \omega \times (\omega \times r) + \frac{d\omega}{dt} \times r \quad (1.51)$$

tel que  $\frac{dr^*}{dt}$  et  $\frac{d^2r^*}{dt^2}$  sont respectivement les vitesses, accélérations relatives du point P par rapport au repère mobile  $(O, X^*, Y^*, Z^*)$ . Le terme  $2\omega \times \frac{dr^*}{dt}$  représente l'accélération complémentaire ou de Coriolis, et le terme  $\omega \times (\omega \times r)$  représente l'accélération centrifuge.

dans le cas ou le repère  $(O, X^*, Y^*, Z^*)$  possède un mouvement de translation par rapport au repère fixe, figure 1.5

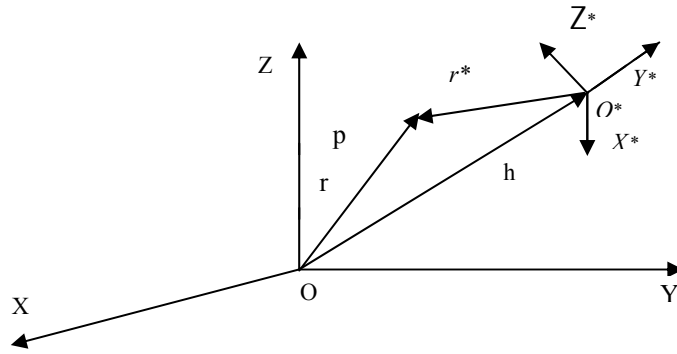


figure 1.5 Mouvement de translation et de rotation d'un repère  
la vitesse et l'accélération absolu du point P s'exprime par :

$$v(t) = \frac{dr^*}{dt} + \frac{dh}{dt} = \frac{dr^*}{dt} + \omega \times r^* + \frac{dh}{dt} \quad (1.52)$$

$$a(t) = \frac{d^2r^*}{dt^2} + 2\omega \times \frac{dr^*}{dt} + \omega \times (\omega \times r^*) + \frac{d\omega}{dt} \times r^* + \frac{d^2h}{dt^2} \quad (1.53)$$

Les équations (1.52) à (1.53) vont être utilisées pour déterminer les vitesses et accélérations linéaires et angulaires de chaque liaison par rapport à la liaison adjacente.

### 1.5.2.1 Application aux repères liés aux liaisons d'un robot

Soit  $v_i$  et  $\omega_i$  les vitesses linéaires et angulaires du repère  $(x_i, y_i, z_i)$  par rapport à la base  $(x_0, y_0, z_0)$ ,  $\omega_i^*$  est la vitesse angulaire du point o par rapport à la base, figure 1.1

on a :  $\omega_i = \omega_{i-1} + \omega_i^*$

avec :

$$\omega_i^* = \begin{cases} z_{i-1} \dot{q}_i & si \ \mathfrak{R} \\ 0 & si \ \mathfrak{T} \end{cases} \quad (1.54)$$

où  $\mathfrak{R}$  désigne une liaison rotationnelle et  $\mathfrak{T}$  une liaison translationnelle.

on obtient donc :

$$\omega_i = \begin{cases} \omega_{i-1} + z_{i-1} \dot{q}_i & si \ \mathfrak{R} \\ \omega_{i-1} & si \ \mathfrak{T} \end{cases} \quad (1.55)$$

$$\omega_i = \begin{cases} \omega_{i-1} + z_{i-1} \dot{q}_i + \omega_{i-1} \times (z_{i-1} \dot{q}_i) & si \ \mathfrak{R} \\ \omega_{i-1} & si \ \mathfrak{T} \end{cases} \quad (1.56)$$

les vitesses et accélérations linéaires de la liaison i sont obtenues en utilisons les équations (1.51) à (1.55) et peuvent être exprimées par :

$$v_i = \begin{cases} \omega_i \times P_i^* + v_{i-1} & si \ \mathfrak{R} \\ z_{i-1} \dot{q}_i + \omega_i \times (\omega_i \times P_i^*) + v_{i-1} & si \ \mathfrak{T} \end{cases} \quad (1.57)$$

$$a_i = \begin{cases} \dot{\omega}_i \times P_i^* + \omega_i \times (\omega_i \times P_i^*) + \dot{v}_{i-1} & si \ \mathfrak{R} \\ z_{i-1} \ddot{q}_i + \dot{\omega}_i \times P_i^* + 2\omega_i \times (z_{i-1} \dot{q}_i) + \omega_i \times (\omega_i \times P_i^*) + v_{i-1} & si \ \mathfrak{T} \end{cases} \quad (1.58)$$



### 1.5.2.2 Équations récursives de mouvement du robot

Après qu'on a obtenu les informations cinématiques de chaque liaison  $(v_i, a_i, \omega_i, \dot{\omega}_i)$ , on va maintenant déterminer les équations de mouvement du robot en utilisant le principe de d'Alembert. Le principe mis en évidence les conditions d'équilibre statique, et s'annonce comme suit : Pour chaque système mécanique, la somme algébrique des forces extérieures et de forces de résistances dans n'importe quelle directions et appliquées sur un point est nulle.

Considérant une chaîne de liaisons articulées, figure 1.6

On définit les paramètres suivants :

$O$  : centre de masse de liaison  $i$ .

$\bar{s}_i$  : position de centre du masse par rapport au repère  $R_i(x_i, y_i, z_i)$ .

$\bar{v}_i, \bar{a}_i$  vitesses et accélérations linéaires absolues du centre de masse par rapport à la base.

$F_i$  : la force extérieure totale exercée sur la liaison  $i$  en son centre de masse.

$N_i$  : le moment extérieure totale exercée sur la liaison  $i$  en son centre de masse.

$I_i$  : matrice d'inertie de la liaison  $i$  calculée autour de son centre de masse et par rapport à la base.

$f_i$  : force appliquée sur la liaison  $i$  par la liaison  $i-1$  par rapport au repère  $R_{i-1}$ .

$n_i$  : moment appliqué sur la liaison  $i$  par la liaison  $i-1$  par rapport  $R_{i-1}$ .

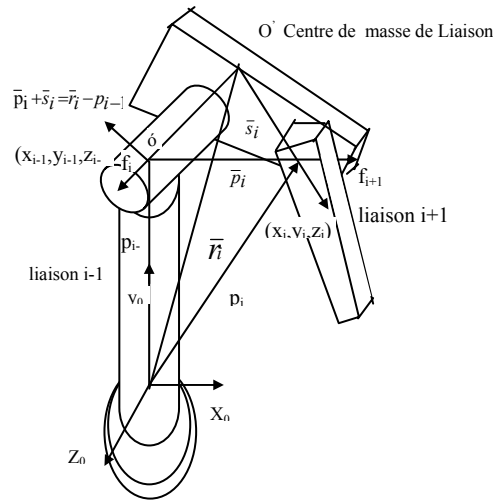


figure 1.6 force et moments pour diverses liaisons

on a :

$$F_i = \frac{d(m_i \bar{v}_i)}{dt} = m_i \bar{a}_i \quad (1.59)$$

$$N_i = \frac{d(I_i \omega_i)}{dt} = I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i) \quad (1.60)$$

La vitesse et l'accélération absolues du centre de masse sont données par :

$$\bar{v}_i = v_i + \omega_i \times \bar{s}_i \quad (1.61)$$

$$\bar{a}_i = \dot{v}_i + \dot{\omega}_i \times \bar{s}_i + \omega_i \times (\omega_i \times \bar{s}_i) \quad (1.62)$$

Les équations relient les forces et moments exercées sur la liaison  $i$  à celles exercées sur la liaison  $i+1$  sont données par :

$$f_i = m_i \bar{a}_i + f_{i+1} \quad (1.63)$$

$$n_i = n_{i+1} + P_i^* \times f_{i+1} + (P_i^* + \bar{s}_i) \times F_i + N_i \quad (1.64)$$

Ces équations récursives peuvent être utilisées pour obtenir les forces et moments pour chaque liaisons  $i$   $f_{n+1}$ ,  $n_{n+1}$  sont respectivement les forces et moments appliquées par l'élément terminal sur un objet extérieur du système.

Dans la modélisation cinématique, on a vu que si la liaison  $i$  est rotationnelle, elle va tourner d'un angle  $q_i$  par rapport au  $R_{i-1}$  suivant l'axe  $z_{i-1}$ . Donc le couple appliqué sur l'articulation  $i$  est la somme de projection de  $n_i$  sur l'axe  $z_{i-1}$  et de moment visqueux. Dans le cas d'une articulation prismatique (translationnelle), cette dernière va se déplacer de  $q_i$  le long de l'axe  $z_{i-1}$ . La force appliquée sur l'articulation  $i$  est donc la somme de projection  $f_i$  sur l'axe  $z_{i-1}$  et une force due au déplacement visqueux.

On obtient donc :

$$\tau_i = \begin{cases} n_i^T z_{i-1} + b_i \dot{q}_i & si \quad \mathfrak{R} \\ f_i^T z_{i-1} + b_i \dot{q}_i & si \quad \mathfrak{T} \end{cases} \quad (1.65)$$

Les équations de N-E se compose de deux ensemble d'équations récursive : les équations directe « forward équations » données par (1.56), (1.57), (1.59) et (1.63), et les équations de retour « backward équations » données par (1.59), (1.60), (1.61), (1.64), (1.65).

Les équations directes propagent les informations cinétiques de la base jusqu'à l'élément terminal. Pour les équations de retour, les couples et force de chaque liaison sont calculés récursivement de l'élément terminal jusqu'à la base.

### 1.5.2.3 Equations récursives modifiées

L'inconvénient majeur des équations récursives précédentes est le fait que les matrices d'inertie  $I_i$  et les paramètres géométriques  $(\bar{s}_i, P_i^*)$  sont établies par rapport à la base, donc il changent quand le robot est en mouvement. Il existe une version améliorée de la méthode de N-E qui consiste à référencer toutes les vitesses, accélérations, matrices d'inertie centres de masse, forces et moments par rapport à ces repères. Par cette nouvelle méthode le temps de calcul diminue remarquablement et devient proportionnel au nombre de d d l.

soit la matrice de rotation du repère  $R_i$  vers le repère  $R_{i-1}$

$${}^{i-1}R_i = \begin{bmatrix} c\theta & -c\alpha s\theta & -s\alpha s\theta \\ s\theta & c\alpha c\theta & -s\alpha c\theta \\ 0 & s\alpha & c\alpha \end{bmatrix} \quad (1.66)$$

les équations de N-E par rapport au repère lié à chaque liaison deviennent :

**Forward equations:**  $i=1, 2, \dots, n$

$${}^i R_0 \omega_i = \begin{cases} {}^i R_{i-1} ({}^{i-1} R_0 \omega_{i-1} + z_{i-1} \dot{q}_i) & si \quad \mathfrak{R} \\ {}^i R_{i-1} ({}^{i-1} R_0 \omega_{i-1}) & si \quad \mathfrak{T} \end{cases} \quad (1.67)$$

$${}^i R_0 \dot{\omega}_i = \begin{cases} {}^i R_{i-1} [{}^{i-1} R_0 \dot{\omega}_{i-1} + z_0 \ddot{q}_i + ({}^{i-1} R_0 \omega_{i-1}) \times z_0 \dot{q}_i] & si \quad \mathfrak{R} \\ {}^i R_{i-1} ({}^{i-1} R_0 \dot{\omega}_{i-1}) & si \quad \mathfrak{T} \end{cases} \quad (1.68)$$

$${}^i R_0 a_i = \begin{cases} ({}^i R_0 \dot{\omega}_i) \times ({}^i R_0 P_i^*) + ({}^i R_0 \omega_i) \times [({}^i R_0 \dot{\omega}_i) \times ({}^i R_0 P_i^*)] + {}^i R_{i-1} ({}^{i-1} R_0 a_{i-1}) \\ {}^i R_{i-1} [z_0 \ddot{q}_i + {}^{i-1} R_0 a_{i-1}] + ({}^i R_0 \dot{\omega}_i) \times ({}^i R_0 P_i^*) + 2 ({}^i R_0 \omega_i) \times ({}^i R_{i-1} z_0 \dot{q}_i) + ({}^i R_i \omega_i) \times [({}^i R_0 \dot{\omega}_i) \times ({}^i R_0 P_i^*)] \end{cases} \quad (1.69)$$

$${}^i R_0 \bar{a}_i = ({}^i R_0 \dot{\omega}_i) \times ({}^i R_0 \bar{s}_i) + ({}^i R_0 \omega_i) \times [({}^i R_0 \omega_i) \times I_i {}^i R_0 \bar{s}_i] + {}^i R_0 a_i \quad (1.70)$$

**Backward equations:**  $i=n-1, \dots, 1$

$${}^i R_0 F_i = m_i {}^i R_0 \bar{a}_i \quad (1.71)$$

$${}^i R_0 N_i = ({}^i R_0 I_i^0 R_i) ({}^i R_0 \dot{\omega}_i) + ({}^i R_0 \omega_i) \times [({}^i R_0 I_i^0 R_i) ({}^i R_0 \omega_i)] \quad (1.72)$$

$${}^i R_0 f_i = {}^i R_0 F_i + {}^i R_{i+1} ({}^{i+1} R_0 f_{i+1}) + ({}^i R_0 P_i^*) \times ({}^{i+1} R_0 f_{i+1}) + ({}^i R_0 P_i^* + {}^i R_0 \bar{s}_i) \times ({}^i R_0 F_i) + {}^i R_0 N_i \quad (1.73)$$

$$\tau_i = \begin{cases} ({}^i R_0 n_i)^T ({}^i R_{i-1} z_0) + b_i \dot{q}_i & \text{si } \mathfrak{R} \\ ({}^i R_0 f_i)^T ({}^i R_{i-1} z_0) + b_i \dot{q}_i & \text{si } \mathfrak{S} \end{cases} \quad (1.74)$$

### 1.5.2.4 Algorithme de NEWTON-EULER

Conditions initiales : n nombre de degré de liberté.

$$\omega_0 = \dot{\omega}_0 = v_0 = 0 \quad (1.75)$$

$$a_0 = [g_x \quad g_y \quad g_z] \quad (1.76)$$

Etape 1 : attribuer au compteur la valeur  $i=1$

Etape 2 : calcule de  ${}^i R_0 \omega_i$ ,  ${}^i R_0 \dot{\omega}_i$ ,  ${}^i R_0 a_i$ ,  ${}^i R_0 \bar{a}_i$

Etape 3 : test si  $i=n$  go to Etape 4 si non  $i=i+1$  go to Etape 2

Etape 4 : calcule de  $f_{n+1}$ ,  $n_{n+1}$  en cas de charge, si il y'a pas de charge, ils sont nulles

Etape 5 : calcule de  ${}^i R_0 F_i$ ,  ${}^i R_0 N_i$ ,  ${}^i R_0 f_i$ ,  ${}^i R_0 n_i$  et  $\tau_i$

Etape 6 : si  $i=1$  go to stop si non  $i=i-1$  go to Etape 6

## 1.6 Résultat du simulation

L'évolution de robot en position, vitesse et accélération, est simulée en utilisant le modèle de connaissance établi par le formalisme de L-E. La mise sous forme d'état des équations différentielles non linéaires trouvées nous permettent d'utiliser la méthode de RANGE-KUTTA (R.K) d'ordre quatre sous les conditions initial nulle.

La mise sous forme matricielle du modèle de robot donne le système suivant [3]:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (1.77)$$

$$\begin{cases} \alpha = \frac{1}{3} l^2 (m_1 + 4m_2 + 3m_2 \cos(q_2)) \\ \beta = m_2 l^2 \left( \frac{1}{3} + \frac{1}{2} \cos(q_2) \right) \\ \gamma = \frac{1}{3} m_2 l^2 \end{cases} \quad (1.78)$$

$$\left\{ \begin{array}{l} f_1 = -m_2 l^2 \dot{q}_2 \sin(q_2) \left( \frac{1}{2} \dot{q}_2 + \dot{q}_1 \right) \\ f_2 = \frac{1}{2} m_2 \sin(q_2) l^2 \dot{q}_1^2 \\ g_1 = \frac{1}{2} m_1 g l \cos(q_1) + m_2 g l \left[ \frac{1}{2} \cos(q_1 + q_2) + \cos(q_1) \right] \\ g_2 = \frac{1}{2} m_2 g l \cos(q_1 + q_2) \end{array} \right. \quad (1.79)$$

tel que :  $b_1$  et  $b_2$  sont les termes de frottement visqueux. On détermine alors la forme d'état suivante :

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{\left[ \frac{(k_1 u_1 - f_1 - g_1 - b_1 \dot{q}_1)}{\alpha} + (-k_2 u_2 + f_2 + g_2 + b_2 \dot{q}_2) \frac{\beta}{\alpha \gamma} \right]}{\left[ 1 - \frac{\beta}{\alpha \gamma} \right]} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{\left[ \frac{(k_2 u_2 - f_2 - g_2 - b_2 \dot{q}_2)}{\gamma} + (-k_1 u_1 + f_1 + g_1 + b_1 \dot{q}_1) \frac{\beta}{\alpha \gamma} \right]}{\left[ 1 - \frac{\beta}{\alpha \gamma} \right]} \end{array} \right. \quad (1.80)$$

Les valeurs numériques des différents paramètres de robot sont [3] :

$$\left\{ \begin{array}{l} l_1 = l_2 = l = 1 \text{ m} \\ m_1 = m_2 = 10 \text{ kg} \\ b_1 = 75 \text{ N/rd/s} \\ b_2 = 10 \text{ N/rd/s} \\ k_1 = 40 \text{ Nm/v} \\ k_2 = 20 \text{ Nm/v} \\ g = 9.81 \text{ ms}^{-2} \end{array} \right. \quad (1.81)$$

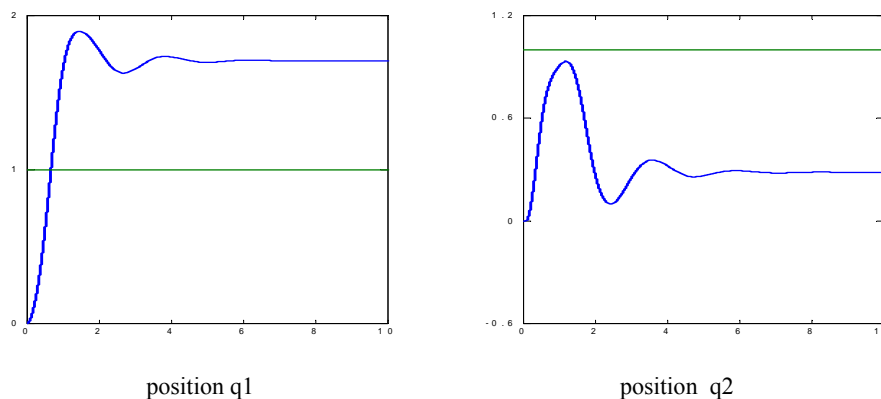
$b_1, b_2$  : frottement visqueux des actionneurs.

$k_1, k_2$  : constantes de couple moteur.

$l_1, l_2$  : longueurs des deux liaisons.

$m_1, m_2$  : masses des deux liaisons

Les réponses de robot en position, vitesse et accélération pour une entrée échelon, sur un horizon de temps suffisant, sont consignées sur la figure 1.7



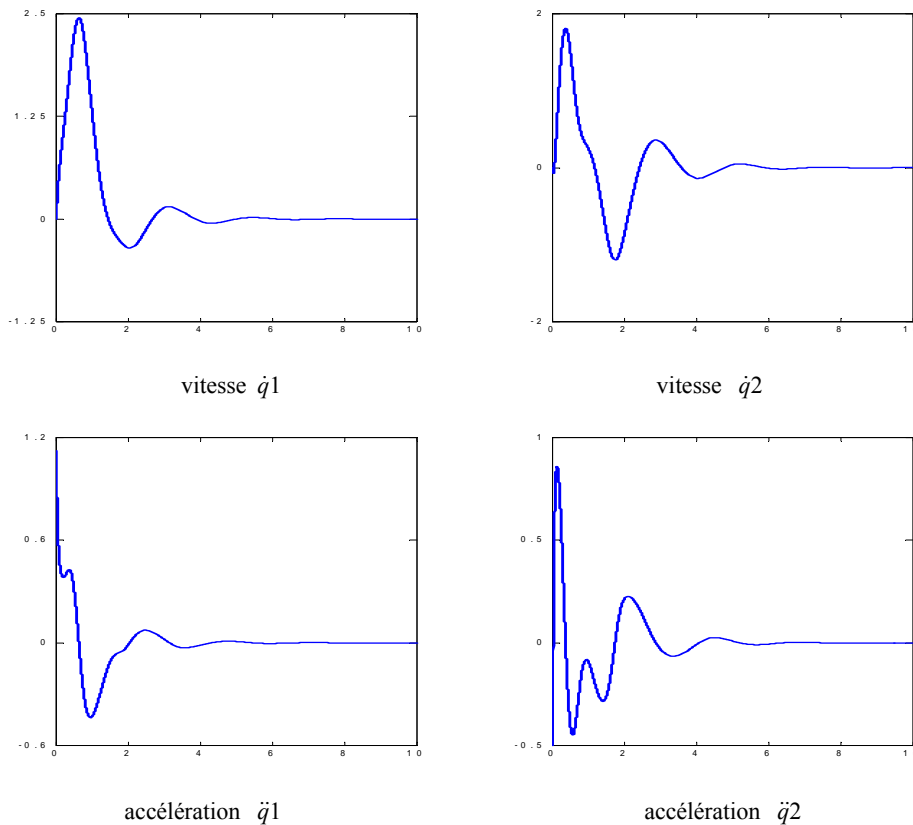


figure 1.7 réponse en boucle ouverte de robot PUMA en position, vitesse, accélération

## 1.6 Validation des modèles dynamiques

Toute phase de modélisation est soumise à la validation. L'obtention d'un modèle dynamique est assujettie à un critique. Pour conformer un modèle trouvé, on est toujours amené à choisir un autre concept dans lequel on modélise le système en utilisant les lois différentes de celles utilisées pour le modèle initial.

Dans notre cas, on s'est intéressé à modéliser le robot en utilisant deux formalismes, celui de principe du moindre action (L-E) et celui du principe de d'ALAMBERT (N-E).

La récursivité de l'algorithme de N-E lui permet une implémentation directe sur ordinateur. L'utilisation de cet algorithme, pour le calcul manuel des équations différentielles du modèle, est une étape très délicate.

Donc le but de cette partie est de confirmer que les deux modèles, obtenus par les deux formalismes, sont identiques.

D'après la figure présentée, les deux modèles se valident si les erreurs définies par :  $\varepsilon_{1,2} = q_{d1,2} - q_{1,2}$  ;  $\dot{\varepsilon}_{1,2} = \dot{q}_{d1,2} - \dot{q}_{1,2}$  ;  $\ddot{\varepsilon}_{1,2} = \ddot{q}_{d1,2} - \ddot{q}_{1,2}$  soient très faibles.

Les figures 1.9 montrent la variation des positions, vitesses et d'accélération pour chaque trajectoire désirée et celles générées par le modèle dynamique d'E-L. on présente alors à côté de chaque caractéristique cinématique leurs erreurs.

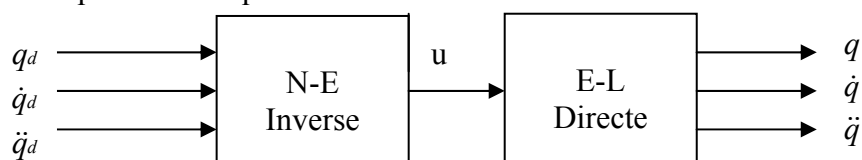
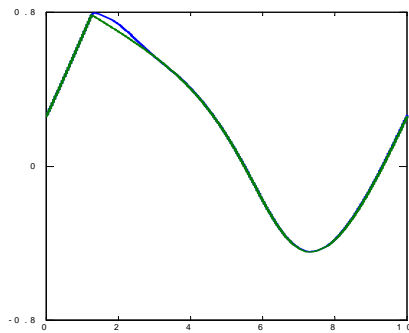
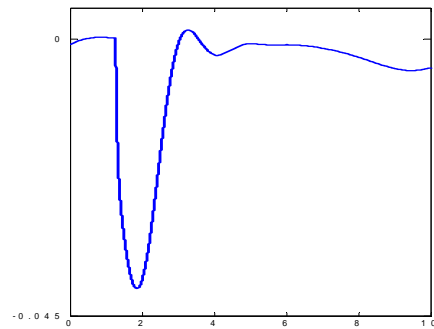
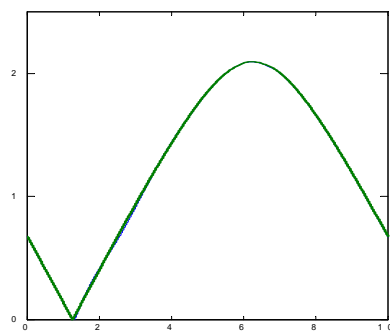
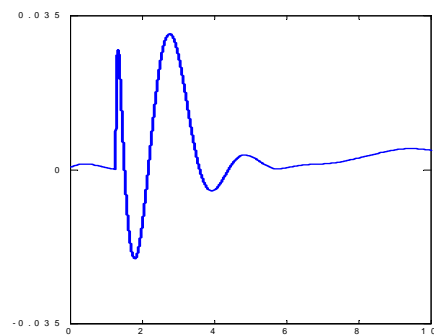
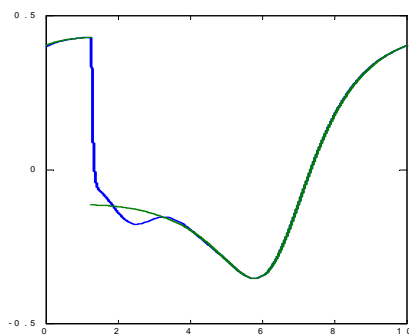
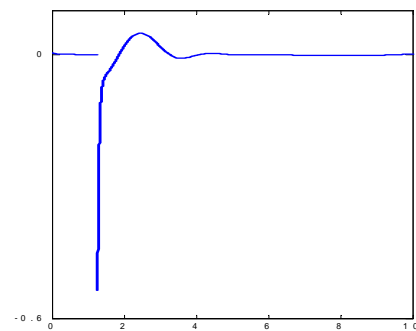
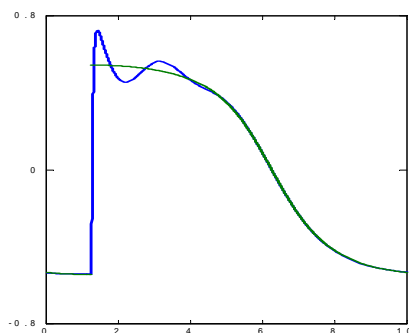
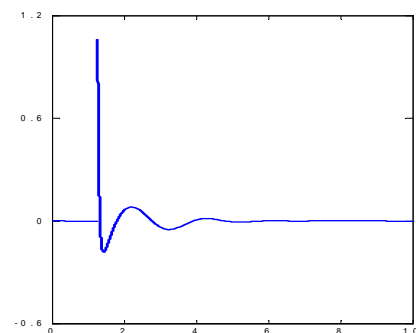


figure 1.8 couplage des deux modèles NE-EL

**Remarque**

Toutes les simulations ont été réalisées avec un pas de calcul de  $dt=10^{-3}$

validation de position  $q_1$ erreur de validation de position  $e_1$ validation de position  $q_2$ erreur de validation de position  $e_2$ validation de vitesse  $\dot{q}_1$ erreur de validation de vitesse  $\dot{e}_1$ validation de vitesse  $\dot{q}_2$ erreur de validation de vitesse  $\dot{e}_2$

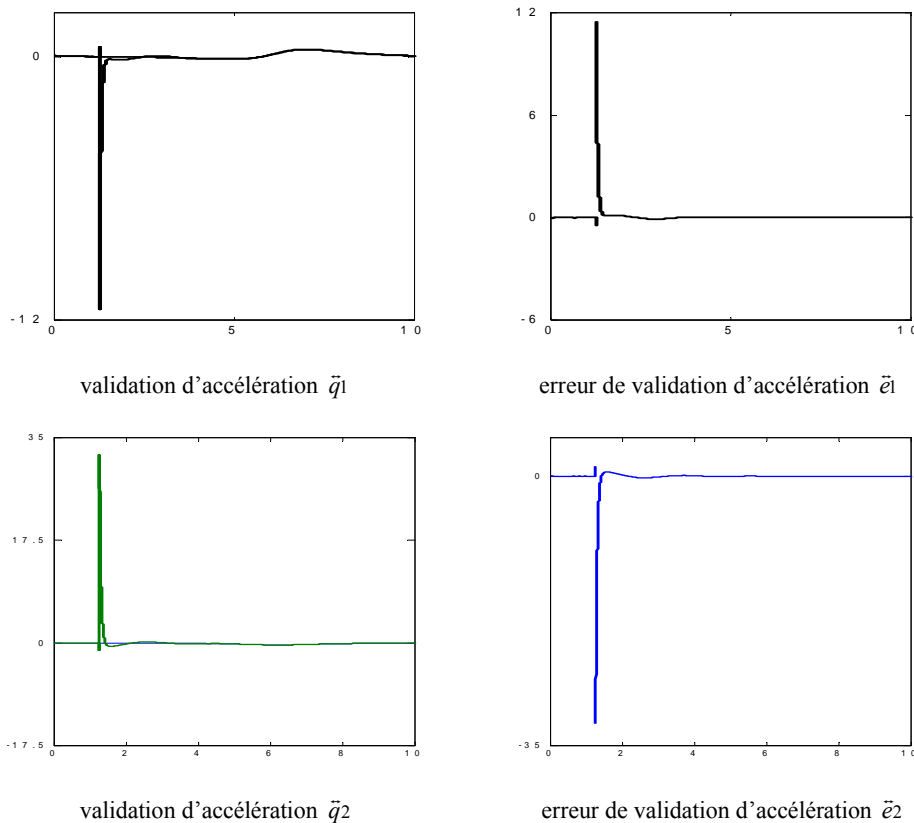


figure 1.9 validation en positions, vitesses et accélérations de robot PUMA

## 1.8 Linéarisation

L'idée principale de la linéarisation est de transformer les dynamiques d'un système non linéaire en dynamique linéaire, de sorte que les techniques de contrôle linéaires peuvent être appliquées.

### 1.8.1 Linéarisation entrée/sortie

Nous allons maintenant présenter une méthodologie afin d'étudier une commande pour ce type de système. La modélisation de robot manipulateur PUMA nous a montrée qu'il est un système non-linéarité et fortement couplé. Les commandes pour chaque sortie ne sont pas indépendante, mais nous oblige à envisager une commande découplante pour ce système. Pour étudier une commande classique de type linéarisation entrées/sorties, plusieurs solutions s'offrent à nous. La première est de faire une linéarisation approchée autour d'un point d'équilibre du système c'est-à-dire un découplage linéaire d'un modèle linéarisé du système [4]. Cette approche est intéressante et se prête bien aux problèmes de régulation. La deuxième, qui est nommée linéarisation exacte entrées/sorties [5] permet elle de grandes transitions du point d'équilibre, mais en revanche cette méthode peut faire apparaître des problèmes de singularités. Il est important de noter que quelle que soit la méthode choisie, l'état du système devra être mesuré. Nous avons choisi d'appliquer dans cette étude la méthode de linéarisation exacte qui nous permettra d'avoir une commande indépendante du point de vue fonctionnement.

### 1.8.2 Linéarisation exacte entrées/sorties [5]

La linéarisation exacte est une approche pour la construction de correcteurs non linéaires. L'idée principale de cette méthode est d'effectuer une transformation algébrique d'un système dynamique non linéaire, en un système totalement ou partiellement linéaire. Ainsi après transformation, toutes les techniques de synthèse de correcteurs linéaires peuvent être appliquées. L'intérêt de cette méthode est qu'elle effectue une transformation exacte du système d'état non linéaire, sans passer par une approximation linéaire.

La linéarisation exacte d'un système peut donc être vue comme une méthode pour transformer un système vers un système ayant une forme plus simple. Cette méthode a été utilisée avec succès dans de nombreuses applications [5].

Soit  $\Sigma$  un système non linéaire affine multi-entrées, multi-sorties (MIMO), qui peut être représenté par les équations d'état suivante :

$$(\Sigma) \begin{cases} \dot{x} = f(x) + \sum_{k=1}^m g^k(x) u_k \\ y_j = h_j(x) \end{cases} \quad \text{avec } 1 \leq j \leq p \quad (1.82)$$

où  $X \in \Omega$  est le vecteur d'état avec  $\Omega \in R^n$  et  $m$  le nombre d'entrées. Les fonctions  $f, g_1, \dots, g_m$  sont des applications vectorielles de  $\Omega$  dans  $R^n$  et les fonctions  $h_j$  sont des fonctions de valeurs réelles définies sur  $\Omega$ . Nous nous intéresserons plus particulièrement aux systèmes carrés donc on posera  $p=m$ .

une représentation vectorielle peut être donnée par :

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix}, \quad g^k(x) = \begin{pmatrix} g_{1k}(x) \\ \vdots \\ g_{nk}(x) \end{pmatrix} \quad \text{et} \quad h(x) = \begin{pmatrix} h_1(x) \\ \vdots \\ h_m(x) \end{pmatrix} \quad (1.83)$$

Les éléments de  $f, g_k$  et  $h$  sont des fonctions lisses, c'est-à-dire que ces fonctions sont indéfiniment dérivables par rapport à chacun de leurs arguments.

Etant donné une fonction lisse  $h_j(x)$ , le gradient de  $h_j$  est noté  $\nabla h_j$  et s'exprime par :

$$\nabla h_j = \frac{\partial h_j}{\partial X} \quad (1.84)$$

Ce gradient est représenté par un vecteur colonne d'éléments  $\nabla h_j(x_k) = \frac{\partial h_j}{\partial x_k}$ . de la même façon,

Etant donné un champ de vecteurs  $f(X)$ , le jacobien de  $f$  est noté par  $\nabla f$  et s'exprime par :

$$\nabla f = \frac{\partial f}{\partial X} \quad (1.85)$$

Le jacobien est une matrice  $n \times n$  qui a pour éléments  $(\nabla f)_{kj} = \frac{\partial f_k}{\partial x_j}$

On pose maintenant une fonction scalaire  $h_j(x)$  lisse et un champ de vecteurs lisse  $f(x)$

On définit une nouvelle fonction scalaire,  $L_f h_j$  appelée dérivée de lie de  $h_j$  par rapport à  $f$  [5] comme suit :

$$L_f h_j(x) = \sum_{k=1}^m \frac{\partial h_j(x)}{\partial x_k} f_k(x) = \frac{\partial h_j(x)}{\partial x_k} f(x) \quad (1.86)$$

En reprenant le modèle du système donné par l'équation (1.82), on peut écrire la première dérivée de  $y_j$  à l'aide des dérivées de lie.



$$\dot{y}_j = \frac{\partial h_j(x)}{\partial X} \dot{X} = \frac{\partial h_j(x)}{\partial X} (f(x) + \sum_{k=1}^m g_k(x) u_k) \quad (1.87)$$

on obtient alors

$$\dot{y}_j = L_f h_j(x) + \sum_{k=1}^m L_{g_k} h_j(x) u_k \quad (1.88)$$

Si  $L_{g_k} h_j(x) = 0$  les entrées  $u_k$  n'ont pas d'influence sur les sorties. On note ainsi  $r_j$  le plus petit nombre entier pour qu'une des entrées  $u_k$  apparaisse dans la  $r_j^{ieme}$  dérivée de la sortie  $y_j$ .

On a alors

$$y_j^{(r_j)} = L_f^{(r_j)} h_j(x) + \sum_{k=1}^m L_{g_k} (L_f^{(r_j-1)} h_j(x)) u_k \quad (1.89)$$

Les nombres obtenus  $r_j$  sont appelés les degrés relatifs.

On définit la matrice de découplage  $m \times m$ ,  $A(X)$  comme suit

$$\Delta(x) = \begin{pmatrix} L_{g_1} L_f^{(r_1-1)} h_1(x) & \dots & L_{g_m} L_f^{(r_1-1)} h_1(x) \\ \vdots & & \vdots \\ L_{g_1} L_f^{(r_m-1)} h_m(x) & \dots & L_{g_m} L_f^{(r_m-1)} h_m(x) \end{pmatrix} \quad (1.90)$$

On définit également le vecteur  $\Delta_0(x)$  par :

$$\Delta_0(x) = \begin{pmatrix} L_f^{(r_1)} h_1(x) \\ \vdots \\ L_f^{(r_m)} h_m(x) \end{pmatrix} \quad (1.91)$$

Il est alors possible de réécrire l'équation (1.89) en utilisant les équations (1.90) et (1.91)

$$y = \begin{pmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{pmatrix} = \Delta_0(x) + \Delta(x) \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} \quad (1.92)$$

Le système d'équation  $(\Sigma)$  est découplé sur  $\Omega \subset R^n$  si et seulement si :  $\text{rang } \Delta(x) = m, \forall x \in \Omega$

Si cette condition est satisfaite alors le retour d'état non-linéaire

$$u(x) = \alpha(x) + \beta(x)v \quad (1.93)$$

$$\begin{cases} \alpha(x) = -\Delta^{-1}(x)\Delta_0(x) \\ \beta(x) = \Delta^{-1}(x) \end{cases} \quad (1.94)$$

tel que  $v$  est le nouveau vecteur d'entrée découple le système  $(\Sigma)$  sur  $\Omega$ .

Le système bouclé  $(h, f + g\alpha, g\beta)$  possède un comportement entrées/sorties linéaire décrit par :

$$y_j^{(r_j)} = v_j \quad \forall j \in [1, \dots, m] \quad (1.95)$$

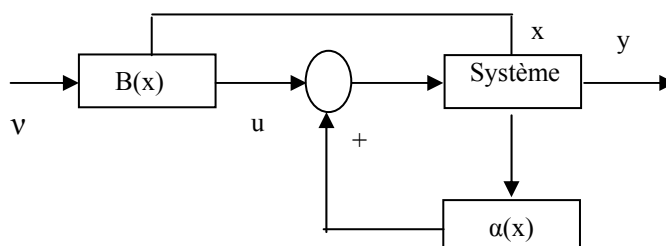


figure 1.10 représentation de linéarisation d'un système non linéaire.

On effectue sur le système ainsi linéarisé figure 1.10 par ce retour d'état non linéaire un second bouclage linéaire afin d'imposer au système la dynamique désirée. Toutes les méthodes de synthèse de correcteurs linéaires pourront ainsi être utilisées. Il est important de noter que cette méthode aboutit à un découplage total si on a  $\sum r_j = n$ , c'est-à-dire que le système découplé est linéaire pour toutes les variables d'état.

On obtient après découplage le schéma de la figure 1.11

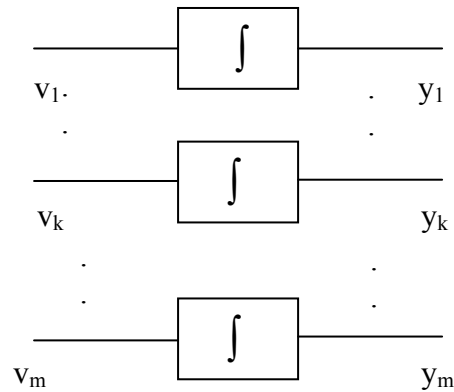


figure 1.11 Système découplé

### 1.8.3 Application au robot manipulateur PUMA

Nous allons maintenant appliquer cette méthode à un robot manipulateur PUMA à deux degrés de liberté. Le vecteur d'état du système est donc d'ordre 4, Le modèle utilisé pour le robot sera donc un modèle non linéaire affine donné par l'équation (1.80).

$u_1$  et  $u_2$  représente les entrées de système et  $x_1, x_2, x_3, x_4$  sont les états de système.

Le calcul de degrés relatifs donne  $r_1=r_2=2$ , donc la linéarisation est exacte et se trouve complètement découplé, ( $r_1 + r_2 = 4 = n$ ) et chaque sortie peut être représenté par un intégrateur,

Le retour d'état peut donc s'exprime par

$$\alpha(x) = -\Delta(x)^{-1}\Delta_0(x) \quad (1.96)$$

$$\text{et } \beta(x) = \Delta^{-1}(x) \quad (1.97)$$

la linéarisation exacte de robot manipulateur PUMA est donné par le système linéaire suivant :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = v_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = v_2 \end{cases} \quad (1.98)$$

## 1.9 Conclusion

Les formalismes de N-E et L-E permettent d'établir un système d'équations différentielles reliant les coordonnées généralisées aux forces généralisées, obtenant ainsi le modèle dynamique du robot. Le modèle cinématique inverse est un modèle spécifiant la relation entre les coordonnées cartésiennes dans l'espace opérationnel et les coordonnées généralisés.

Le modèle dynamique obtenu à partir du formalisme de L-E couplé avec celui de N-E aboutit à une commande en boucle ouverte. Ce couplage permet une validation des modèles dynamiques du robot

La linéarisation exacte entrée/sortie est basée sur la recherche d'une transformation non linéaire qui transforme le modèle de robot qui est non linéaire en un modèle linéaire à l'aide d'une commande par retour état.

# *CHAPITRE II*

**Etude des performances de la commande CDM**

## 2.1 Introduction

Le principe de la commande robuste repose sur la structure de commande à contre-réaction et sur les modèles linéaires temps invariant (LTI) des systèmes dynamiques. Le principe de la commande à contre-réaction est au cœur des théories classiques et modernes de la commande et définit une structure de commande très générale fondée sur la comparaison entre l'information supportée par un ensemble de signaux de sorties mesurés par des capteurs et l'information supportée par un ensemble de signaux de référence souhaités. Les différences constatées lors de cette comparaison sont traitées par le système de commande afin d'élaborer un signal de commande affectant le système.

La sensibilité des algorithmes de commande classique (PID) aux variations paramétriques du procédé à contrôler nécessite la recherche des performances plus élevées qui conduit à explorer de nouvelles techniques de commande permettant de dépasser les limites d'autres techniques. La nouvelle méthode de commande utilisée est la méthode de diagramme des coefficients qui est basée sur un outil d'analyse des performances appelée diagramme des coefficients [6].

## 2.2 Développement de la commande : méthode de diagramme des coefficients

La méthode de diagramme des coefficients est classée dans les méthodes d'approche algébriques appliquées pour les boucles polynomiales, Elle est classée entre les méthodes classique et la théorie de contrôle moderne. Un diagramme spécial nommé diagramme des coefficients, est un outil de choix des meilleures performances. La simplicité de la structure du contrôleur (RST) rend la CDM très puissante pour la commande des systèmes.

Les performances de la CDM sont spécifiées dans la fonction de transfert en boucle fermée et ont une relation avec les paramètres algébriques du contrôleur. La CDM garde l'équilibre entre l'exactitude des performances et la complexité du contrôleur [7]. Le principe fondamental de la CDM est illustré par l'équation caractéristique en boucle fermée, le diagramme des coefficients.

## 2.3 Polynôme caractéristique du système en boucle fermée

La forme générale de polynôme caractéristique d'un système linéaire en boucle fermée donné par la figure 2.1 est comme suit :

$$P(s) = a_n s^n + \dots + a_1 s + a_0 = \sum_{i=0}^n a_i s^i \quad (2.1)$$

tel que :  $a_i$  sont les coefficients du polynôme caractéristique

les indices de stabilité  $\gamma_i$  le constant de temps équivalent  $\tau$  et les limites de stabilité  $\gamma_i^*$  sont exprimées en fonction de l'équation (2.1), Ceci est nécessaire pour le développement de la commande CDM.

$$\gamma_i = \frac{a_i^2}{a_{i+1} a_{i-1}} \quad i=1, \dots, n-1, \quad \gamma_0 = \gamma_n = \infty \quad (2.2)$$

$$\tau = \frac{a_1}{a_0} = \frac{t_s}{\lambda} \quad (2.3)$$

tel que :  $t_s$  est le temps de réponse désiré pour le système global et  $\lambda$  une constante comprise dans l'intervalle [ 2.5 3 ]

$$\gamma_i^* = \frac{1}{\gamma_{i-1}} + \frac{1}{\gamma_{i+1}} \quad (2.4)$$

La relation entre ces paramètres est défini par l'équation (2.5) et (2.6)

$$\frac{a_{i+1}}{a_i} = \frac{a_j}{a_{j-1}} \frac{1}{\gamma_i \gamma_{i-1} \gamma_{j+1} \gamma_j} \quad i > j \quad (2.5)$$

$$a_i = \frac{a_0 \tau^i}{\gamma_{i-1} \gamma_{i-2}^2 \dots \gamma_1^{i-1}} \quad (2.6)$$

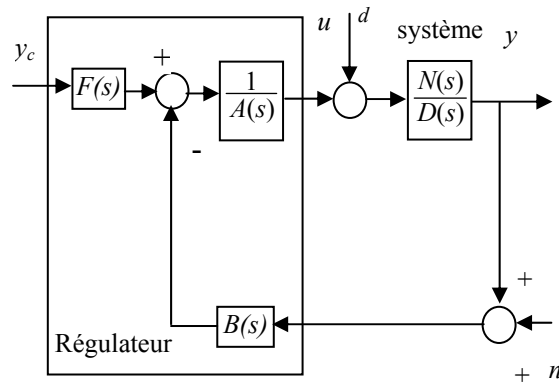


figure 2.1 schéma block de commande d'un système par CDM

Le polynôme caractéristique donné par l'équation (2.1) doit être exprimée en fonction des paramètres  $a_0, \tau$  et  $\gamma_i$  comme suit :

$$P(s) = a_0 \left[ \left\{ \sum_{i=2}^n \left( \prod_{j=1}^{i-1} \frac{1}{\gamma_{i-j}^j} \right) (\tau s)^i \right\} + \tau s + 1 \right] \quad (2.7)$$

Le constant de temps équivalent de l'ordre  $i$  dépend de l'ordre zéro comme suit :

$$\tau_i = \frac{a_{i+1}}{a_i} = \frac{\tau}{\gamma_i \dots \gamma_2 \gamma_1} \quad (2.8)$$

Les indices de stabilité de quatrième ordre sont donnés par rapport de première ordre comme suit :

$$\gamma_{ij} = \frac{a_i^2}{a_{i+j} a_{i-j}} = \prod_{k=1}^{j-1} (\gamma_{i+j-k} \gamma_{i-j+k})^k \gamma_i^j \quad (2.9)$$

Les indices de stabilité de deuxième ordre donnent de bonne performance [3].

Les polynômes  $A_c(s)$ ,  $B_c(s)$  et  $F_c(s)$  de correcteur CDM, sont choisis comme suit :

$$A_c(s) = \sum_{i=1}^{m_1} l_i s^i, \quad B_c(s) = \sum_{i=1}^{m_2} k_i s^i \quad \text{et} \quad F_c(s) = \frac{P(s)}{N(s)} \Big|_{s=0} \quad (2.10)$$

En absence de perturbations le degré des deux polynômes  $A_c(s)$  et  $B_c(s)$  sont égale à  $\text{deg}[D(s)] - 1 = n - 1$ , c'est à dire  $m_1 = m_2 = n - 1$ , par contre, en présence de perturbations ou du bruit de mesure, les degrés  $m_1$  et  $m_2$  sont choisis de telle sorte à minimiser l'effet de la perturbation et du bruit [8].

## 2.4 Diagramme des coefficients

Les performances de la commande CDM, indice de stabilité  $\gamma_i$ , la constante de temps équivalent  $\tau$  et la limite de stabilité  $\gamma_i^*$  sont indiquées dans la partie droite de diagramme des coefficients tandis que les paramètres de polynôme caractéristique  $a_i$  sont représentés sur la partie gauche de diagramme, la constante  $\tau$  d'ordre zéro est représentée par une ligne liant la valeur 1 par  $\tau$  [9].

$$y(s) = \frac{2}{1+3s+4s^2} \quad (2.11)$$

La figure 2.2 représente le diagramme des coefficients de système de l'équation (2.11) en boucle fermée ayant le polynôme caractéristique suivant :

$$P(s) = 0.5 s^5 + 1.8 s^4 + 2.5 s^2 + 1.5 s + 0.4 \quad (2.12)$$

Les paramètres de CDM sont :

$$\gamma_i = [2 \quad 2 \quad 2 \quad 2.5] \quad (2.13)$$

$$\tau = \frac{a_1}{a_0} = 3.75 \quad (2.14)$$

$$\gamma_i = [0.5 \quad 1 \quad 0.9 \quad 0.5] \quad (2.15)$$

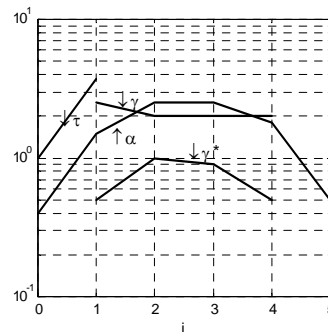


figure 2.2 le diagramme des coefficients de système donné par l'équation (2.11)

Le degré de convexité de la courbe tracée à partir des coefficients du polynôme caractéristique renseigne sur la robustesse de correcteur [10]. L'inclinaison de la partie de la courbe des  $a_i$  nous donne la constante de temps équivalente, ce qui fait que cette pente donne des renseignements sur la rapidité de système en boucle fermée.

## 2.5 Les conditions de stabilité et de robustesse

La stabilité et la robustesse de commande CDM est étudiées par un série de recherche [11][12] de critère de ROUTH-HURWITZ, la condition de stabilité de système d'ordre 3 est comme suit :

$$a_2 a_1 > a_0 a_3 \rightarrow \gamma_1 \gamma_2 > 1 \quad (2.16)$$

La condition de stabilité pour un système d'ordre quatre est donné par :

$$a_2 > \frac{a_1 a_4}{a_3} + \frac{a_0 a_3}{a_1} \rightarrow \gamma_2 > \gamma_2^* \quad (2.17)$$

La condition de stabilité et de l'instabilité de système d'ordre 5 et plus est donnée comme suit :

$$\gamma_i > 1.12 \gamma_i^* \quad i=2, \dots, (n-2) \quad (2.18)$$

$$\gamma_i \gamma_{i+1} \leq 1 \quad i=2, \dots, (n-2) \quad (2.19)$$

La robustesse indique la réalisation de stabilité de système lorsque les pôles sont variables sur l'axe imaginaire. Ainsi la condition de robustesse est donné par  $\gamma_i > 1.5 \gamma_i^*$  au lieu de  $\gamma_i = 2.5$ ,  $\gamma_2 = \gamma_3 = 2$  où  $\gamma_4 \sim \gamma_{n-1} = 2$ .

## 2.6 La forme standard de CDM

La forme standard de CDM (forme de MANABE) [13] recommandée est de choisir les indices de stabilités suivantes :

$$\gamma_i = [2 \quad 2 \quad 2 \quad 2.5], \quad i=1, \dots, (n-1), \quad \gamma_0 = \gamma_n = \infty \quad (2.20)$$

La forme standard de CDM donne la bonne performance, un dépassement propre, temps de réponse désiré.

## 2.7 Algorithme de commande CDM

On considère le système en boucle fermée donnée par la figure 2.1, tel que y est la sortie de système, u est la commande,  $y_c$  est la consigne, d est la perturbation aléatoire et n est le bruit, le numérateur et le dénominateur sont N(s) et B(s), A(s), F(s) et B(s) sont les polynômes de contrôleur.

La sortie de système à contrôlée est :

$$y = \frac{N(s)F(s)}{P(s)} r + \frac{A(s)N(s)}{P(s)} d - \frac{N(s)B(s)}{P(s)} n \quad (2.21)$$

où P(s) est le polynôme caractéristique et donné comme suit :

$$P(s) = D(s)A(s) + N(s)B(s) = \sum_{i=0}^n a_i s^i \quad (2.22)$$

Les performances sont reliées à 3 entrées r, d et n où sont donnés par l'équation (2.21).

Les étapes a réalisée pour choisir les paramètres de contrôleur CDM sont :

- les polynômes  $A(s)$ ,  $B(s)$  et  $F(s)$  sont choisis convenablement pour l'élimination de perturbation.
- le constant de temps équivalent  $\tau$ .
- les indices de stabilité  $\gamma_i$  pour un système d'ordre supérieur et les limites de stabilité  $\gamma_i^*$ .
- le diagramme des coefficients de système en boucle fermé en faisant un petit ajustement pour satisfaire les performances exigées.

## 2.8 Résultats de simulation

L'objectif de cette partie est d'étudier l'influence des paramètres de synthèse de la commande CDM sur plusieurs systèmes, à savoir, temps de réponse  $t_s$ , le paramètre  $\lambda$ , le coefficient  $\alpha_0$ , l'indice de stabilité  $\gamma$ , en effet le choix de ces paramètres influent considérablement sur les performances du système, l'algorithme de la CDM a été appliqué à quatre type de système :



- Système stable
- Système stable à phase non minimale
- Système instable
- Système instable à phase non minimale

Le but est de tester l'efficacité de la commande CDM et d'obtenir des résultats suffisamment généraux concernant le choix des paramètres  $t_s, \lambda, \gamma$ . La référence imposée aux systèmes est un signal carré de commande. Pour chaque type de système et pour chaque paramètre de commande choisi, on présente les résultats de simulation concernant l'évolution de la sortie et de la commande en absence et en présence de perturbations.

### 2.8.1 Etude d'un système stable

Soit la fonction de transfert de système stable suivante :

$$y(s) = \frac{3}{2 + 2s + s^2} \cdot u(s) \quad (2.23)$$

Dont les pôles et les zéros sont stable et ayant pour valeur  $s_1 = -1+i$ ,  $s_2 = -1-i$ . La simulation de ce système en boucle ouverte pour une entrée échelon, présentée sur la figure 2.3 montre que le régime statique est établi à 6s. Le système répond très rapidement en suivant exactement la référence imposée avec des pics de commande suffisante.

L'évolution de système simulé pour plusieurs valeurs de temps de réponse respectivement pour  $t_s=0.01$ ,  $t_s=0.1$ ,  $t_s=0.5$ ,  $t_s=1$  est représentée sur les figures 2.4 et 2.5, en gardant  $\lambda=2.5$ ,  $\gamma=2.5$ , cette augmentation joue sur la dynamique de sortie, on constate la diminution de l'effort de commande.

L'effet de paramètre  $\lambda$  sur la dynamique de système est représenté sur les figures 2.6 et 2.7, on remarque l'influence de variation de  $\lambda$ , les valeurs de  $\lambda$  comprise dans l'intervalle  $[2.5 \ 3]$  donne de bonne performance, l'augmentation de  $\lambda$  donne une commande puissante avec diminution de temps de réponse, tandis que la diminution de ce paramètre donne la diminution de l'effort de commande avec augmentation de temps de réponse.

L'influence de variation de  $\gamma$  sur l'effort de commande et la dynamique de sortie est illustré sur les figures 2.8 et 2.9, on constate que les performances sont affecté pour  $\gamma=2.5$ , tandis que l'augmentation de l'indice de stabilité ( $\gamma > 2.5$ ) rend les commandes puissante et leur diminution,  $\gamma < \gamma^*$  le temps de réponse croie et certain cas perdre la stabilité de système, ce qui montre que l'intervalle  $[\gamma^* \ \gamma]$  donne la stabilité et la bonne performance.

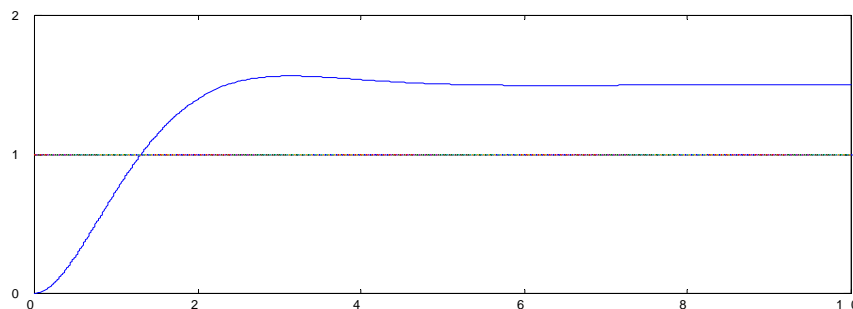
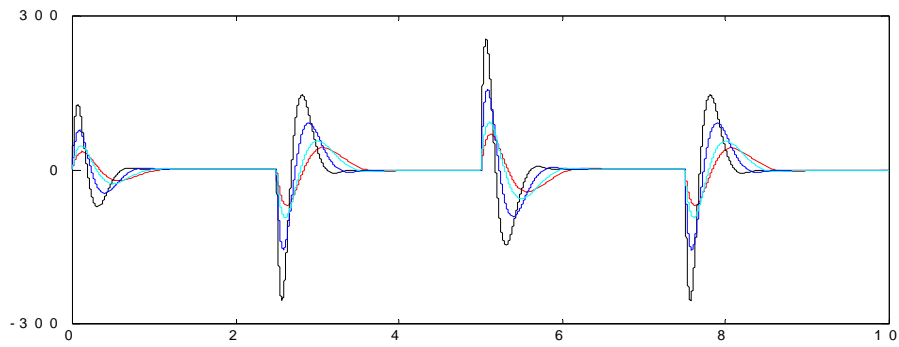
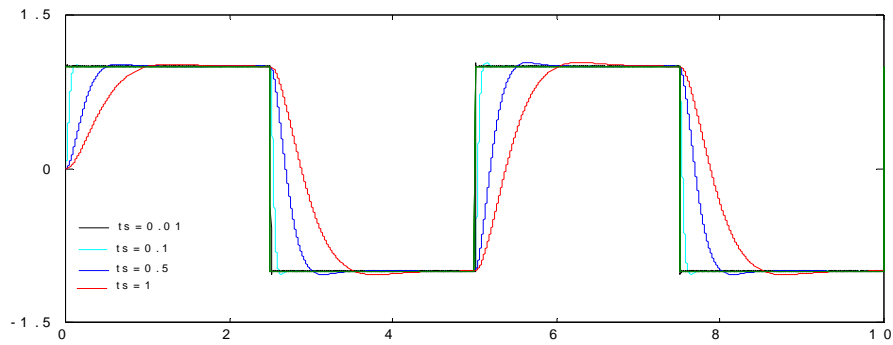


figure 2.3 réponse d'un système stable en boucle ouverte à une entrée échelon.

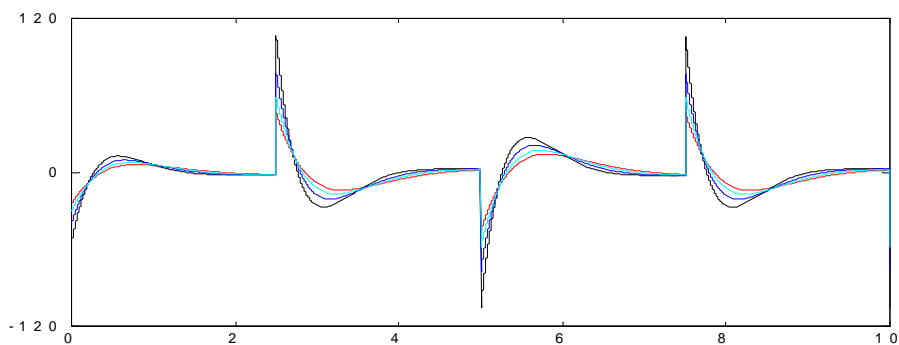


évolution des commandes

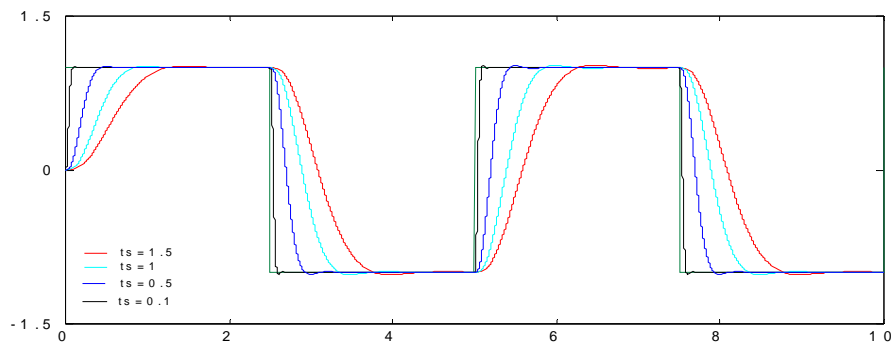


réponses de système

figure 2.4 présentation sortie/commande de système stable pour  $\lambda=2.5, \gamma=2.5$  et variation de  $t_s$  en absence de perturbation

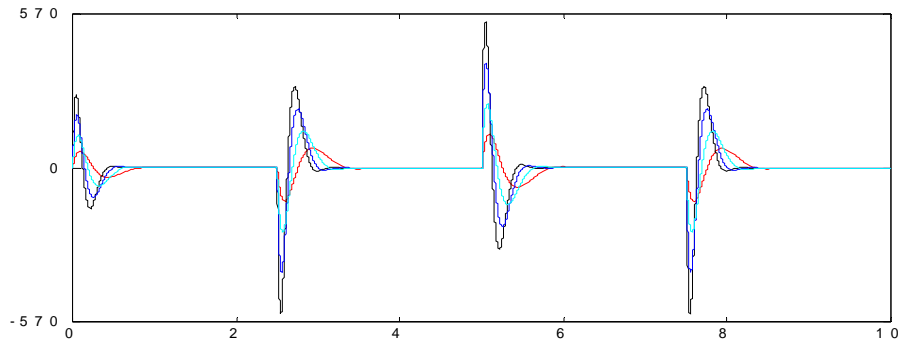


évolution des commandes

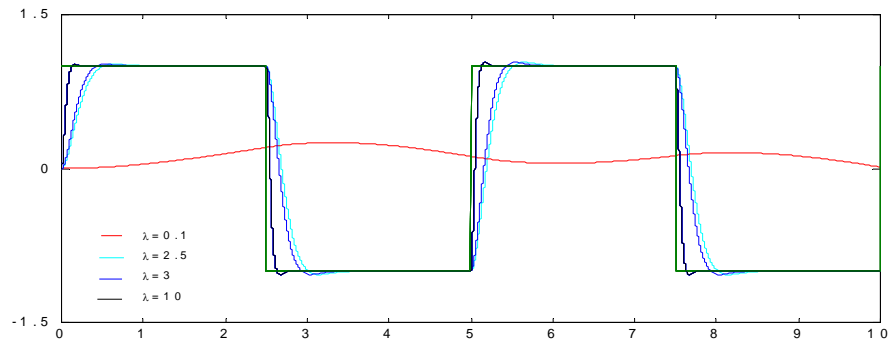


réponses de système

figure 2.5 présentation sortie/commande de système stable pour  $\lambda=2.5, \gamma=2.5$  et variation de  $t_s$  en présence de perturbation

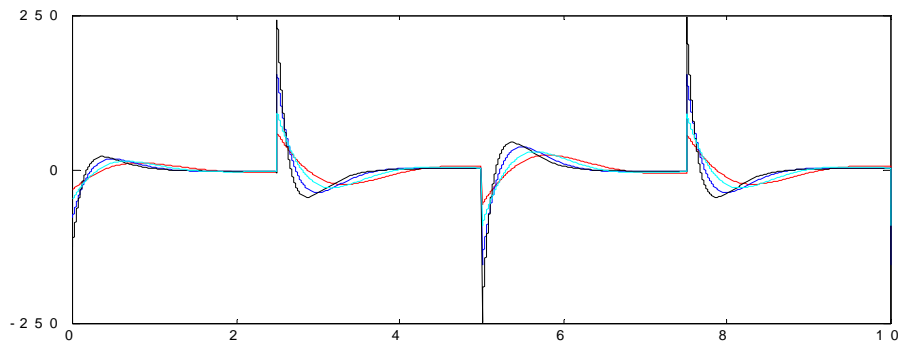


évolution des commandes

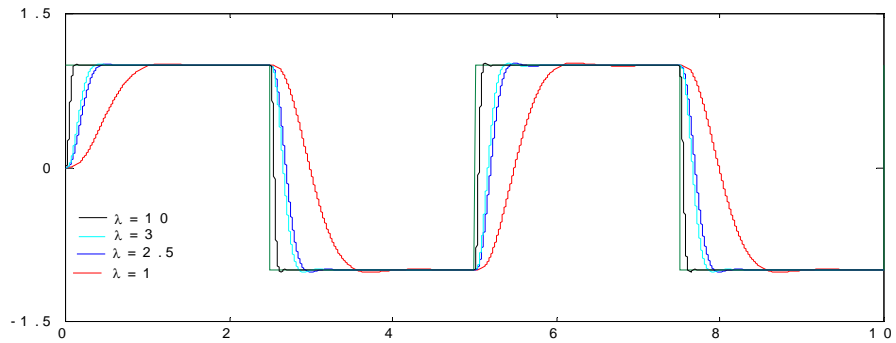


réponses de système

figure 2.6 présentation sortie/commande de système stable pour  $t_s=0.5, \gamma=2.5$  et variation de  $\lambda$  en absence de perturbation

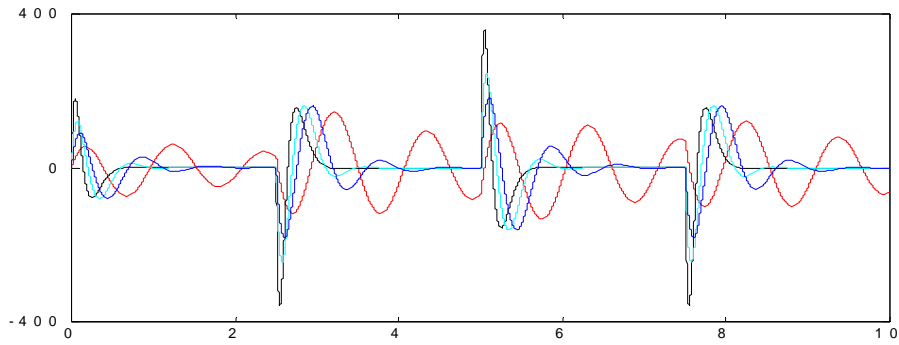


évolution des commandes

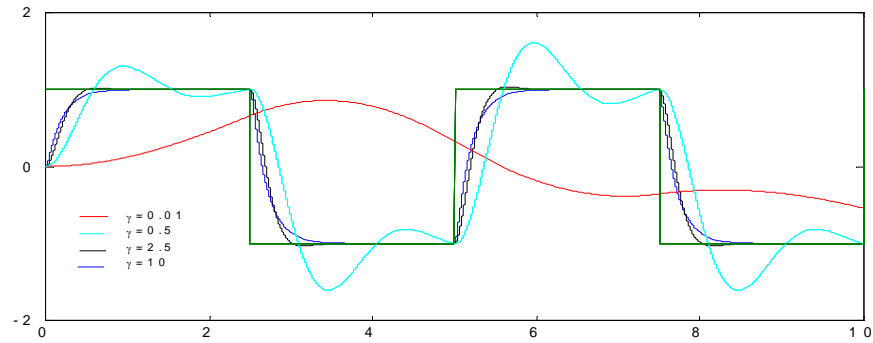


réponses de système

figure 2.7 présentation sortie/commande de système stable pour  $t_s=0.5, \gamma=2.5$  et variation de  $\lambda$  en présence de perturbation.

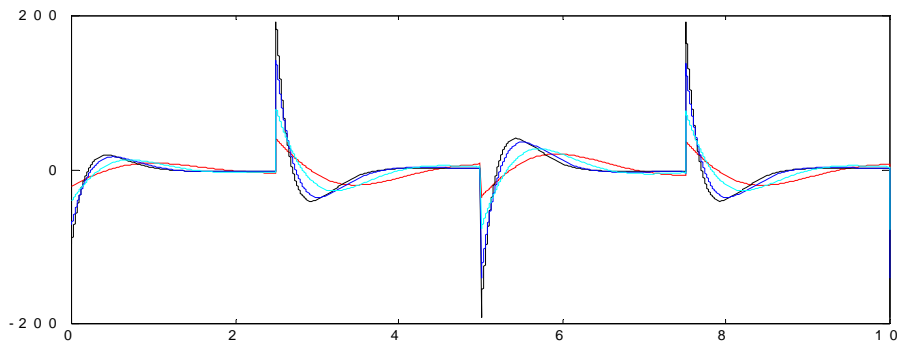


évolution des commandes

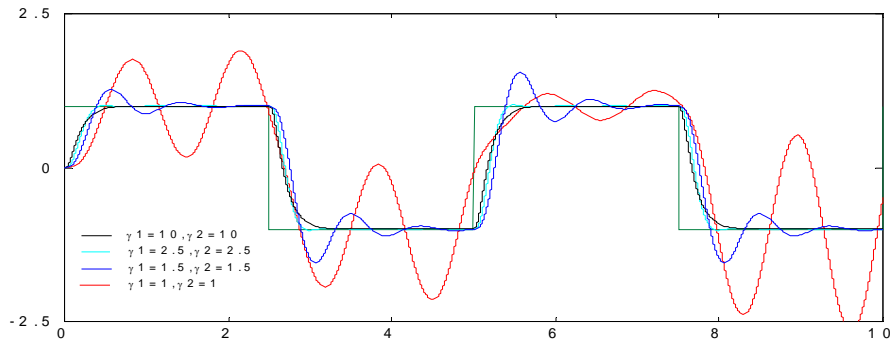


réponses de système

figure 2.8 présentation sortie/commande de système stable pour  $t_s=2.5$ ,  $\lambda=2.5$  et variation de  $\gamma$  en absence de perturbation



évolution des commandes



réponses de système

figure 2.9 présentation sortie/commande de système stable pour  $t_s=2.5$ ,  $\lambda=2.5$  et variation de  $\gamma$  en présence de perturbation.

## 2.8.2 Etude d'un système stable à phase non minimale

Soit le système suivant :

$$y(s) = \frac{-1+s}{4+4s+4s^2} \cdot u(p) \quad (2.24)$$

Présentant un zéro  $s=1$  (instable) et de pôles  $s_1 = -0.5 + 0.86i$ ,  $s_2 = -0.5 - 0.86i$  stables. La simulation de système en boucle ouverte pour une entrée échelon est présentée sur la figure 2.10 et montre que le régime statique est établi pour 5 s.

L'influence de variation de temps de réponse  $t_s$  est représentée sur les figures 2.11 et 2.12, on constate que l'augmentation de temps de réponse rend la commande puissante et la diminution de ce paramètre donne la diminution de l'effort de commande.

L'effet de paramètre  $\lambda$  est illustré sur les figures 2.13 et 2.14, l'augmentation de ce paramètre ( $\lambda > 3$ ) rend la commande puissante avec diminution de temps de réponse, cependant la réduction de ce paramètre ( $\lambda < 2.5$ ) donne l'atténuation de l'effort de commande et augmentation de  $t_s$ .

La variation de l'indice de stabilité  $\gamma$  est représentée sur les figures 2.15 et 2.16, lorsque  $\gamma < \gamma^*$ , la simulation montre l'augmentation de temps de réponse et certaine valeur rend le système instable, néanmoins l'agrandissement de  $\gamma$  produit une commande puissante.

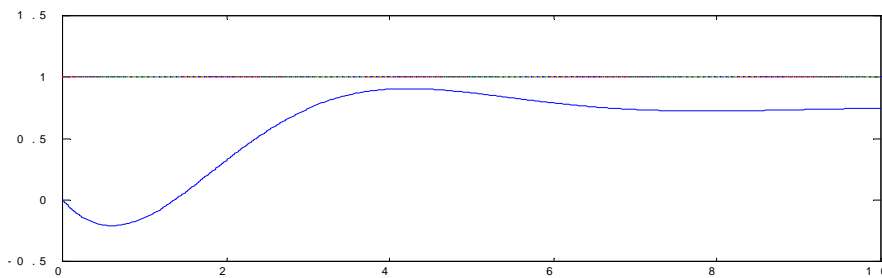
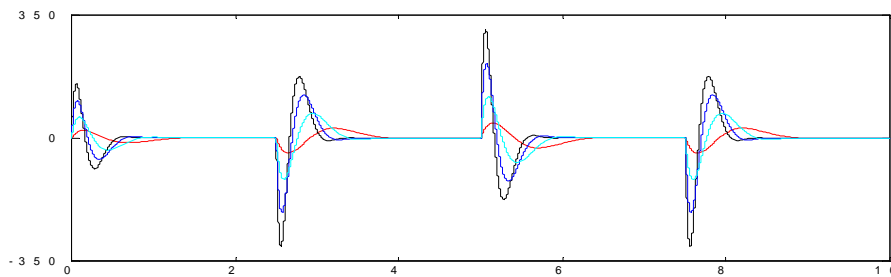
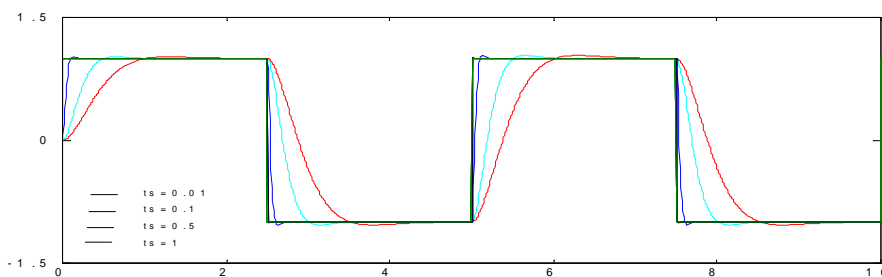


figure 2.10 réponse d'un système stable à P.N.M en boucle ouverte à une entrée échelon

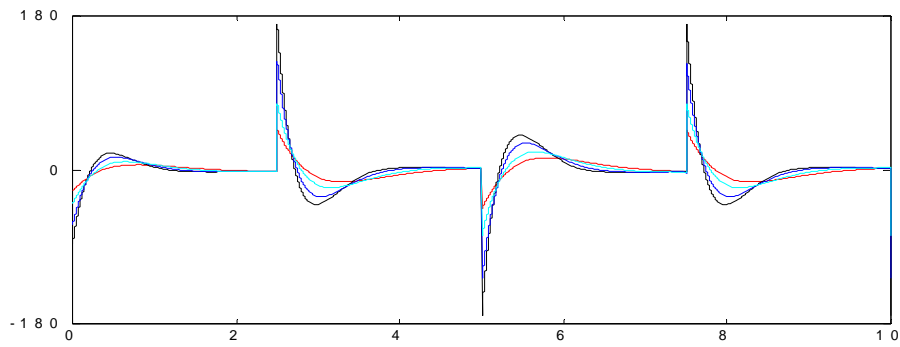


évolution des commandes

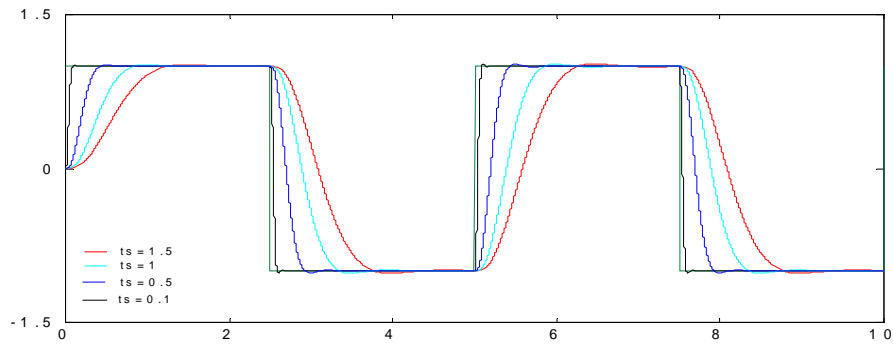


réponses de système

figure 2.11 présentation sortie/commande de système stable à PNM pour  $\lambda=2.5$ ,  $\gamma=2.5$  et variation de  $t_s$  en absence de perturbation

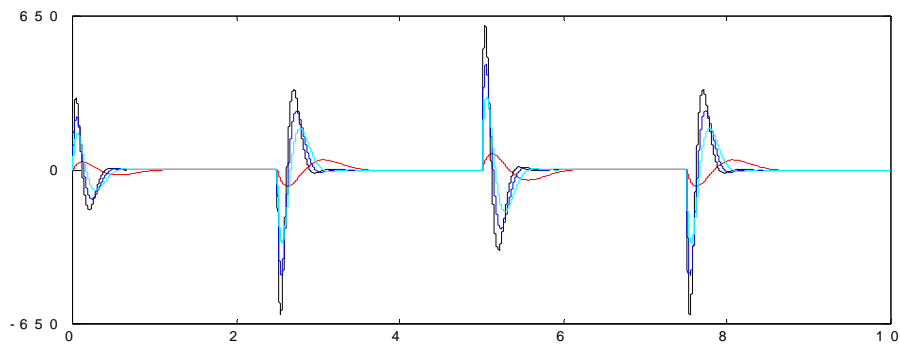


évolution des commandes

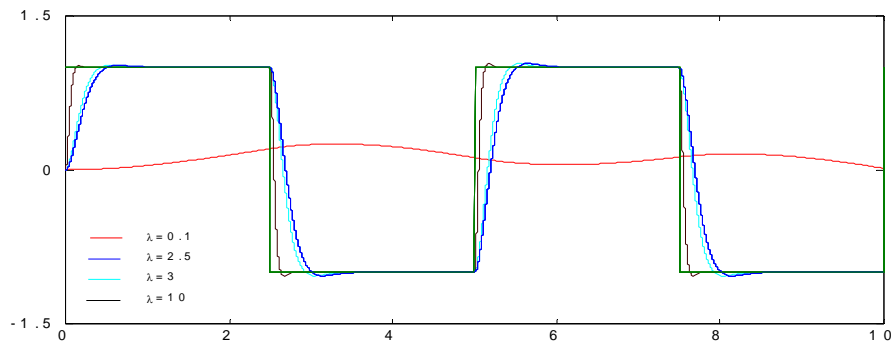


réponses de système

figure 2.12 présentation sortie/commande de système stable à PNM pour  $\lambda=2.5$ ,  $\gamma=2.5$  et variation de  $t_s$  en présence de perturbation.

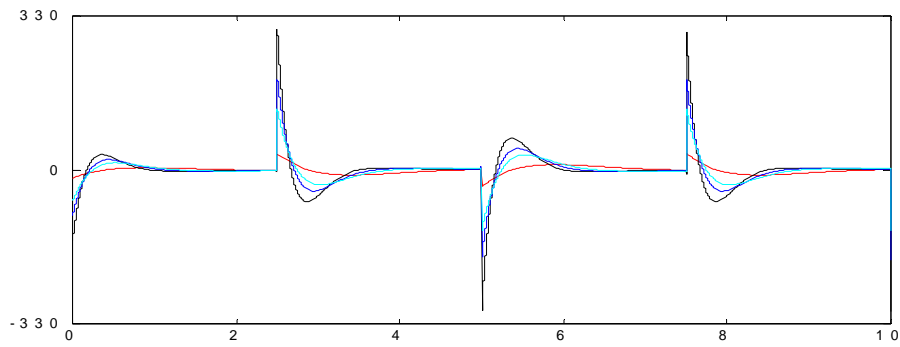


évolution des commandes

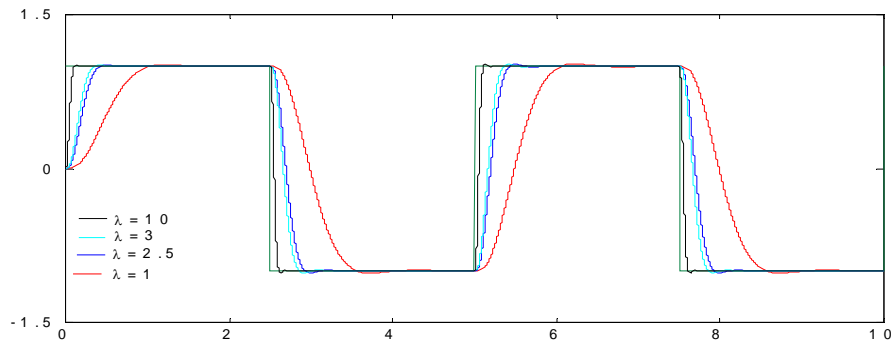


réponses de système

figure 2.13 présentation sortie/commande de système stable à PNM pour  $t_s=0.5$ ,  $\gamma=2.5$  et variation de  $\lambda$  en absence de perturbation

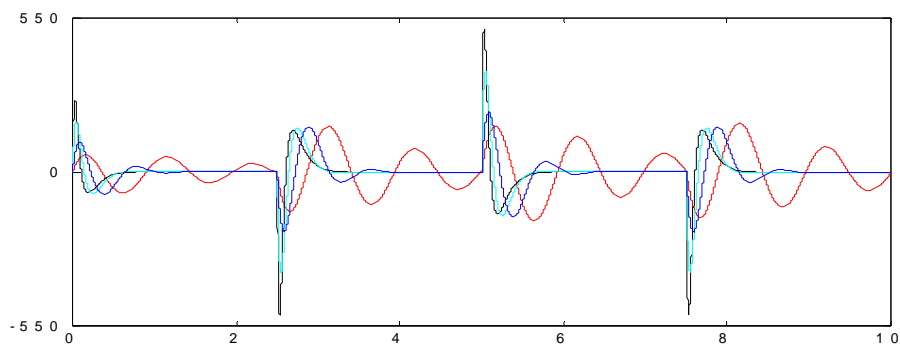


évolution des commandes

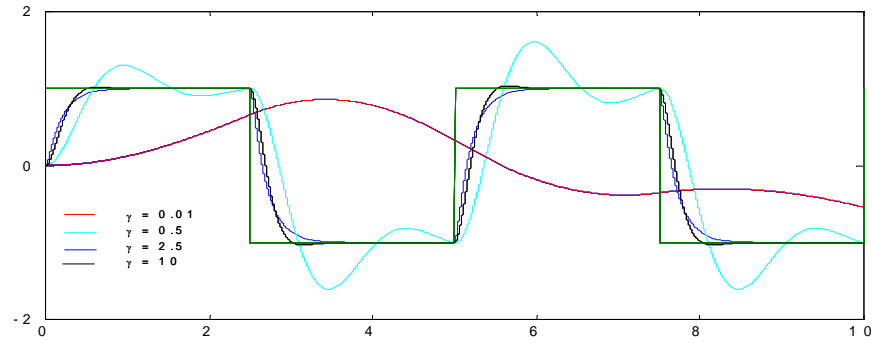


réponses de système

figure 2.14 présentation sortie/commande de système stable à PNM pour  $t_s=0.5$ ,  $\gamma=2.5$  et variation de  $\lambda$  en présence de perturbation.



évolution des commandes



réponses de système

figure 2.15 présentation sortie/commande de système stable à PNM pour  $t_s=0.5$ ,  $\lambda=2.5$  et variation de  $\gamma$  en absence de perturbation

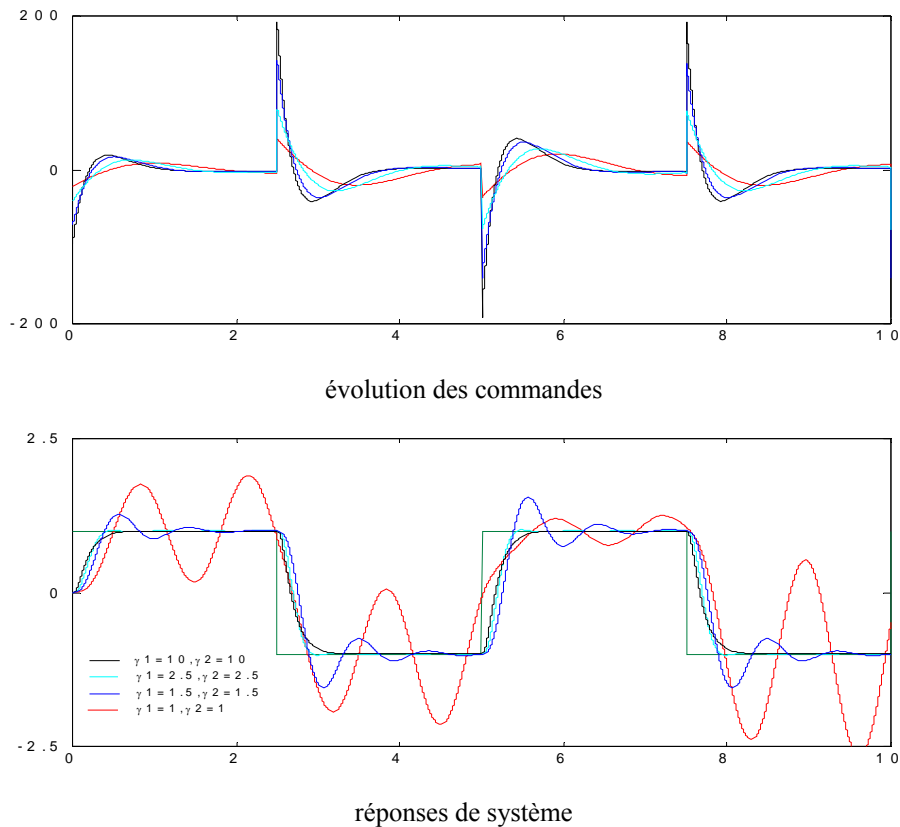


figure 2.16 présentation sortie/commande de système stable à PNM pour  $t_s=0.5, \lambda=2.5$  et variation de  $\gamma$  en présence de perturbation.

### 2.8.3 Etude d'un système instable

Soit à étudier le système instable suivant :

$$y(s) = \frac{1}{2 - 4s + 5s^2} u(p) \tag{2.24}$$

Présentant deux pôles instable respectivement  $s_1 = 0.4 + 0.48i, s_2 = 0.4 + 0.48i$ .

La réponse de ce système en boucle ouverte excité par une commande unitaire est donnée sur la figure 2.17. Les résultats de simulation concernant ce système présentent l'influence des paramètres  $t_s, \lambda, \gamma$  sur la sortie et la commande, on constate que l'instabilité des pôles en boucle ouverte n'a aucun effet sur la dynamique en boucle fermée évitant ainsi toute instabilité.

La simulation montre que l'effort de commande augmente avec la diminution de temps de réponse et diminue avec l'augmentation de temps de réponse.

L'augmentation de paramètre  $\lambda$  ( $\lambda > 3$ ) rend la commande puissante avec réduction de temps de réponse, mais la diminution de leur valeur donne l'atténuation de l'effort de commande avec augmentation de temps de réponse.

L'effet de variation de  $\gamma$  est illustré sur les figures 2.22 et 2.23, on remarque que la valeur standard  $\gamma=2.5$  donne de bonne performance, l'augmentation de ce paramètre rend la commande puissante, tandis que la diminution de  $\gamma$  donne la diminution de l'effort de commande avec augmentation de temps de réponse et le système risque de perdre sa stabilité.



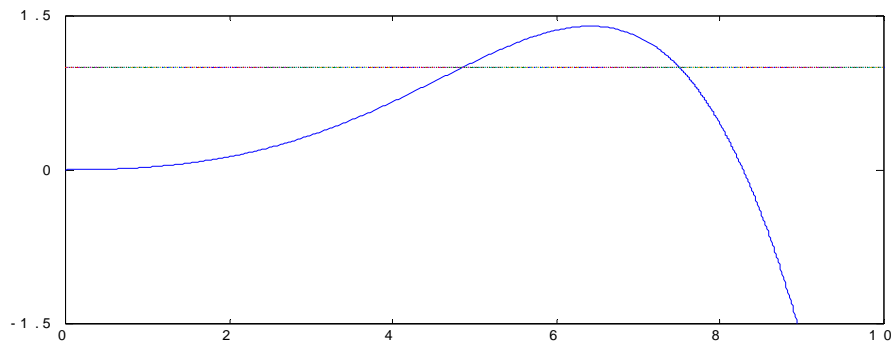
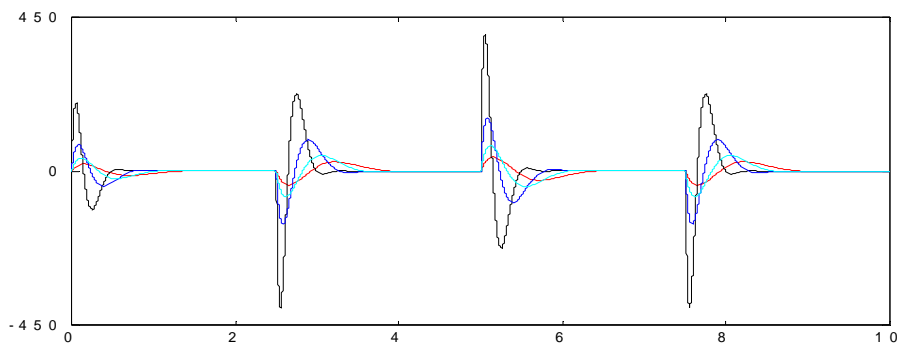
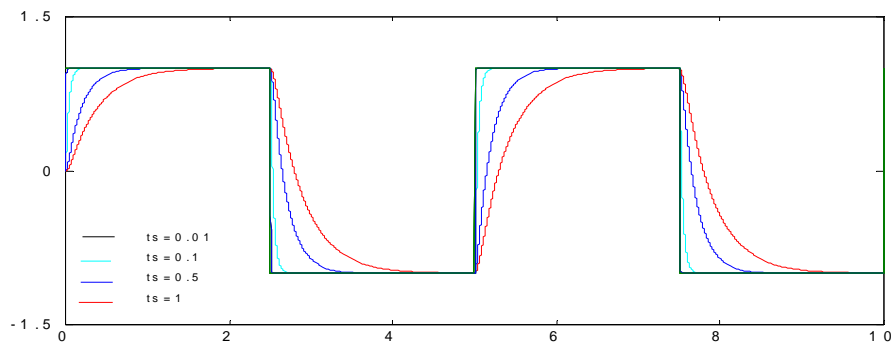


figure 2.17 réponse d'un système instable en boucle ouverte à une entrée échelon

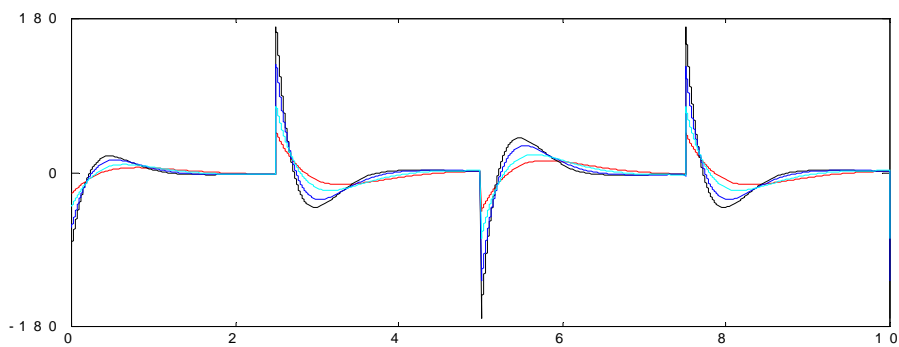


évolution des commandes

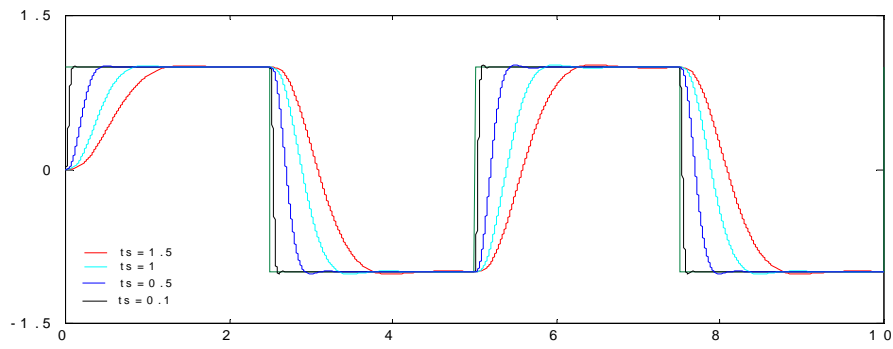


réponses de système

figure 2.18 présentation sortie/commande de système stable à PNM pour  $\lambda=2.5$ ,  $\gamma=2.5$  et variation de  $t_s$  en absence de perturbation

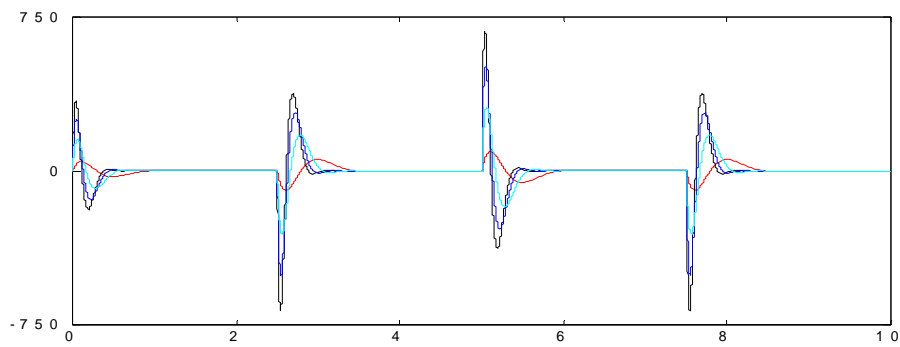


évolution des commandes

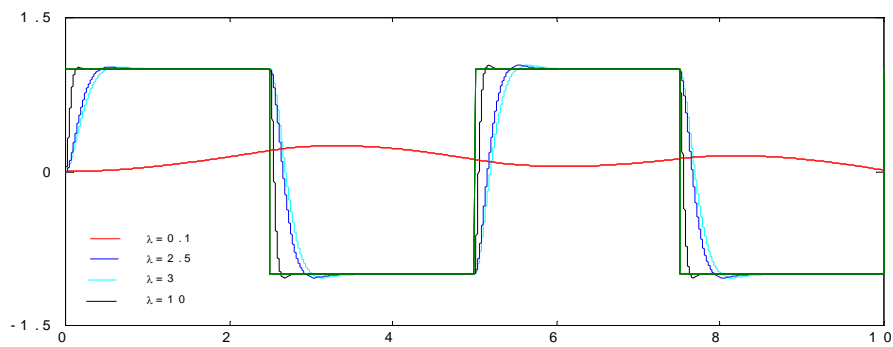


réponse de système

figure 2.19 présentation sortie/commande de système stable à PNM pour  $\lambda=2.5$ ,  $\gamma=2.5$  et variation de  $t_s$  en présence de perturbation.

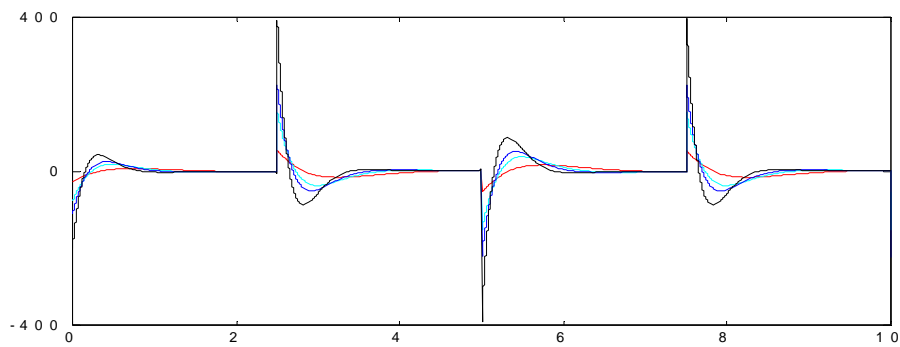


évolution des commandes

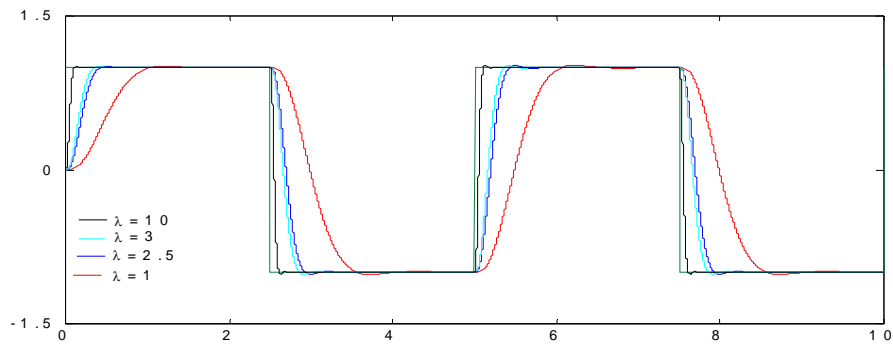


réponse de système

figure 2.20 présentation sortie/commande de système stable à PNM pour  $t_s=0.5$ ,  $\gamma=2.5$  et variation de  $\lambda$  en absence de perturbation.

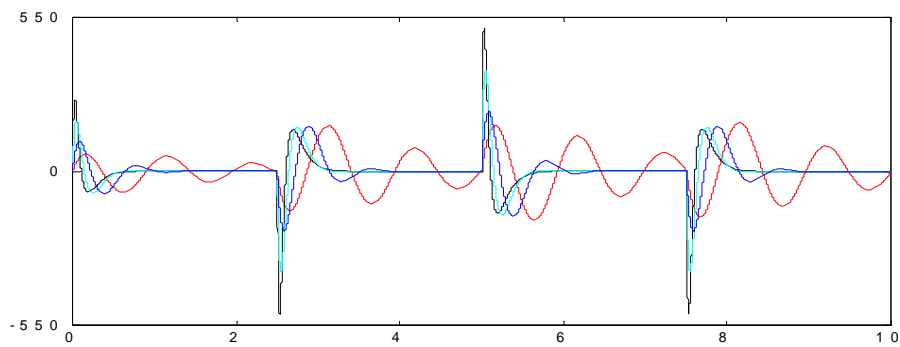


évolution des commandes

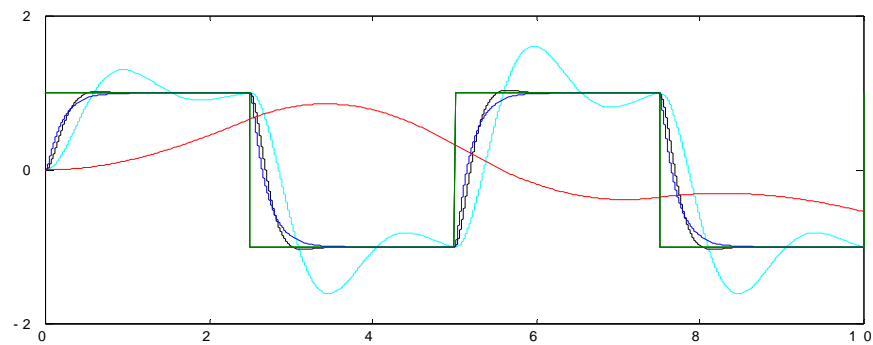


réponse de système

figure 2.21 présentation sortie/commande de système stable à PNM pour  $t_s=0.5$ ,  $\gamma=2.5$  et variation de  $\lambda$  en présence de perturbation.

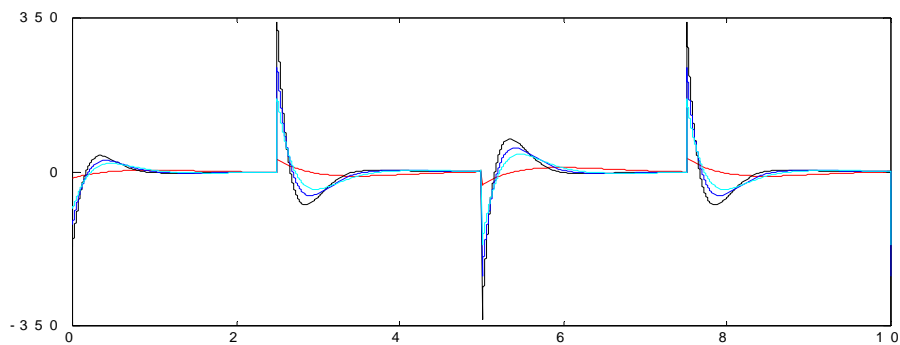


évolution des commandes

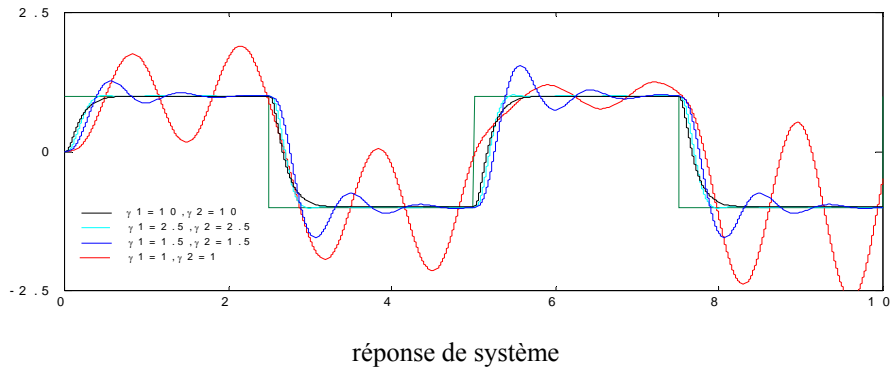


réponse de système

figure 2.22 présentation sortie/commande de système stable à PNM pour  $t_s=0.5, \lambda=2.5$  et variation de  $\gamma$  en absence de perturbation



évolution des commandes



réponse de système  
 figure 2.23 présentation sortie/commande de système stable à PNM pour  $t_s=0.5, \lambda=2.5$  et variation de  $\gamma$  en présence de perturbation.

### 2.8.4 Etude d'un système instable à phase non minimale

Soit le système suivant :

$$y(s) = \frac{-1+s}{1-2s+4s^2} u(s) \tag{2.26}$$

L'effet de variation de temps de réponse sur l'effort de commande et la sortie de ce système est représenté sur les figures 2.25 et 2.26, on constate l'augmentation de l'effort de commande par diminution de temps de réponse et la diminution de l'effort de commande par augmentation de temps de réponse.

Le choix de paramètre  $\lambda$  est très délicat. En effet, il est limité dans l'intervalle [2.5 3], l'augmentation de ce paramètre rend la commande puissante, néanmoins la diminution de  $\lambda$  donne une augmentation de temps de réponse avec diminution de l'effort de commande.

La valeur standard de  $\gamma$  donne de bonne performance, l'augmentation de l'indice de stabilité rend la commande puissante, mais la diminution de ce paramètre donne l'augmentation de temps de réponse et certain cas le système perdre sa stabilité.

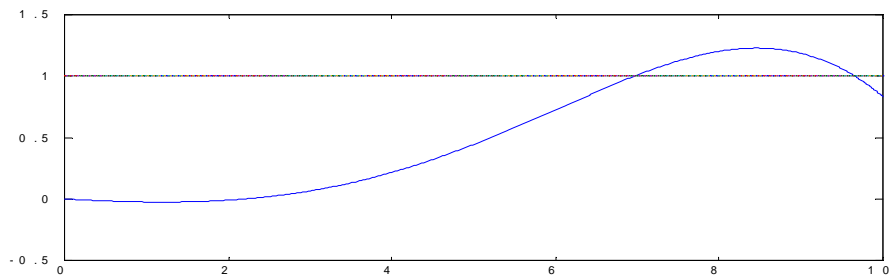
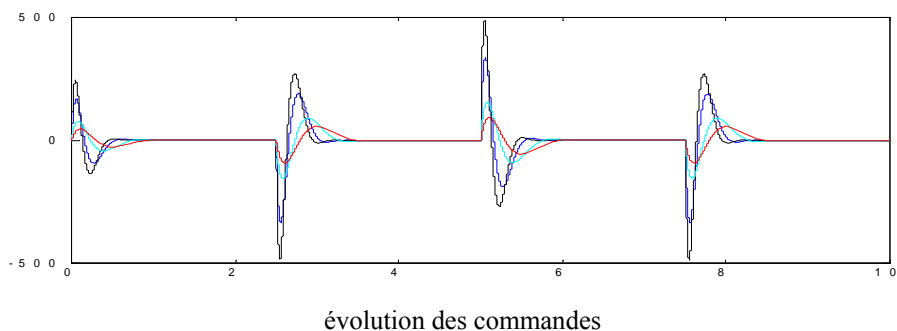
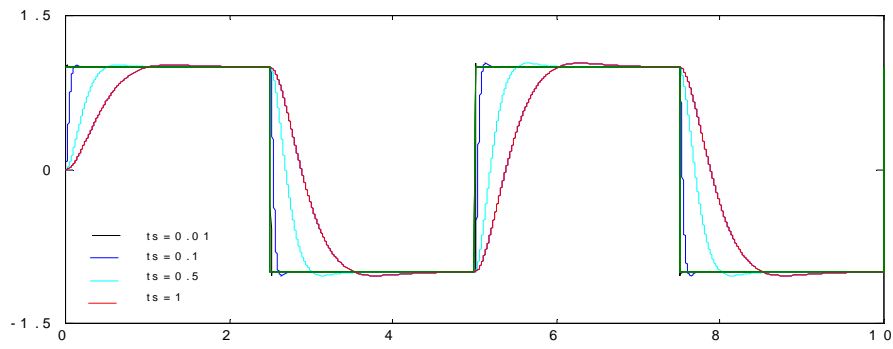


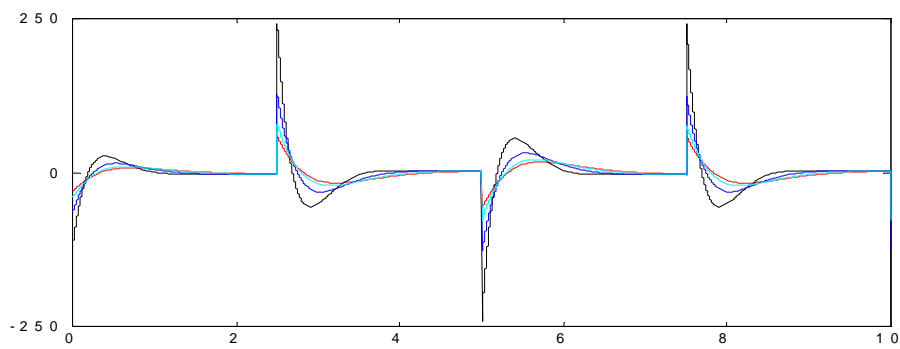
figure 2.24 réponse d'un système instable à PNM en boucle ouverte à une entrée échelon



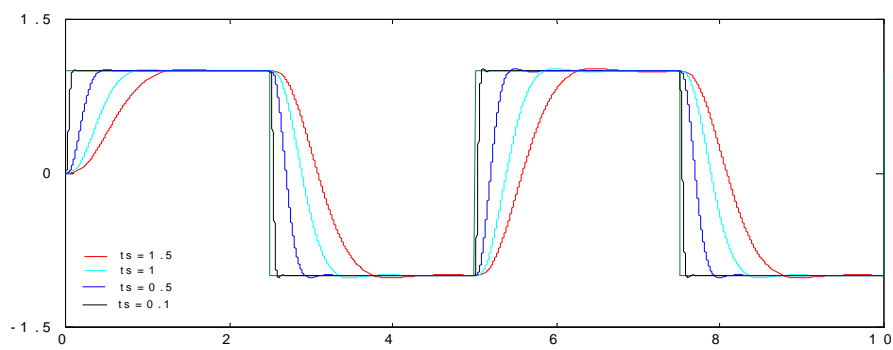


réponse de système

figure 2.25 présentation sortie/commande de système stable à PNM pour  $\lambda=2.5, \gamma=2.5$  et variation de  $t_s$  en absence de perturbation

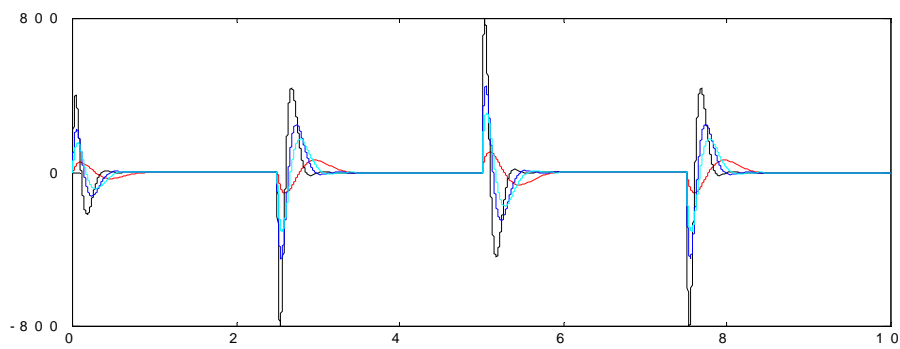


évolution des commandes

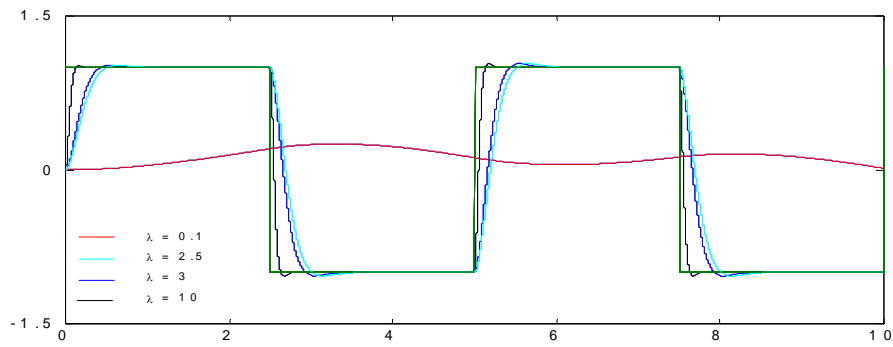


réponse de système

figure 2.26 présentation sortie/commande de système stable à PNM pour  $\lambda=2.5, \gamma=2.5$  et variation de  $t_s$  en présence de perturbation.

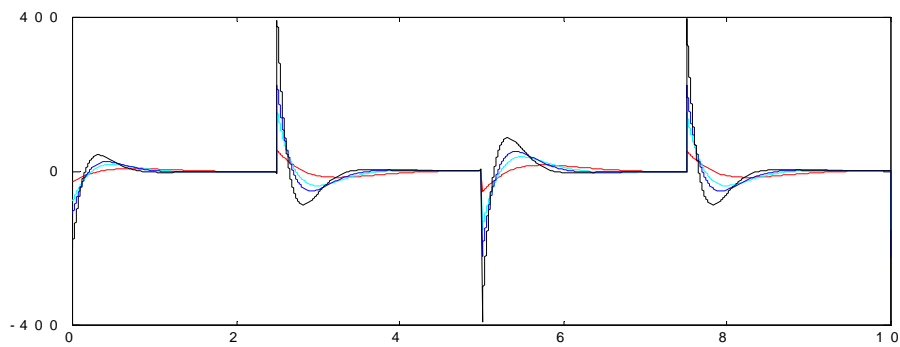


évolution des commandes

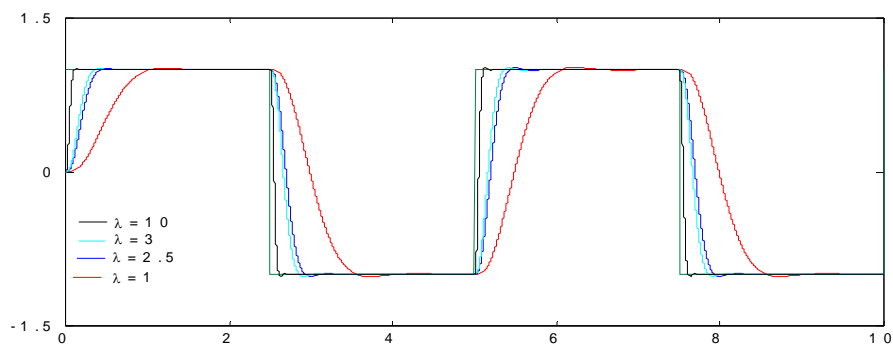


réponse de système

figure 2.27 présentation sortie/commande de système stable à PNM pour  $t_s=0.5$ ,  $\gamma=2.5$  et variation de  $\lambda$  en absence de perturbation

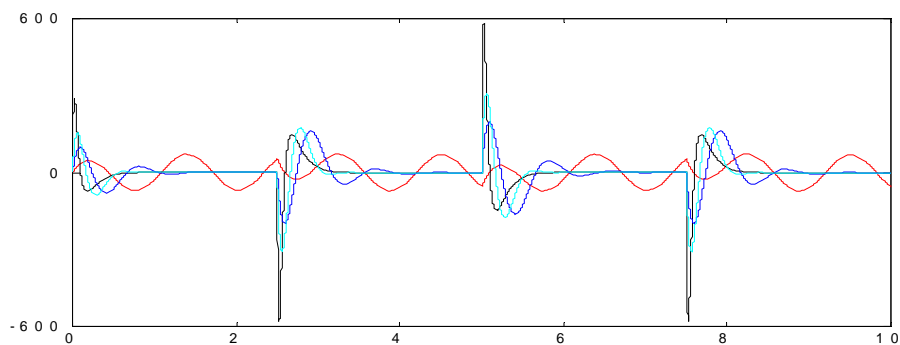


évolution des commandes



réponse de système

figure 2.28 présentation sortie/commande de système stable à PNM pour  $t_s=0.5$ ,  $\gamma=2.5$  et variation de  $\lambda$  en présence de perturbation.



évolution des commandes

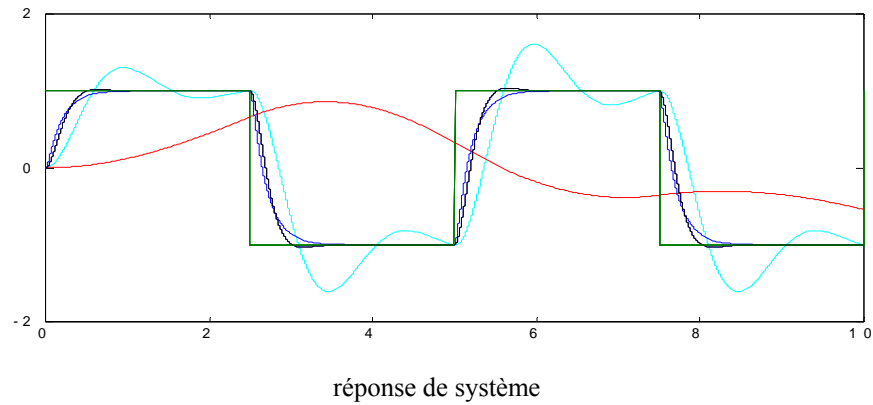


figure 2.29 présentation sortie/commande de système stable à PNM pour  $t_s=0.5, \lambda=2.5$  et variation de  $\gamma$  en absence de perturbation.

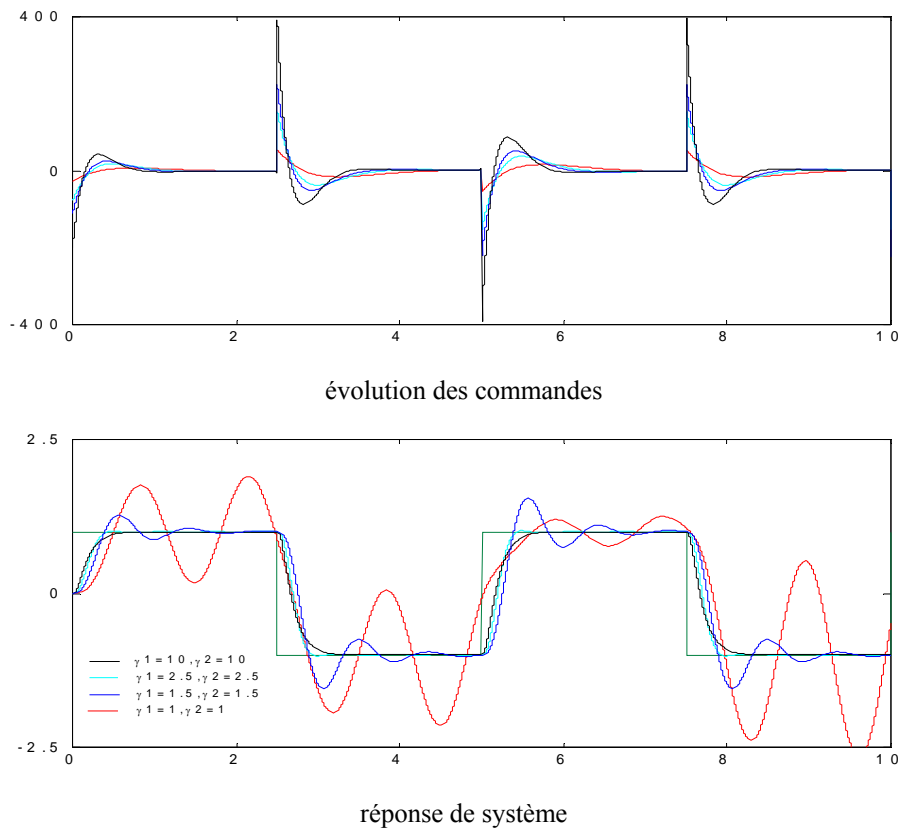


figure 2.30 présentation sortie/commande de système stable à PNM pour  $t_s=0.5, \lambda=2.5$  et variation de  $\gamma$  en présence de perturbation.

## 2.8 Influence de la sortie à une perturbation en régime établi

Pour tester l'efficacité de CDM à des perturbations en échelon, on applique au cours de l'évolution de la sortie en régime établi un échelon unitaire, les résultats de simulation pour les quatre types de système sont illustrés sur les figure 2.31, 2.32, 2.33, 2.34 respectivement, ils montrent que cette perturbation n'a aucun effet sur la sortie puisque le système poursuit son régime établi, cette compensation est du à la présence d'un intégrateur dans le contrôleur CDM.

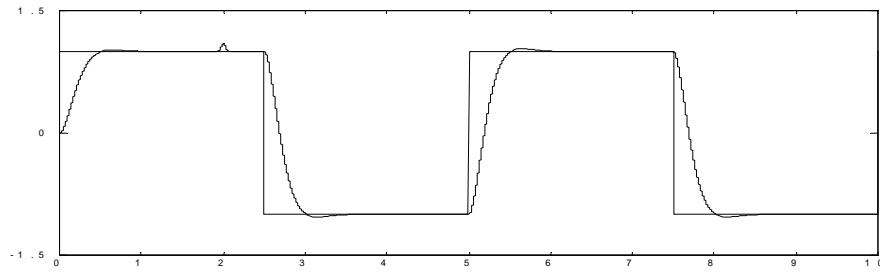
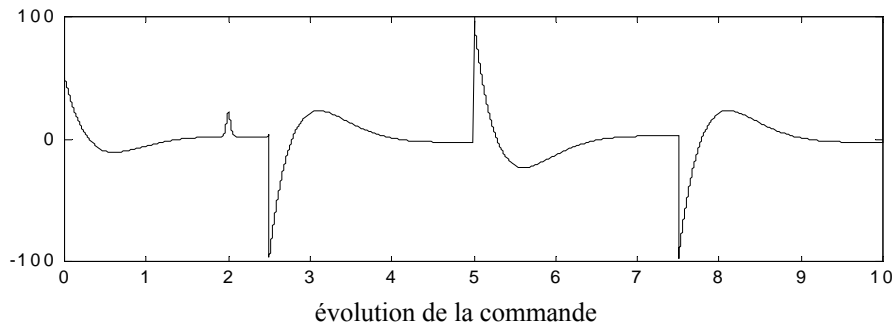


figure 2.31 présentation sortie/commande d'un système stable à une perturbation en échelon pour  $t_s=0.5$ ,  $\lambda=2.5$ ,  $\gamma=2.5$

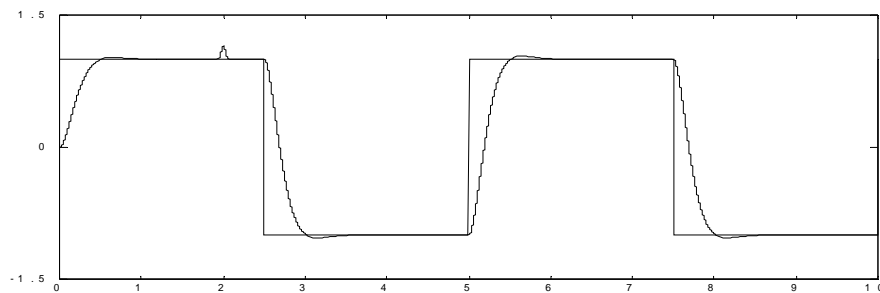
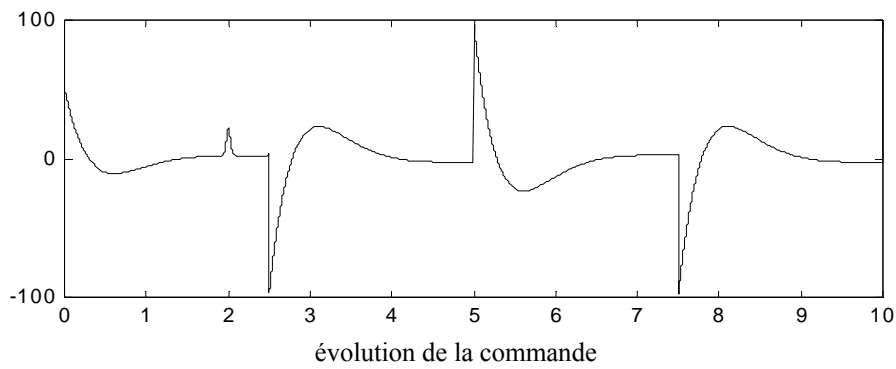
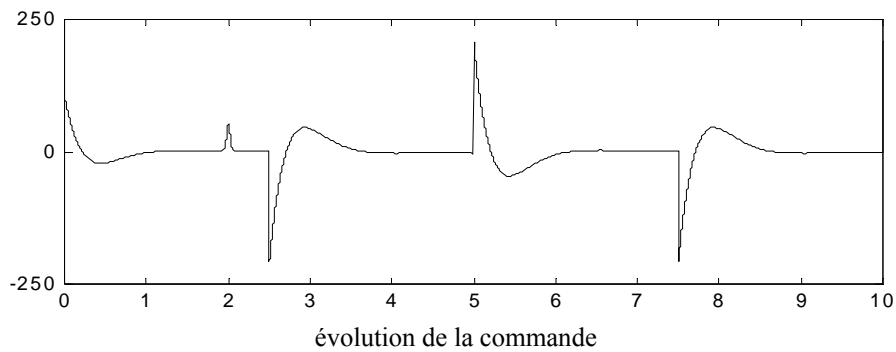


figure 2.32 présentation sortie/commande d'un système stable à PNM à une perturbation en échelon pour  $t_s=0.5$ ,  $\lambda=2.5$ ,  $\gamma=2.5$





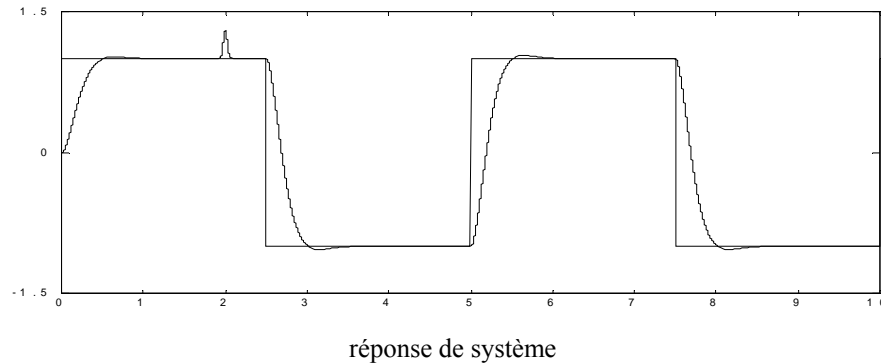


figure 2.33 présentation sortie/commande d'un système instable à une perturbation en échelon pour  $t_s=0.5$ ,  $\lambda=2.5$ ,  $\gamma=2.5$

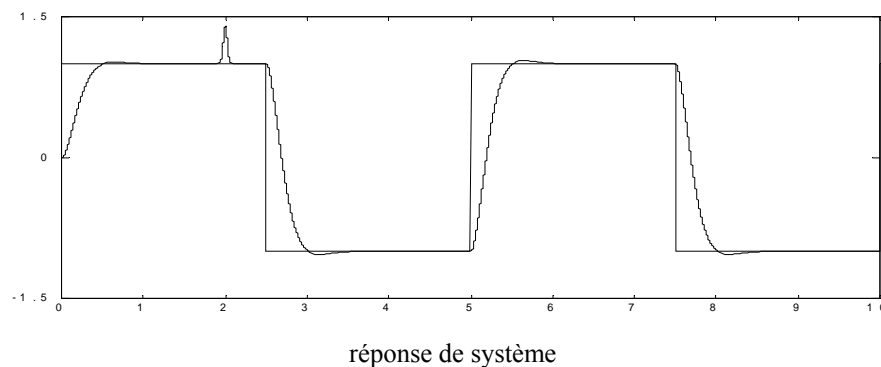
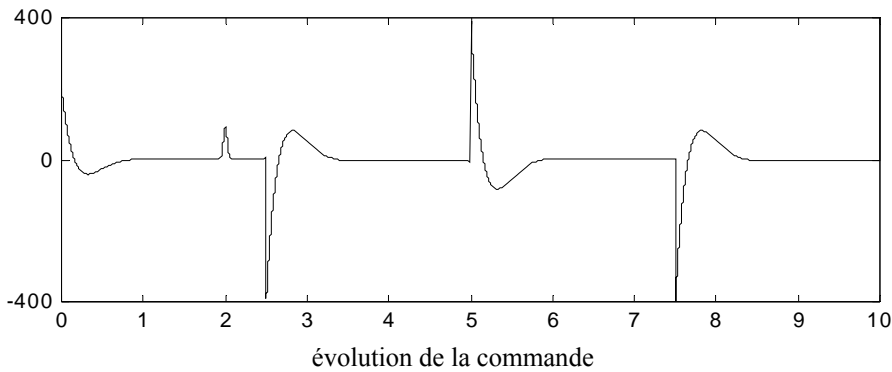


figure 2.34 présentation sortie/commande d'un système instable à PNM à une perturbation en échelon pour  $t_s=0.5$ ,  $\lambda=2.5$ ,  $\gamma=2.5$

## 2.9 Synthèse et discussion des résultats obtenus

- **Temps de réponse**

Le paramètre temps de réponse  $t_s$  est toujours choisi supérieur à la période d'échantillonnage  $t_e$ . La simulation montre que le temps de réponse est en relation avec la commande, ainsi le temps de réponse doit être choisis de façon que la commande soit réalisable et a des valeurs nominale

- **Le paramètre  $\lambda$**

Pour les différents systèmes étudiés, on a constaté que la valeur de  $\lambda=2.5$  donne des bonnes résultats dans l'intervalle  $[2.5 \ 3]$ , l'influence de ce paramètre est l'inverse de l'influence de  $t_s$  qui est illustré par l'équation (2.3), donc l'augmentation de ce paramètre donne la croissance de la commande avec diminution de temps de réponse et leur diminution donne la diminution de la commande avec augmentation de temps de réponse.

- **L'indice de stabilité  $\gamma$**

Le paramètre  $\gamma$  indique la stabilité de système, d'après les simulations précédentes on conclure que  $\gamma$  doit être choisis supérieur à  $\gamma^*$  et pas loin de  $\gamma$  standard pour que la commande reste réalisable et suffisante.

## 2.11 Conclusion

Ce chapitre est consacré au développement théorique de l'algorithme de commande CDM, suivit de la simulation de quatre types de systèmes présentant chacun des particularités spécifiques. Les constatations et les remarques ont été élaborées à partir des fonctions de transfert continues à paramètre connus.

L'étude précédente nous a affirmée que l'algorithme de CDM est capable de commandé des système simples et complexes et ceci grâce à un choix très particulier des paramètres de synthèse permettant d'élaborer des commandes douces et des sorties suivants exactement la référence (erreur de poursuite faible). Les performances des systèmes commandés par CDM sont grandement influencés par le choix de temps de réponse, le paramètre  $\lambda$  et l'indice de stabilité, donc une grande attention doit être accordé a leurs choix, pour spécifier le comportement en boucle fermée (la dynamique).

# *CHAPITRE III*

**Application de la commande CDM au robot  
manipulateur PUMA**

### 3.1 Introduction

Autre domaine d'actualité de grand intérêt et d'importance durant ces dernières années, a été la théorie de conception de contrôleurs robustes satisfaisant des exigences de stabilité et de performance pour un système, c'est-à-dire, de contrôleurs qui maintiennent ces exigences, non seulement pour les valeurs nominales, mais également pour les situations où le système est à modélisation incertaine et soumis à de fortes perturbations.

Durant ces dernières années la méthodologie de commande méthode de diagramme des coefficients a connu un développement important [14], En outre, la commande CDM est facile à implanter et simple à comprendre par des personnes avec de faibles connaissances en commande.

De plus, comme il a été expliqué dans le chapitre précédent, cette méthodologie présente la vertu de pouvoir être appliquée à des systèmes avec dynamique difficile, par exemple, ceux à comportement de phase non minimale ou instable, il permet en outre de traiter des systèmes multivariables et non linéaires.

La commande CDM de système multivariable est similaire à celle présenté par S.E. HAMAMCI et M.KOKSAL [15], elle est basée sur le modèle de représentation obtenu par le découplage du modèle, ce type de modèle facilite l'extension de l'algorithme de commande CDM des systèmes monovariabiles présentés précédemment au cas multivariable. Ceci revient à considérer plusieurs boucles de commandes indépendantes, dans le cas de commande indépendante, il est clair qu'aucun terme de couplage ne peut être présenté, la méthode de commande doit être capable de rejeter les perturbations tout en assurant la stabilité en boucle fermée. Le comportement non linéaire du modèle obtenu dans le chapitre de modélisation nécessite l'utilisation de la linéarisation exacte de modèle de robot qui permet d'obtenir un modèle linéaire et découplé du manipulateur, vu que la technique de commande s'applique au système linéaire.

### 3.2 Application de la commande CDM au robot manipulateur PUMA

La caractéristique importante de CDM est la simplicité, la robustesse et l'élimination de perturbation ou est nécessaire au robot manipulateur, pour illustrés les résultats précédents on applique cette technique de commande au robot PUMA.

$i=1,2$  désigne le contrôleur.

pour les deux contrôleurs, les polynômes de commande CDM en absence de perturbation sont:

$$A_i(s) = l_i s + l_{0i} \quad (3.1)$$

$$B_i(s) = k_{1i} s + k_{0i} \quad (3.2)$$

$$F_i(s) = a_0 \quad (3.3)$$

tel que :  $l_i, l_{0i}, k_{1i}, k_{0i}$  sont les paramètres des deux contrôleurs.

les polynômes caractéristiques en boucle fermée et en fonction des paramètres sont :

$$P(s) = a_{2i} s^2 + a_{1i} s + a_{0i} \quad (3.4)$$

la forme standard de CDM nous donne les indices de stabilité de commande du robot manipulateur PUMA suivants:

$$\gamma_{1i} = 2.5 \text{ et } \gamma_{0i} = \gamma_{2i} = \infty \quad (3.5)$$

et les limites de stabilité sont:

$$\gamma_{1i}^* = 0.4 \quad (3.6)$$

les paramètres des contrôleurs sont choisis comme suit :

$$l_i = [0 \quad 6.4 \times 10^{-3}] \quad (3.7)$$

$$k_i = [0.4 \quad 10] \quad (3.8)$$

sont obtenu pour un  $t_{si} = 0.1$  s

finalement on obtient les polynômes caractéristiques suivants :

$$P_i(s) = 6.4 \times 10^{-3} s^2 + 0.4 s + 10 \quad (3.9)$$

en présence de perturbations les polynômes de contrôleurs sont choisis comme suit :

$$A_i(s) = l_i s \quad (3.10)$$

$$B_i(s) = k_{0i} + k_{1i} s + k_{2i} s^2 \quad (3.11)$$

$$F_i(s) = a_{0i} \quad (3.12)$$

où  $l_i, k_{2i}, k_{1i}, k_{0i}$  sont les paramètres des deux contrôleurs

les polynômes caractéristiques en boucle fermée et en fonction des paramètres sont :

$$P(s) = a_3 s^3 + a_2 s^2 + a_1 s + a_0 \quad (3.13)$$

La forme standard de CDM nous donne les indices de stabilité de commande du robot manipulateur PUMA suivante:

$$\gamma_{1i} = 2.5, \gamma_2 = 2 \text{ et } \gamma_{0i} = \gamma_3 = \infty \quad (3.14)$$

donc

$$\gamma_i = [2.5 \quad 2] \quad (3.15)$$

et les limites de stabilité sont:

$$\gamma_i^* = [0.5 \quad 0.4] \quad (3.16)$$

les paramètres des deux contrôleurs sont choisies comme suit :

$$l_i = [5.12 \times 10^{-5} \quad 0] \quad (3.17)$$

$$k_i = [6.4 \times 10^{-3} \quad 0.4 \quad 10] \quad (3.18)$$

finalement on obtient le polynôme caractéristique suivant :

$$P(s) = 5.12 \times 10^{-5} s^3 + 6.4 \times 10^{-3} s^2 + 0.4 s + 10 \quad (3.19)$$

sont obtenu pour un  $t_{si} = 0.1$  s

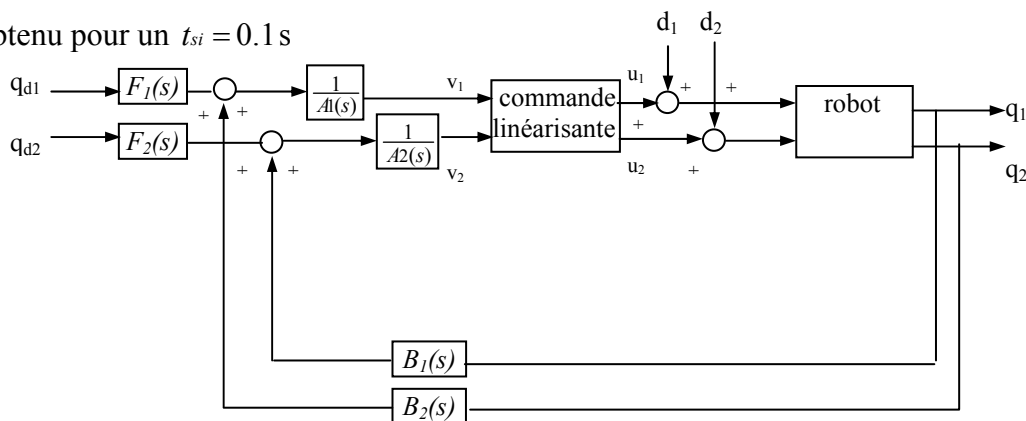


figure 3.1 Schéma bloc de la commande CDM d'un robot manipulateur PUMA

tel que :

$d_1$  et  $d_2$  sont les perturbations et  $m_1$  et  $m_2$  sont les bruits de mesure.

### 3.3 Influence des paramètre de CDM sur la commande de robot

Dans cette section on fait varier les paramètres de CDM tel que  $t_s$ , le paramètre  $\lambda$  et l'indice de stabilité  $\gamma$  appliqué au robot manipulateur PUMA pour tester les performances et l'efficacité de l'algorithme de commande. Les figures ci-dessous représentent l'influence des paramètres de CDM en absence et en présence des perturbations.

On prend  $t_{s1} = t_{s2} = t_s$ ,  $\lambda_1 = \lambda_2 = \lambda$ .

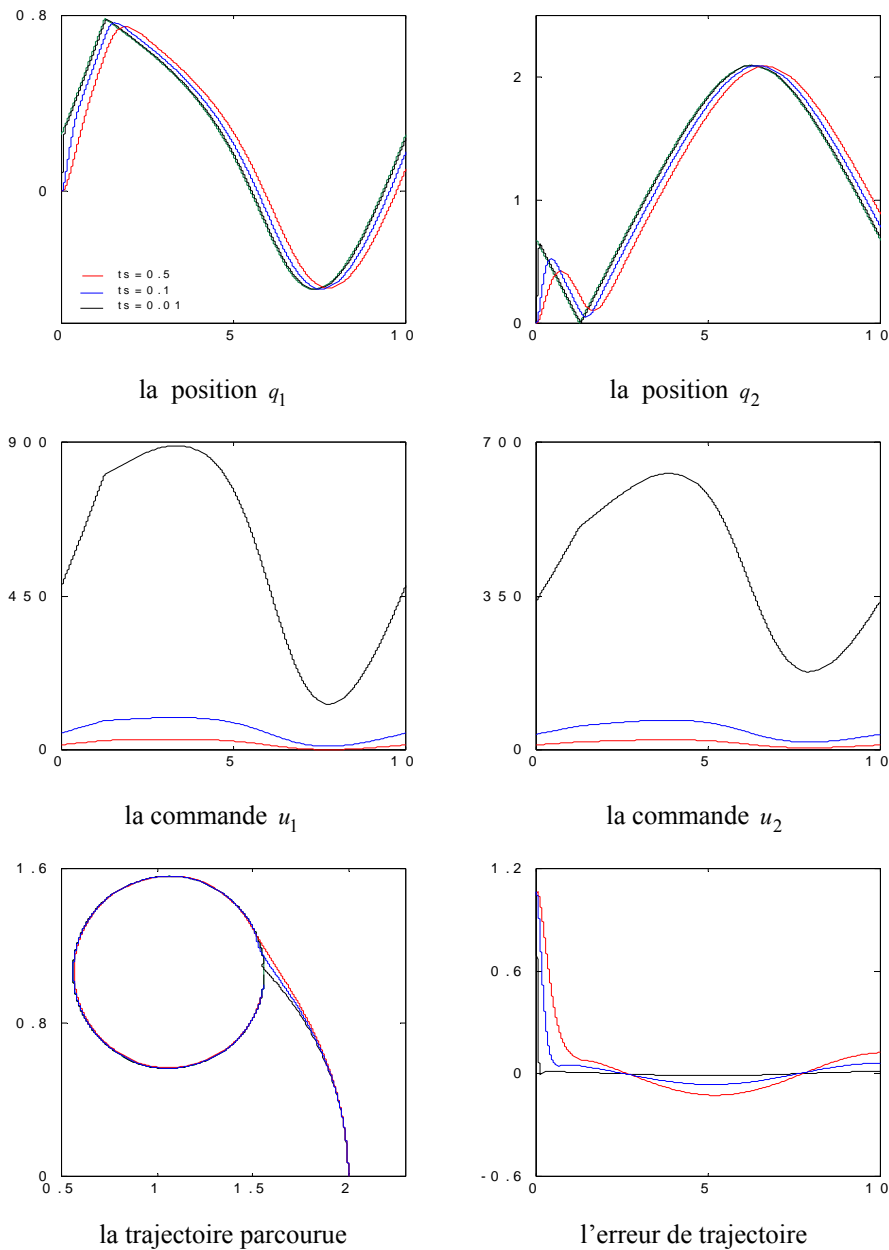


figure 3.2 trajectoire parcourus et désiré, positions  $q_1$ ,  $q_2$  et commandes  $u_1$ ,  $u_2$  pour les paramètres de CDM  $\lambda = 2.5$ ,  $\gamma_i = 2.5$  et variation de  $t_s$  en absence de perturbation

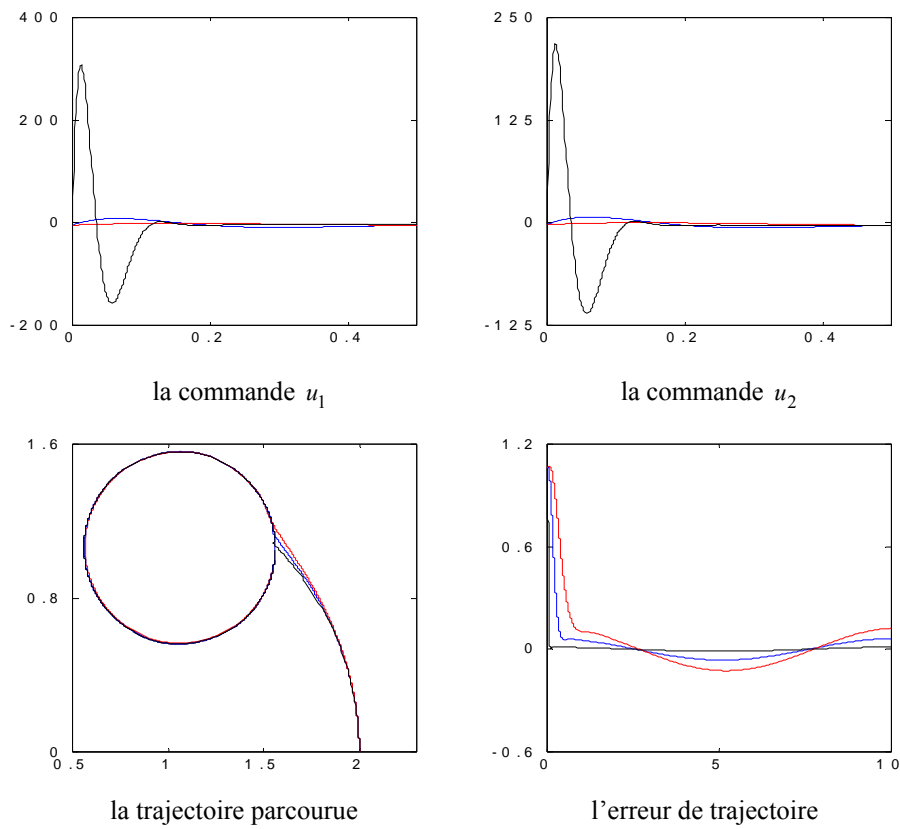
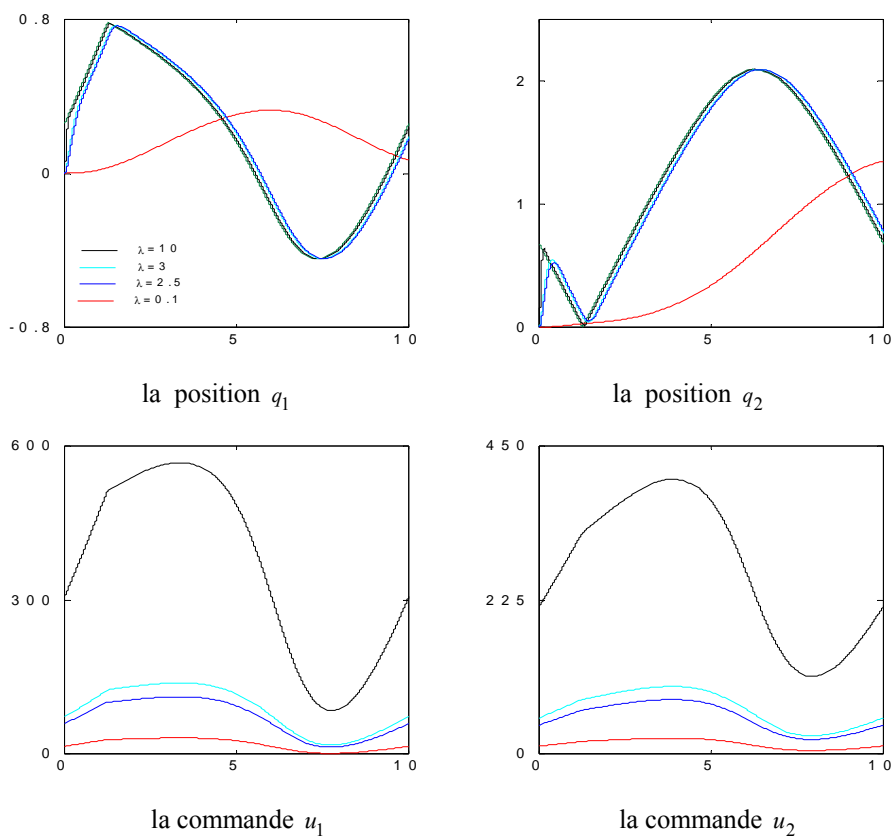
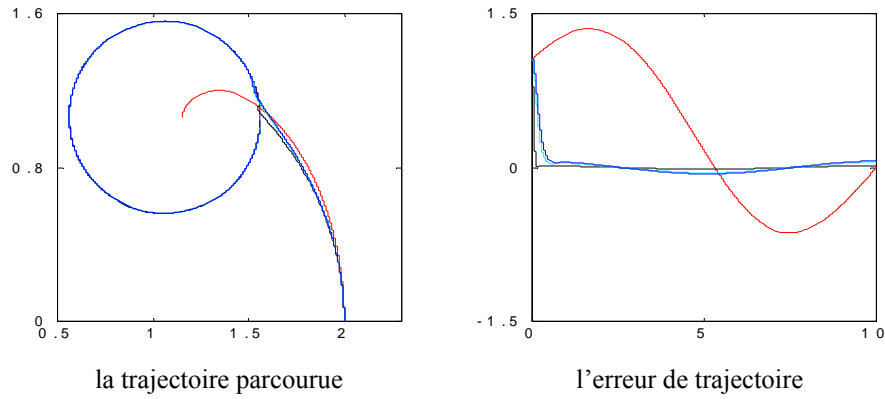
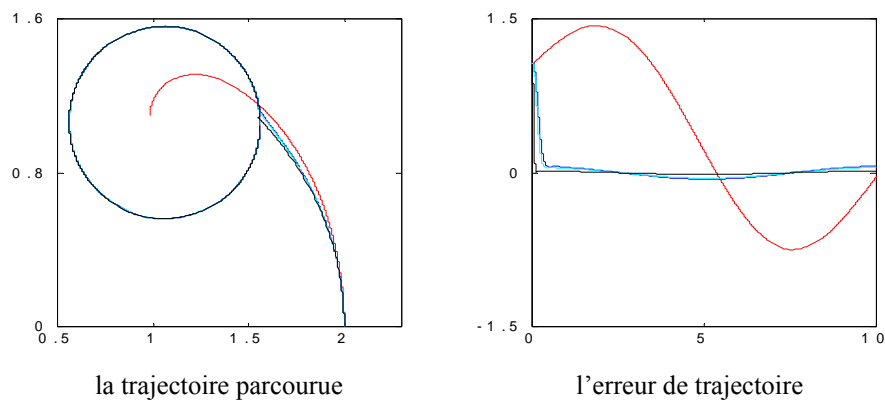
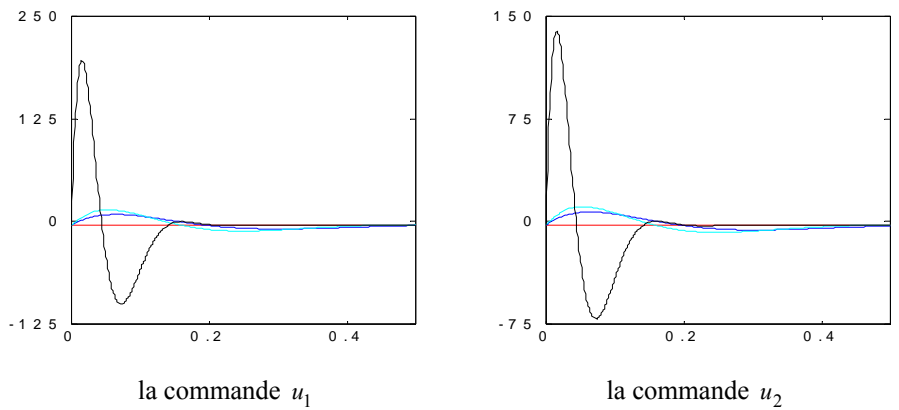


figure 3.3 trajectoire parcourus et désiré, positions  $q_1$ ,  $q_2$  et commandes  $u_1$ ,  $u_2$  pour les paramètres de CDM  $\lambda=2.5, \gamma_1=2.5, \gamma_2=2$  et variation de  $t_s$  en présence de perturbation

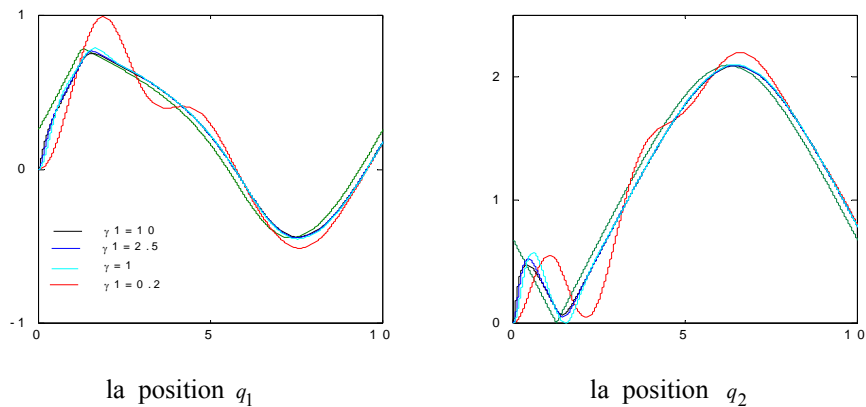




la trajectoire parcourue l'erreur de trajectoire  
 figure 3.4 trajectoire parcourus et désiré, positions  $q_1$ ,  $q_2$  et commandes  $u_1$ ,  $u_2$  pour les paramètres de CDM  $t_s=0.1$ ,  $\gamma_i=2.5$  et variation de  $\lambda$  en absence de perturbation



la trajectoire parcourue l'erreur de trajectoire  
 figure 3.5 trajectoire parcourus et désiré, positions  $q_1$ ,  $q_2$  et commandes  $u_1$ ,  $u_2$  pour les paramètres de CDM  $t_s=0.1$ ,  $\gamma_1=2.5$ ,  $\gamma_2=2$  et variation de  $\lambda$  en présence de perturbation



la position  $q_1$  la position  $q_2$



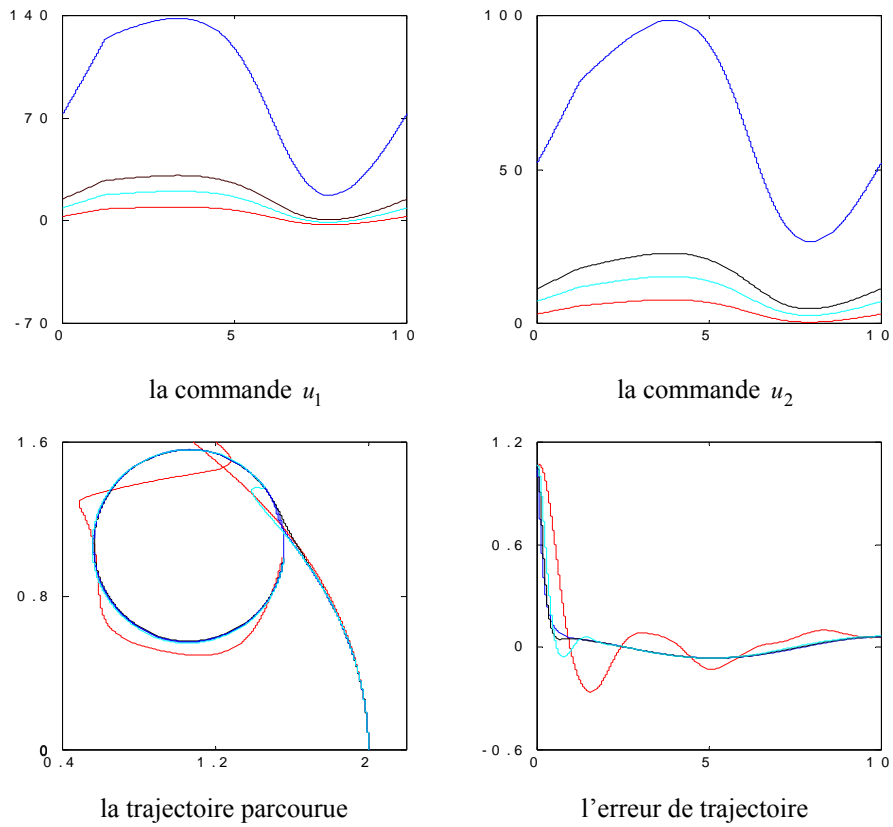
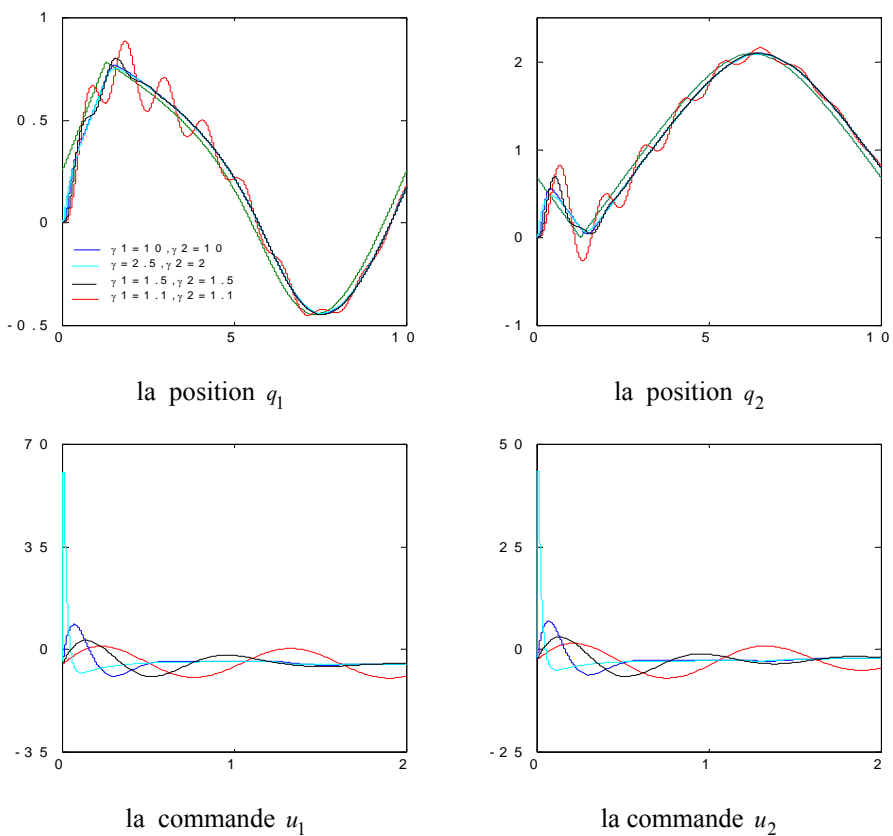


figure 3.6 trajectoire parcourus et désiré, positions  $q_1$ ,  $q_2$  et commandes  $u_1$ ,  $u_2$  pour les paramètres de CDM  $t_s=0.1$ ,  $\lambda=2.5$  et variation de  $\gamma_i$  en absence de perturbation



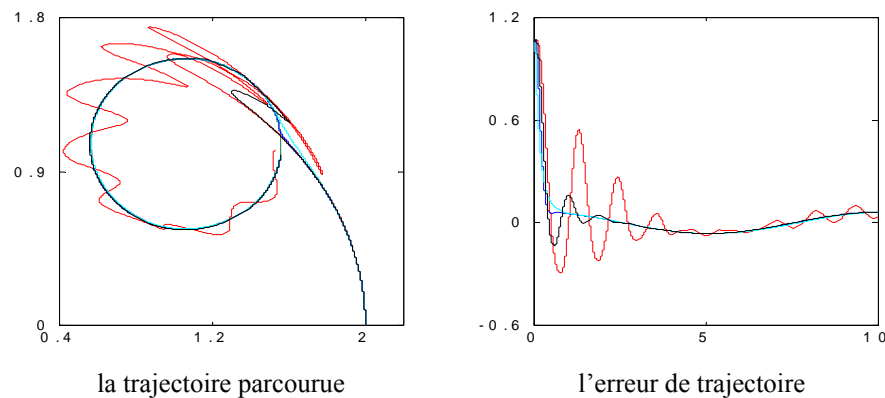


figure 3.7 trajectoire parcourus et désiré, positions  $q_1$ ,  $q_2$  et commandes  $u_1$ ,  $u_2$  pour les paramètres de CDM  $t_s = 0.1$ ,  $\lambda = 2.5$  et variation de  $\gamma_i$  en présence de perturbation

### 3.4 Résultats de simulation

Après avoir étudié en détail le développement et les performances de la commande CDM, on a proposé dans cette partie l'application de l'algorithme de CDM au modèle de robot PUMA après la linéarisation exacte présentés dans le premier chapitre.

Le modèle de robot a été simulée avec un pas de calcul de  $dt=10^{-3}s$ , en utilisant la méthode de R.K du quatrième ordre. La trajectoire imposée au robot est une trajectoire circulaire dans l'espace, L'étude par simulation nous montrera la puissance et les capacités de l'algorithme de commande qui donne une bonne poursuite de trajectoire.

Tout les résultats sont présentés sur des figures dont chacun comporte, les positions et les positions désirées, les commandes, la trajectoire parcourue et l'erreur de trajectoire pour chaque variable généralisée ( $q_1$ ,  $q_2$ )

#### Remarque

$t_{si}$ ,  $\lambda_i$ ,  $\gamma_i$  représentent les paramètres de synthèse de la commande à savoir.

- $t_{si}$  est le temps de réponse
- $\lambda_i$  paramètre à choisir
- $\gamma_i$  indice de stabilité

A partir du choix des paramètres de synthèse de la commande CDM, les résultats de simulation concernant le robot PUMA pour la trajectoire circulaire sont présentés.

Une augmentation de temps du réponse  $t_{s1}=t_{s2}=0.1s$  et  $t_{s1}=t_{s2}=0.5s$  illustrée sur les figures 3.2 et 3.3 donne la diminution de l'effort de commande et la diminution de ce paramètre donne la l'augmentation de l'effort de commande.

L'influence de paramètre  $\lambda$  sur la commande de système est présentée sur les figures 3.4 et 3.5, on constate que dans l'intervalle  $[2.5 \ 3]$ , la valeur 2.5 donne de bonne résultats, mais pour des valeurs supérieur à 3 on constate l'augmentation de l'effort de commande et diminution de temps de réponse. Par contre, pour des valeurs de  $\lambda$  inférieur à 2.5 la simulation montre la diminution de l'effort de commande et augmentation de temps de réponse.

L'effet de paramètre  $\gamma$  sur la stabilité et la dynamique de robot est donnée sur les figures 3.6 et 3.7, on constate l'influence de l'augmentation de ce paramètre ( $\gamma > 2.5$ ) qui rend la commande puissante, tandis que la diminution de  $\gamma < \gamma^*$  donne l'augmentation de temps de réponse avec

diminution de l'effort de commande et certain valeur de  $\gamma$  rend le système instable.

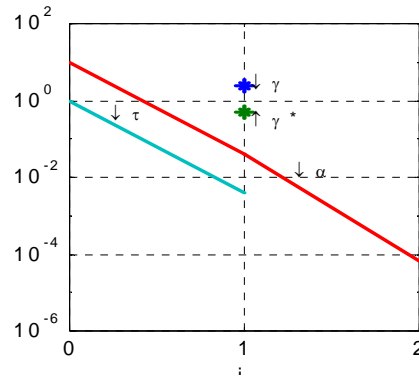


figure 3.8 le diagramme des coefficients de commande de robot PUMA en absence de perturbations

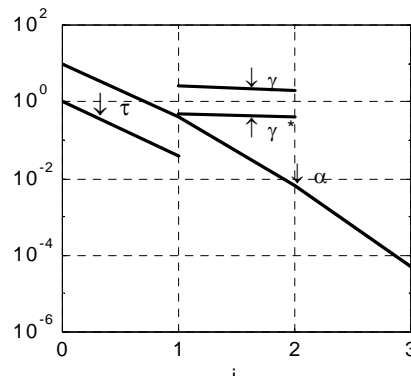
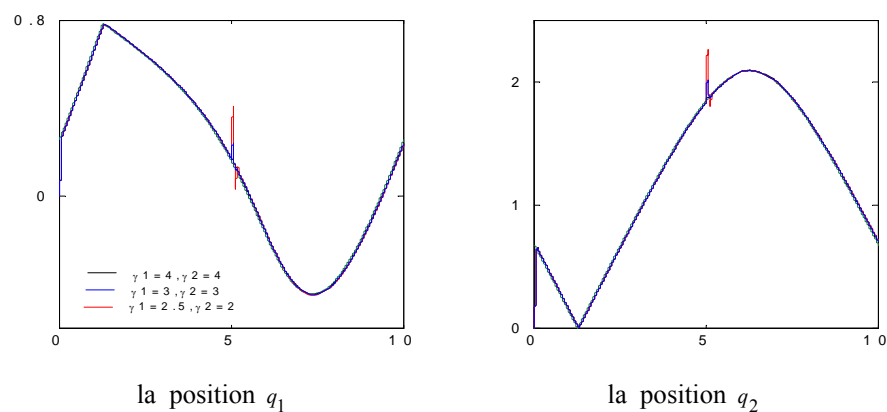


figure 3.9 le diagramme des coefficients de commande de robot PUMA en présence de perturbations

Les diagrammes des coefficients correspondant au correcteurs CDM déterminés sont représentés sur les figures 3.8 et 3.9, d'après les interprétations du diagramme des coefficients, données précédemment en s'intéressant à la convexité de courbe des  $a_i$ , on remarque que la commande est robuste et assure la rapidité désirée, la réponse temporelle de système est représentée dans la figure 3.10 qui montre que les paramètres de régulateur choisis sont capables de commander le robot en présence de perturbation, La figure 3.11 montre que la réponse en boucle fermée réalise les performances désirées avec application d'une variation paramétrique au robot, ainsi le contrôleur est robuste, ça montre que la boucle fermée réalise les performances désirées.

L'amélioration de réponse du système se fait par l'ajustement de l'indice de stabilité  $\gamma_i$ , illustré sur les figures 3.10, et 3.11



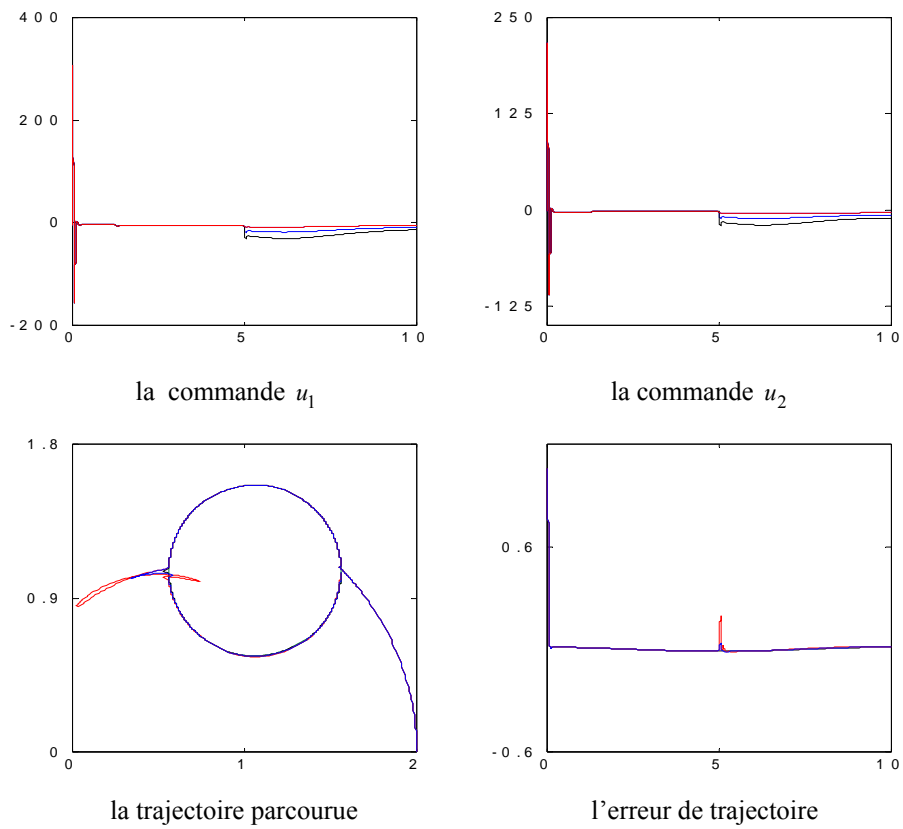
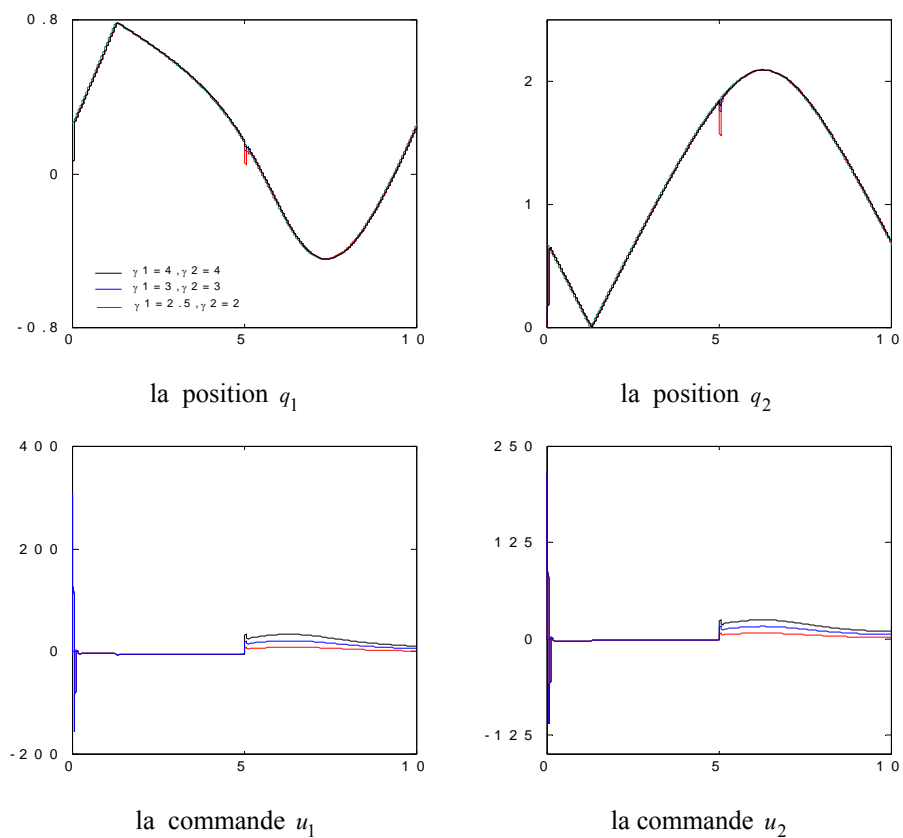


figure 3.10 trajectoire parcourue et désiré, positions  $q_1$ ,  $q_2$  et commandes  $u_1$ ,  $u_2$  pour les paramètres de CDM  $t_s = 0.1$ ,  $\lambda = 2.5$  et variation de  $\gamma_i$  avec application de la perturbation



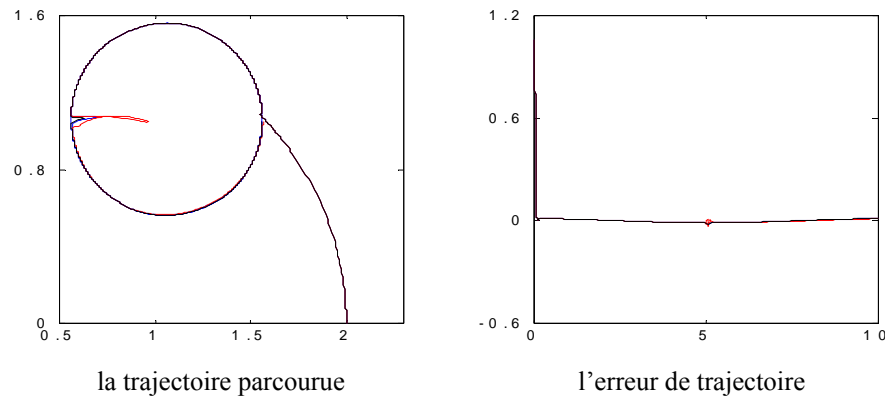


figure 3.11 trajectoire parcourue et désiré, positions  $q_1$ ,  $q_2$  et commandes  $u_1$ ,  $u_2$  pour les paramètres de CDM  $t_s=0.1$ ,  $\lambda = 2.5$  et variation de  $\gamma_i$  avec variation paramétrique

## Conclusion

Ce chapitre a été consacré à l'application de la CDM au robot manipulateur PUMA qui évolue dans l'espace à deux dimensions

Après un bref aperçu sur les différents travaux effectués jusqu'à présent par plusieurs chercheurs dans le domaine multivariable, nous décrivons le modèle de représentation continu multivariable, à partir d'un modèle sous forme de fonction de transfert linéaire et découplé.

Il est clair d'après les résultats obtenus, qu'une valeur de temps de réponse doit être choisie pour que les commandes soient suffisantes et réalisables et on conclut qu'il y a un compromis entre le temps de réponse et de la commande, en effet la diminution de temps de réponse rend les commandes plus actives. Les valeurs de paramètre  $\lambda$  supérieures à 3 rendent les commandes fortes, mais les valeurs inférieures à 2.5 donnent la diminution de la commande et l'augmentation de temps de réponse. L'indice de stabilité  $\gamma$  indique la stabilité du système, on remarque que les valeurs de  $\gamma$  supérieures à 2.5 donnent la diminution de temps de réponse et rendent la commande puissante, tandis que les valeurs inférieures à la limite de stabilité  $\gamma^*$  donnent l'augmentation de temps de réponse et la diminution de l'effort de commande.

Enfin nous pouvons conclure qu'une bonne poursuite de trajectoires désirées pour le robot manipulateur PUMA en présence de perturbation et de variation paramétrique de modèle ne peut être réalisée que par un choix adéquat des paramètres de synthèse de la commande CDM.

# CHAPITRE IV

**Etude de la commande prédictive généralisée**

## 4.1 Introduction

Depuis 1959, la commande par ordinateur a fait de grands progrès, tant sur le plan matériel que sur le plan méthode [16]. Le développement extraordinaire des microprocesseurs a provoqué des changements importants dans la conception des systèmes de commande. Leurs performances et leur faible coût les rendent aptes à prendre intégralement en charge les aspects de commande avec des performances nettement supérieures à celles des régulateurs analogiques [16]. Pour tirer réellement profit des capacités des microprocesseurs, il ne suffit pas de reproduire le comportement des régulateurs classiques analogiques, mais il faut mettre en œuvre des techniques d'automatique spécifiques et plus performantes, développées pour la commande par ordinateur.

À la fin des années 80, l'un des derniers algorithmes de commande prédictive est né, à savoir, l'algorithme de commande prédictive généralisée (GPC) [17], cette stratégie de commande appartient à la famille des algorithmes à horizon étendu. En effet la sortie d'un système n'est plus prédite à un instant proche du retard mais à des instants futurs, relatifs à un certain intervalle choisi [18].

Cette loi prend en charge les caractéristiques structurelles du processus, le changement de ces caractéristiques est dû à la constante de temps des éléments constituant le processus et aux blocs d'acquisition et de conversion de signaux [18]. L'idée principale est de transformer le problème classique de commande de la sortie basée sur l'instant présent, par la commande d'une prédiction de la sortie à un ou plusieurs instants futurs. Cette notion de sortie présente consiste à prendre en compte l'effet du retard pourvu que cette prédiction soit faite à un instant plus au moins égal à l'instant présent augmenté du retard.

Une de ces méthodes appelée commande prédictive généralisée a été proposée par CLARK en 1987 dans cette technique utilisée, le modèle de représentation CARIMA, est un système quadratique au sens d'un horizon fuyant qui minimisera cette commande.

Dans ce chapitre nous donnons la représentation de l'algorithme de commande prédictive généralisée, avec le calcul de la sortie prédite et le prédicteur nous présentons une étude dans laquelle on donne à la loi de commande prédictive généralisée.

## 4.2 Développement de la commande prédictive généralisée

La G.P.C est l'un des derniers membres de la famille des commandes à horizon étendu, elle se veut une généralisation de certains algorithmes et se base sur cinq concepts [19] :

- Le modèle de prédiction utilisé CARIMA est une extension du modèle CARMA, où on incorpore un effet intégral dans le but d'éliminer l'écart permanent de l'effet des perturbations constantes.
- L'utilisation de la prédiction à horizon fini supérieur au retard.
- La résolution récursive de l'équation de DIOPHANTINE.
- L'introduction de la pondération sur les incréments de commande dans le critère.
- Le choix de l'horizon de commande à partir duquel, tous les incréments de commande sont pris égaux à zéro.

### 4.2.1 Modélisation

La première phase a entamée pour la synthèse d'une commande est la modélisation du procédé a contrôlée, pour cela le modèle adopté doit représente les dynamiques tout en offrant une structure compatible et conforme avec le calcul de la loi de commande, la représentation choisi repose sur un modèle paramétrique linéaire perturbé [19], qui est le suivant :

$$A(q^{-1})Y(t)=B(q^{-1})U(t-1)+X(t) \quad (4.1)$$

Les polynômes A et B sont définis par

$$A(q^{-1})=1+a_1q^{-1}+a_2q^{-2}+\dots+a_nq^{-n} \quad (4.2)$$

$$B(q^{-1})=b_0+b_1q^{-1}+b_2q^{-2}+\dots+b_mq^{-m} \quad (4.3)$$

avec

$$n_a = \text{degré de polynôme } A(q^{-1}) \text{ et } n_b = \text{degré de polynôme } B(q^{-1})$$

Cette relation nous donne le modèle ARMA du système.

Le retard 1 (un) est du à la discrétisation de modèle continu. Si le système possède un retard différent de zéro, les premiers éléments du polynôme  $B(q^{-1})$  sont nuls dans l'équation (4.3),  $U(t)$  est la commande ou l'entrée du procédé,  $y(t)$  est la sortie ou la variable mesurée et  $X(t)$  est le terme de perturbation qui se présente sous forme d'un modèle MA (Moyenne glissant).

$$X(t)=C(q^{-1})\zeta(t) \quad (4.4)$$

avec

$$C(q^{-1})=1+c_1q^{-1}+\dots+c_nq^{-nc} \quad (4.5)$$

Dans cette équation,  $\zeta(t)$  est une séquence de bruit blanc à moyenne nulle et à variance finie, en les combinées avec l'équation (4.1), on obtient le modèle CARMA.

$$A(q^{-1})y(t)=B(q^{-1})u(t-1)+C(q^{-1})\zeta(t) \quad (4.6)$$

Introduisons un effet intégral, en modélisant le terme de perturbation  $X(t)$  comme suit

$$X(t)=C(q^{-1})\xi(t)/\Delta \quad (4.7)$$

Ainsi on obtient le modèle CARIMA

$$A(q^{-1})Y(t)=B(q^{-1})u(t-1)+C(q^{-1})\frac{\xi(t)}{\Delta(q^{-1})} \quad (4.8)$$

où

$$\Delta(q^{-1})=1-q^{-1} \quad (4.9)$$

Pour facilité le développement de l'algorithme posons  $C(q^{-1})=1$ . [19]

### 4.2.2 Prédiction de la sortie

La commande prédictive généralisée consiste à prédire la sortie sur un horizon  $j$  supérieur au retard. Réécrivons l'équation (4.5) à l'instant  $t+j$  :

$$y(t+j)=\frac{B(q^{-1})}{A(q^{-1})}u(t+j-1)+\frac{\xi(t+j)}{A(q^{-1})\Delta(q)} \quad (4.10)$$



- $q^{-1} \frac{B(q^{-1})}{A(q^{-1})}$  : fonction de transfert relative à la sortie et à la commande.
- $\frac{1}{A(q^{-1})\Delta(q^{-1})}$  : fonction de transfert relative à la sortie et à la perturbation aléatoire  $\xi(t)$
- Le terme  $\frac{\xi(t+j)}{A(q^{-1})\Delta(q^{-1})}$  dépend de  $(\xi(t+j), \dots, \xi(t+1), \xi(t), \xi(t-1), \dots)$

Les termes  $(\xi(t), \xi(t-1), \dots)$  peuvent être calculés à partir de l'information disponible à l'instant t, c'est à dire à partir de mesures  $(y(t), y(t-1), \dots)$  et des commandes  $(u(t-1), u(t-2), \dots)$ , les termes  $(\xi(t+j), \dots, \xi(t+1))$  constituent la partie non prédictive.

Pour expliciter analytiquement cette décomposition, introduisant l'identité polynomiale suivante.

$$1 = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + F_j(q^{-1}) \tag{4.11}$$

Appelée équation DIOPHANTINE, obtenue à l'aide de l'algorithme d'EUCLIDE [20] qui permet d'effectuer la division de 1 par le polynôme  $A(q^{-1})\Delta(q^{-1})$  jusqu'à l'ordre j. par la suite nous donnons un algorithme récursif donnons les coefficients des polynômes  $E_j, F_j$ . Nous pouvons remarquer que les polynômes  $E_j, F_j$  dépend uniquement de  $A(q^{-1})$  et de l'intervalle de prédiction j tel que :

$$d^0(E(q^{-1})) = j - 1 \tag{4.12}$$

et

$$d^0(F(q^{-1})) = d^0(A(q^{-1})) \tag{4.13}$$

Si l'équation (4.10) est multipliée par  $E_j(q^{-1})A(q^{-1})\Delta(q^{-1})$ , on obtient la décomposition de l'information connu.

$$E_j(q^{-1})A(q^{-1})\Delta(q^{-1})y(t+j) = E_j(q^{-1})B(q^{-1})\Delta(q^{-1})u(t+j-1) + E_j(q^{-1})\xi(t+1) \tag{4.14}$$

En substituant  $E_j(q^{-1})A(q^{-1})\Delta(q^{-1})$  de l'équation (4.11) on obtient :

$$y(t+j) = F_j(q^{-1})B(q^{-1})\Delta(q^{-1})U(t+j-1) + F_j(q^{-1})Y(t) + F_j(q^{-1})\xi(t+j) \tag{4.15}$$

Comme  $E_j(q^{-1})$  est de degré j-1, les composantes de bruit sont toutes dans la future.

La minimisation de l'erreur de prédiction au sens de moindre carrés aboutit à la détermination du prédicteur optimal compte tenue des informations disponible à l'instant t.

$$\hat{y}(t+j/t) = G_j(q^{-1})\Delta U(t+j-1) + F_j(q^{-1})Y(t) \tag{4.16}$$

avec

$$G_j(q^{-1}) = E_j(q^{-1})B(q^{-1}) \tag{4.17}$$

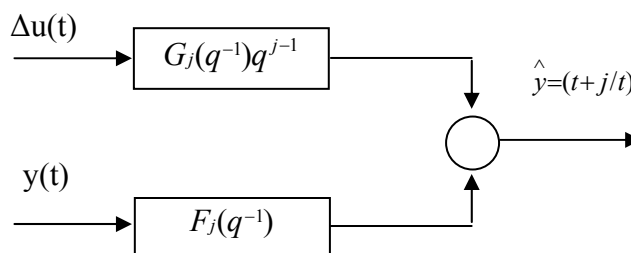


figure 4.1 structure du prédicteur optimale

Dans la commande prédictive généralisée, nous considérons un ensemble de prédiction sur  $j$  pas, où  $j$  varie dans un intervalle appelé horizon de prédiction. Le prédicteur  $y(t+k/t)$  dépend entièrement des données disponibles mais pour  $j > k$ , les hypothèses doivent être faite sur les actions de commande future.

### 4.3 Résolution itérative de l'équation de DIOPHANTINE

Comme cela a été énoncé dans la section (4.2.2), nous proposons un algorithme pour la résolution de l'équation (4.11), l'implémentation de la G.P.C nécessite la résolution de l'équation de DIOPHANTINE [20], le développement itérative de cette équation ne présente aucune difficulté du point de vue mathématique c'est une résolution itérative qui a été développée par D.W. CLARK [21].

Considérons l'équation suivante :

$$1 = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}) \quad (4.18)$$

Supposons que  $E(q^{-1})$  et  $F_j(q^{-1})$  sont connus et cherchons à calculer  $E_{j+1}(q^{-1})$  et  $F_{j+1}(q^{-1})$ , ces deux polynômes doivent vérifier l'équation (4.18) pour  $j+1$ .

$$1 = E_{j+1}(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-(j+1)}F_{j+1}(q^{-1}) \quad (4.19)$$

On obtient alors par soustraction des polynômes (4.16) et (4.18)

$$A(q^{-1})\Delta(q^{-1})(E_{j+1}(q^{-1}) - E_j(q^{-1})) + q^{-j}(q^{-1}F_{j+1}(q^{-1}) - F_j(q^{-1})) = 0 \quad (4.20)$$

tel que  $d^\circ(E_j) = j-1$ ,  $d^\circ(E_{j+1}) = j$  et  $d^\circ(E_{j+1} - E_j) = j$

Ecrivons cette différence sous la forme

$$E_{j+1}(q^{-1}) - E_j(q^{-1}) = R(q^{-1}) + r_j(q^{-j}) \quad (4.21)$$

de telle sorte que

$$A(q^{-1})\Delta(q^{-1})R(q^{-1}) + q^{-j}(q^{-1}F_{j+1}(q^{-1}) - F_j(q^{-1}) + A(q^{-1})\Delta(q^{-1})r_j) = 0 \quad (4.22)$$

pour

$$R(q^{-1}) = 0 \quad (4.23)$$

$$F_{j+1}(q^{-1}) = qF_j(q^{-1}) - A(q^{-1})\Delta(q^{-1})r_j \quad (4.24)$$

d'où les récursions qui permettent de calculer les coefficients des polynômes  $F_{j+1}(q^{-1})$  et  $E_{j+1}(q^{-1})$  connaissant les polynômes  $F_j(q^{-1})$  et  $E_j(q^{-1})$  et posons :

$$F_j(q^{-1}) = f_{j,0} + f_{j,1}q^{-1} + \dots + f_{j,na}q^{-na} \quad (4.25)$$

$$A(q^{-1})\Delta(q^{-1}) = a_0^* + a_1^*q^{-1} + \dots + a_{na+1}^*q^{-(na+1)} \quad (4.26)$$

de l'équation (4.23) et (4.24) on obtient :

$$r_j = f_{j,0} \quad (4.27)$$

$$f_{j+1,i} = f_{j,i+1} - \dot{a}_{i+1}r_j \quad (4.28)$$

Les conditions initiales étant pour  $j = 1$

$$1 = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}) \quad (4.29)$$

Le coefficient  $a_0^*$  étant égal à 1 pour que cette identité soit vérifiable, il faut que le coefficient de  $E_j(q^{-1})$  soit égal à 1, comme le degré de  $q^{-1}F_j(q^{-1})$  est égal au degré du polynôme

$A(q^{-1})\Delta(q^{-1})$ , le degré de  $E_1(q^{-1})$  doit être zéro.  
donc

$$E_1(q^{-1})=1 \quad (4.30)$$

$$E_{j+1}(q^{-1})=E_j(q^{-1})+q^{-1}r_j \quad (4.31)$$

$$F_1(q^{-1})=q(1-A(q^{-1})\Delta(q^{-1}))E_{j+1}(q^{-1})-B(q^{-1})F_{j+1}(q^{-1}) \quad (4.32)$$

#### 4.4 Loi de commande prédictive

L'objectif de la loi de commande prédictive est de conduire la sortie future du système  $y(t+j)$  à la référence  $w(t+j)$  comme c'est illustré dans la figure 4.2.

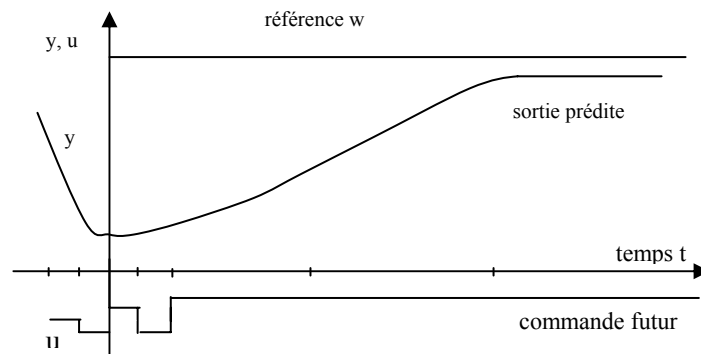


figure 4.2 référence, commande et sortie dans la G.P.C

Ceci est fait par l'utilisation de l'approche d'horizon fuyant par la quelle à chaque instant d'échantillonnage on a :

- La séquence de référence future  $w(t+j)$  est calculée.
- Le modèle de prédiction de l'équation (4.16) est utilisé pour générer un ensemble de sorties  $y(t+j/t)$  avec l'erreur de prédiction  $e(t+j)$  correspondant au système.
- Une fonction quadratique incluant des erreurs futures et des commandes, qui sont minimisées et en tenant compte qu'après un certain horizon de commande tous les incréments de commande futur sont égal à zéro.
- Seule le premier élément  $u(t)$  de la séquence est appliqué au système.
- Tous les vecteurs de données sont décalés de telle sorte que les calculs peuvent être répétés à chaque instant d'échantillonnage.

#### 4.5 Critère quadratique

La loi de commande prédictive considérée G.P.C est associée à la minimisation du critère quadratique suivant :

$$J(N_1, N_2, t) = E \left\{ \sum_{j=N_1}^{N_2} [y(t+j) - W(t+j)]^2 + \sum_{j=1}^{N_2} \lambda(j) [\Delta(q^{-1})u(t+j-1)]^2 \right\} \quad (4.33)$$

- $w(t+j)$  : une séquence de référence future initialement connue.
- $N_1$  : est l'horizon minimum de prédiction.

- $N_2$  : est l'horizon maximum de prédiction.
- $\lambda(j)$  : scalaire positif agissant sur les incréments de commande futur.

#### 4.6 Calcul du prédiction a j-pas

L'expression de la sortie à l'instant  $t+j$  s'écrit :

$$\begin{cases} y(t+1)=G_1(q^{-1})\Delta u(t)+F_1(q^{-1})y(t)+E_1(q^{-1})\zeta(t+1) \\ y(t+2)=G_2(q^{-1})\Delta u(t+1)+F_2(q^{-1})y(t)+E_2(q^{-1})\zeta(t+2) \\ \vdots \\ y(t+N_2)=G_{N_2}(q^{-1})\Delta u(t+1)+F_j(q^{-1})y(t)+E_j(q^{-1})\zeta(t+N_2) \end{cases} \quad (4.34)$$

Le terme  $y(t+j)$  est constituer de trois parties :

- Le premier dépend des actions de commande future (terme à déterminer).
- Le deuxième dépend des commandes connues.
- Le troisième dépend des signaux de bruit futures (terme imprédictible).

Le prédicteur à  $j$ -pas de l'équation (4.8) peut être écrit sous forme matricielle et cela dans le but de :

- Faire une prédiction en terme d'incrément de commande.
- Décomposer la sortie prédite en deux termes.
  - un dépend des commandes futures.
  - l'autre ne comportant que des signaux connus à l'instant  $t$ .

soit

$$\hat{y}(t+j/t)=\hat{y}_1(t+j)+\hat{y}_2(t+j/t) \quad (4.35)$$

nous avons

$$\hat{y}(t+j/t)=G_j(q^{-1})\Delta u(t+j-1)+F_j(q^{-1})y(t) \quad (4.36)$$

en effet

$$E_j(q^{-1})=\frac{1-q^{-1}F_j(q^{-1})}{A(q^{-1})\Delta(q^{-1})} \quad (4.37)$$

où

$$G_j(q^{-1})=B(q^{-1})E_j(q^{-1}) \quad (4.38)$$

donc

$$G_j(q^{-1})=B(q^{-1})\frac{1-q^{-1}F_j(q^{-1})}{A(q^{-1})\Delta(q^{-1})} \quad (4.39)$$

Le polynôme  $G_j(q^{-1})$  peut être décomposé en deux parties

$$G_j(q^{-1})=\tilde{G}_j(q^{-1})+q^{-j}\bar{G}_j(q^{-1}) \quad (4.40)$$

avec

$$\tilde{G}_j(q^{-1})=g_{j,0}+g_{j,1}q^{-1}+\dots+g_{j,j-1}q^{-(j-1)} \quad (4.41)$$

$$\bar{G}_j(q^{-1})=g_{j,j}+\dots+\dots+g_{j,nb-1}q^{-(nb-1)} \quad (4.42)$$

donc

$$\hat{y}(t+j/j)=\tilde{G}_j(q^{-1})\Delta(q^{-1})u(t+j-1)+\bar{G}_j(q^{-1})\Delta(q^{-1})u(t-1)+F_j(q^{-1})y(t) \quad (4.43)$$

En identifiant l'équation (4.35) et (4.43) nous aurons :

$\hat{y}_1(t+j)=\tilde{G}_j(q^{-1})\Delta u(t+j-1)$  est la fonction de la valeur présente et des valeurs futures des incréments ( $\Delta u(t+i)$   $i>0$ ).

$\hat{y}_2(t+j/t)=\bar{G}_j(q^{-1})\Delta U(t-1)+F_j(q^{-1})y(t)$  est la fonction des données disponibles à l'instant t.

Considérons maintenant la séquence  $(f(t+1), f(t+2), \dots)$ , qui est en fonction des données disponibles à l'instant t mais, seulement il faut réécrire  $\bar{G}_j=q^j(G_j-\tilde{G}_j)$ , ce qui nous permet d'écrire le prédicteur à j-pas sous la forme matricielle :

$$\begin{bmatrix} y(t+1/t) \\ y(t+2/t) \\ \vdots \\ y(t+n/t) \end{bmatrix} = \begin{bmatrix} g_{10} & \cdot & \cdot & \cdot & \cdot & 0 \\ g_{21} & g_{20} & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{j-1} & g_{j-2} & \cdot & \cdot & \cdot & g_{j0} \end{bmatrix} \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+j-1) \end{bmatrix} + \begin{bmatrix} (G_1 - g_{10}) + F_1 y(t) \\ q(G_2 - g_{20} - g_{10} q^{-1})\Delta u(t) + F_2 y(t) \\ \vdots \\ q^{N_2-1}(G_{N_2} - g_{j0} - g_{10} q^{-1} - \dots - g_{j-1} q^{j-1})\Delta u(t) + F_j y(t) \end{bmatrix} \quad (4.44)$$

sous une autre forme:

$$\hat{Y} = G\tilde{U} + f \quad (4.45)$$

Où tous les vecteurs sont de dimension  $(N_2 \times 1)$  et  $g_{i,j} = g_i$  pour  $j = 1, 2, 3, \dots < i$

$$\hat{Y} = [\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+N_2)]^T \quad (4.46)$$

$$U = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_2-1)]^T \quad (4.47)$$

$$f = [f(t), f(t+1), \dots, f(t+N_2)]^T \quad (4.48)$$

la matrice G est triangulaire de dimension  $(N_2 \times N_2)$

$$G = \begin{bmatrix} g_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_1 & g_0 & \cdot & \cdot & \cdot & 0 & 0 \\ g_2 & g_1 & g_0 & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & g_0 & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & g_0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & g_0 & 0 \\ g_{N_2-1} & g_{N_2-2} & \cdot & \cdot & \cdot & \cdot & g_0 \end{bmatrix} \quad (4.49)$$

### 4.7 Minimisation du critère

La commande est élaborée à partir de la minimisation d'un critère quadratique multipas basé sur l'écart entre la sortie prédite et la référence [20], Le critère de l'équation (4.33) s'écrit sous la forme vectoriel suivante :

$$J = E\{(Y-W)^T(Y-W) + \lambda \tilde{U}^T \tilde{U}\} \quad (4.50)$$

avec

$$W = [W(t+1), W(t+2), \dots, W(t+N_2)]^T \quad (4.51)$$

donc

$$J = (G\tilde{U} + f - W)^T(G\tilde{U} + f - W) + \lambda \tilde{U}^T \tilde{U} \quad (4.52)$$

$$\frac{\partial J}{\partial \bar{U}}=0 \quad (4.53)$$

$$\frac{\partial}{\partial \Delta U}(G\Delta U+f-W)^T(G\Delta U+f-W)+\frac{\partial}{\partial \Delta U}(G\Delta U+f-W)(G\Delta U+f-W)^T+2\lambda\Delta U=0 \quad (4.54)$$

$$\Rightarrow G^T(G\Delta U+f-W)+G(G^T\Delta U+f-W)^T+2\lambda\Delta U=0 \quad (4.55)$$

$$\Rightarrow (G^TG+\lambda I)\Delta U+G^T(f-W)=0 \quad (4.56)$$

$$\Delta U=(G^TG+\lambda I)^{-1}G^T(W-f)=0 \quad (4.57)$$

La première commande est :

$$U(t)=U(t-1)+\bar{g}^T(W-f) \quad (4.58)$$

où  $\bar{g}^T$  est la première ligne de  $(G^TG+\lambda I)^{-1}G^T$ .

La loi est utilisée dans le sens de l'horizon fuyant est que le premier élément  $\Delta U(t)$  est calculé et exécuté pour donner la commande  $U(t)=U(t-1)+\Delta U(t)$ , et à l'instant  $t+1$  tous le calcul est répété pour avoir  $\Delta U(t+1)$ .

#### 4.7 Introduction des horizons $N_1, N_2, N_u$

La matrice  $G$  de l'équation (4.18) est de dimension  $(N_2 \times N_2)$  basée sur l'hypothèse faite à propos des actions de commande future, c'est-à-dire après l'horizon de commande  $N_u$  tous les incréments de commande sont pris égaux à zéro soit :

$$\Delta U(t+j-1)=0 \quad \text{pour } j > N_u \quad (4.59)$$

la matrice  $G$  devient de dimension  $(N_2 \times N_u)$ , et donc la matrice à inverser sera de dimension  $(N_u \times N_u)$ .

Si dans le cas où  $N_u$  est égale à un ( $N_u=1$ ), la matrice  $(G^TG+\lambda I)^{-1}$  se réduit à un scalaire dont l'inversion ne pose aucun problème, par l'introduction de l'horizon initial de prédiction  $N_1$  la matrice  $G$  se réduit au dimension  $(N_1-N_2+1) \times N_u$ , ce qui se traduit par une troncature de la matrice  $G$  du haut (ligne) par l'introduction de l'horizon initial de prédiction et à gauche (colonne) par l'horizon de commande  $N_u$  [21].

la matrice  $G$  se présente sous la forme suivante

$$G_2 = \begin{bmatrix} g_{N_1-1} & g_{N_1-2} & \dots & g_0 & 0 & 0 & 0 & 0 & 0 \\ g_{N_1} & g_{N_1-1} & \dots & \dots & g_0 & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_{N_2-1} & g_{N_2-2} & \dots & \dots & \dots & \dots & \dots & \dots & g_{N_2-N_u} \end{bmatrix} \quad (4.60)$$

## 4.8 Conclusion

Ce chapitre a été consacré au développement théorique de l'algorithme de commande prédictive généralisée GPC de base.

Le concept-clé de la commande prédictive réside dans la création d'un effet anticipatif. On exploite pour cela les connaissances explicites sur l'évolution de la trajectoire à suivre dans le futur (connaissances nécessaires requises au moins sur un horizon de quelques points au delà de l'instant présent), la commande prédictive s'avère une structure suffisamment complète pour résoudre un problème très général.

Les avantages de la commande prédictive généralisée se déduisent directement des points précédents; tout d'abord, l'utilisation de modèle CARIMA et le travail sur les incréments de commande assure la précision statique.

Ensuite, on a vu qu'il n'était pas nécessaire d'utiliser une représentation d'état lors de l'implantation de modèle du système, ce qui s'avère intéressant lorsque les modèles proviennent d'une identification directe sous forme de fonction de transfert discrète.

Enfin, la relation fournissant la commande se traduit par un algorithme simple, où beaucoup de calculs peuvent être faits hors ligne, où il ne reste à faire en temps réel que l'acquisition de la sortie du processus et le calcul de la commande, incluent néanmoins l'élaboration de la réponse libre.

Cette relative simplicité permet à la méthode d'avoir des applications dans de très nombreux domaines, et notamment en électromécanique et la robotique où les constantes de temps très petites des processus utilisés nécessitent l'emploi d'algorithmes rapides et peut gourmandes en temps de calcul.

En contrepartie de tous ces avantages, l'algorithme de GPC développé possède deux inconvénients, tout d'abord, il n'existe aucune théorie précise permettant de démontrer la stabilité de la commande, à l'exception de travaux de CLARK [22] assurant la robustesse de la méthode sous certaines conditions de réglage très restrictives.

# CHAPITRE V

**Application de la commande GPC au robot  
manipulateur PUMA**



## 5.1 Introduction

Durant ces dernières années une attention particulière a été accordée aux unités de production flexibles dont les performances prépondérantes sont la qualité, le coût et le temps de production, ces unités requièrent en particulier la rapidité d'exécution des tâches dans un environnement restreint avec des interactions mécaniques qui peuvent avoir lieu dans l'espace de travail, les robots industriels sont en grande majorité des manipulateurs et sont souvent utilisés pour répondre à ces spécifications par programmation des tâches bien spécifiques. Les unités de production nécessitent des lois de commande rapides et robustes vis-à-vis des incertitudes et des variations de la charge durant l'exécution des tâches.

La commande prédictive permet d'atteindre, dans bien des cas, un excellent compromis performance/robustesse. Pour bénéficier de cette potentialité dans la commande non linéaire des robots manipulateurs, les lois de commande prédictives à suivi de trajectoire développées dans le chapitre précédent seront appliquées à un robot manipulateur. L'objectif de commande est la poursuite d'une trajectoire prédéfinie et sa robustesse vis-à-vis des variations de la charge et des frottements secs et visqueux.

Dans cette partie de notre étude, nous appliquons cette technique à la commande d'un robot manipulateur [23] qui représente un système non linéaire et multi-variable dans le but de faire une comparaison avec la commande CDM. Rappelons que, soit la CDM ou GPC sont régies par la même structure de commande RST.

## 5.2 Modèle de représentation

Le modèle obtenu dans le chapitre 1 est un modèle multi-variable à deux entrées et deux sorties couplés. Il se présente sous la forme d'état suivante

$$\begin{cases} \dot{X} = f(X, U, t) \\ Y = h(X) \end{cases} \quad (5.1)$$

X : vecteur d'état constitué de la position, de la vitesse de dimension 2.

Y : vecteur de sortie de dimension 2

U : vecteur d'entrée ou de commande de dimension 2

t : temps.

$$f^T(X, U, t) = [f_1 \quad f_m] \quad (m=2) \quad (5.2)$$

$$h^T(X) = [h_1(X) \quad h_m(X)] \quad (m=2) \quad (5.3)$$

La commande par ordinateur de ce type de système nécessite la connaissance de modèle de la représentation discrète échantillonnée à une période d'échantillonnage  $T_e$  adéquatement choisie. Pour cela la discrétisation de l'équation (5.1) après la linéarisation exacte de robot est nécessaire, après des manipulations mathématiques aboutissant au modèle paramétrique discret suivant :

$$\begin{bmatrix} A_1(q^{-1}) & 0 \\ 0 & A_2(q^{-1}) \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} B_1(q^{-1}) & 0 \\ 0 & B_2(q^{-1}) \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} + \begin{bmatrix} C_1(q^{-1}) & 0 \\ 0 & C_2(q^{-1}) \end{bmatrix} \begin{bmatrix} \zeta_1(q^{-1}) \\ \zeta_2(q^{-1}) \end{bmatrix} \quad (5.4)$$

## 5.2 Résultats de simulation

Dans cette partie on applique l'algorithme de base au modèle de robot PUMA. Ce modèle a été simulé avec un pas de calcul de  $dt=10^{-3}s$  en utilisant la méthode de R-K du quatrième ordre, tandis que l'algorithme de commande prédictive généralisée a été appliqué avec une période d'échantillonnage de  $Te=10\text{ ms}$ .

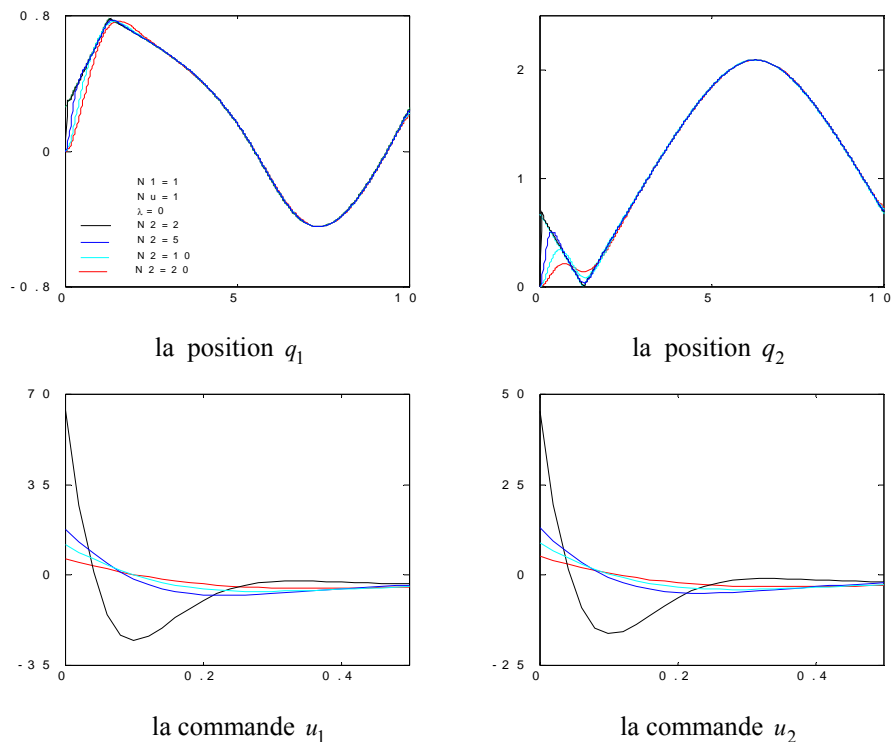
L'étude par simulation nous montrera la puissance et la capacité de l'algorithme de commande qui donne une bonne poursuite des trajectoires. Tous les résultats sont présentés sur des figures dont chacune comporte, les sorties et consignes, les commandes la trajectoire parcourue et l'erreur de trajectoire pour chaque variable généralisée ( $q_1, q_2$ ).

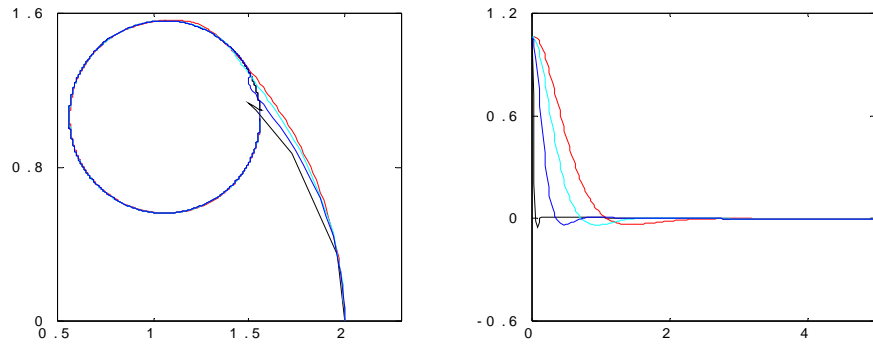
### Remarque

- Tous les résultats ont été réalisés pour  $N_1=N_2=1$ .
- $N_1[i], N_2[i], N_u[i], \lambda_i$  pour  $i=1,2$  : représentent les paramètres de synthèse de la commande.
- $N_1[i]$  : est l'horizon initial de prédiction.
- $N_2[i]$  : est l'horizon maximal de prédiction.
- $N_u[i]$  : est l'horizon de prédiction.
- $\lambda_i$  : est la pondération de la commande.

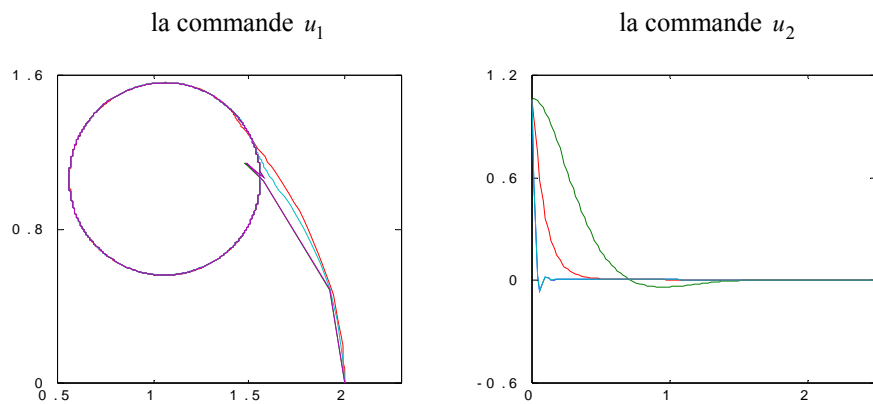
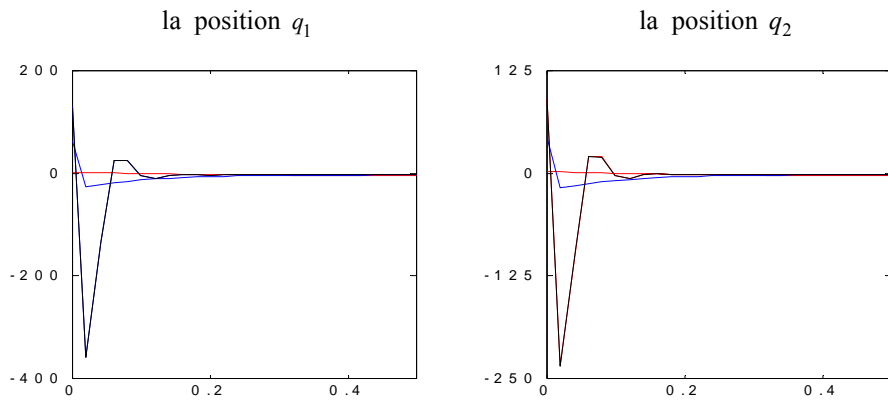
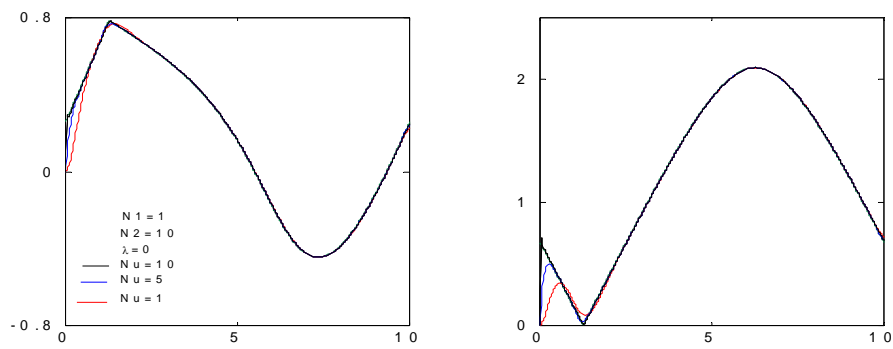
A partir du choix des paramètres de synthèse de GPC, les résultats de simulation sur le robot PUMA pour une trajectoire circulaire sont présentés.

La simulation a été réalisée sur 10 s, le temps que le robot effectue un tour complet, Les figures (5.1), (5.2) et (5.3) montrent l'évolution des variables généralisées  $q_1$  et  $q_2$  de la commandes et de trajectoire parcourus respectivement pour une variation de  $N_2$ , variation de  $N_u$  et variation de  $\lambda$ , et pour des conditions initiales nulle. La même simulation a été réalisée pour les paramètres de synthèse  $N_1[i]=1, N_2[i]=5, N_u[i]=1$  et  $\lambda_i = 0$  et application d'une perturbation, figure 5.4 ensuite une variation paramétrique, figure 5.5.





la trajectoire parcourue l'erreur de trajectoire  
 figure 5.1 positions  $q_1$  et  $q_2$  commandes  $U_1$  et  $U_2$  pour  $N_1[i]=1, N_u[i]=1, \lambda_i=0$   
 et variation de  $N_2$



la trajectoire parcourue l'erreur de trajectoire  
 figure 5.2 positions  $q_1$  et  $q_2$  commandes  $U_1$  et  $U_2$  pour  $N_1[i]=1, N_2[i]=1, \lambda_i=0$   
 et variation de  $N_u$

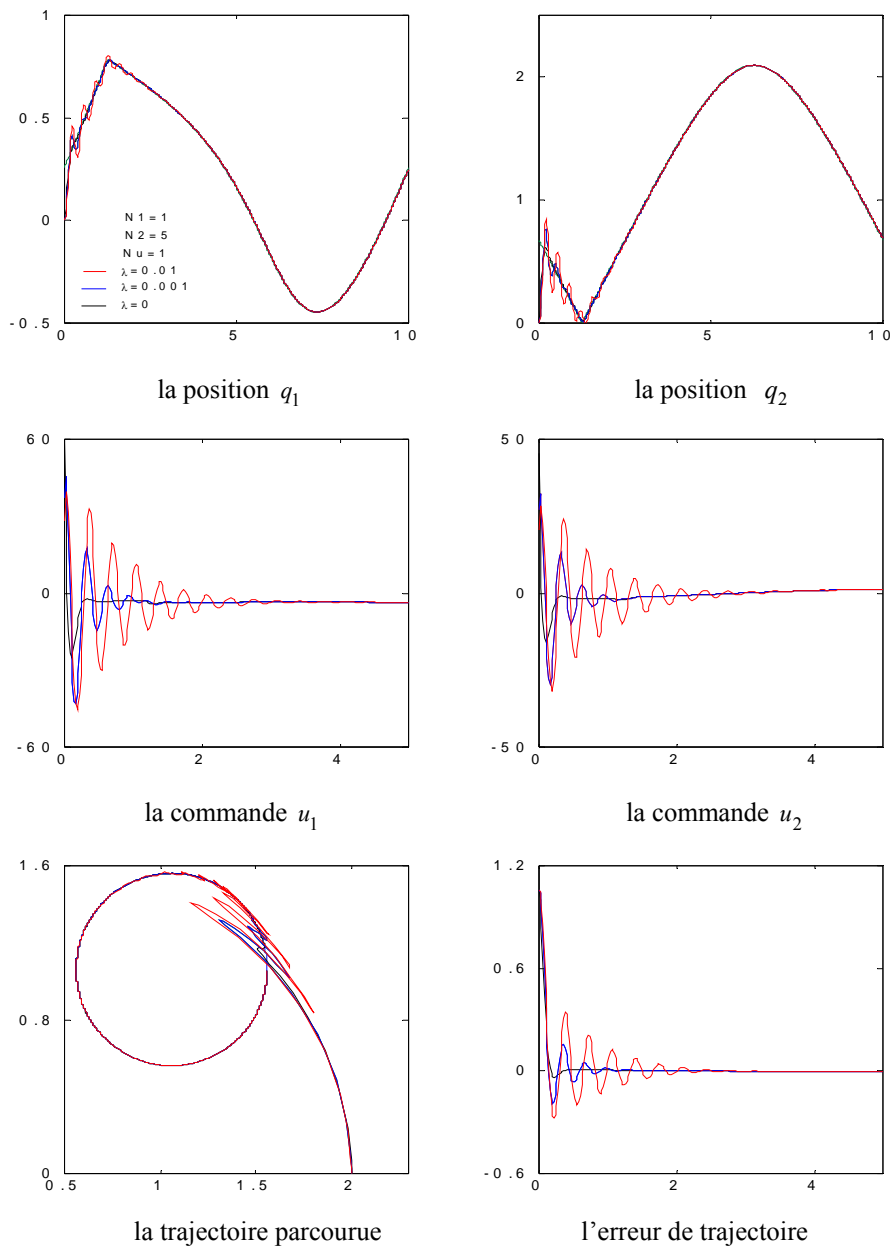
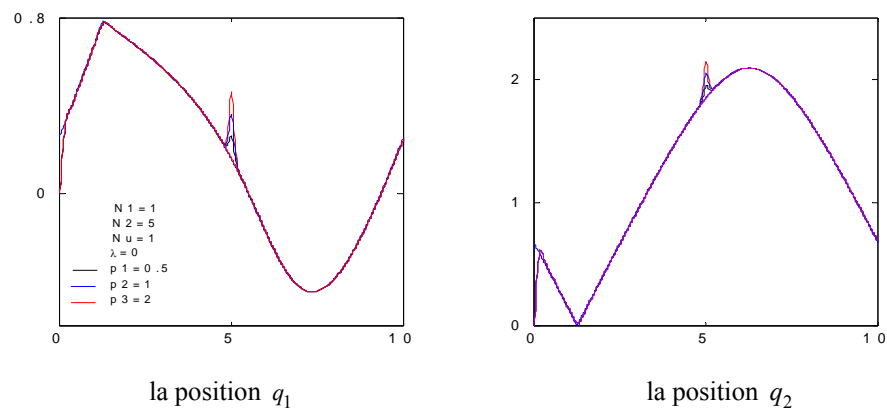


figure 5.3 positions  $q_1$  et  $q_2$  commandes  $U_1$  et  $U_2$  pour  $N_1[i]=1, N_2[i]=5, N_u[i]=1$ , et variation de  $\lambda_i$



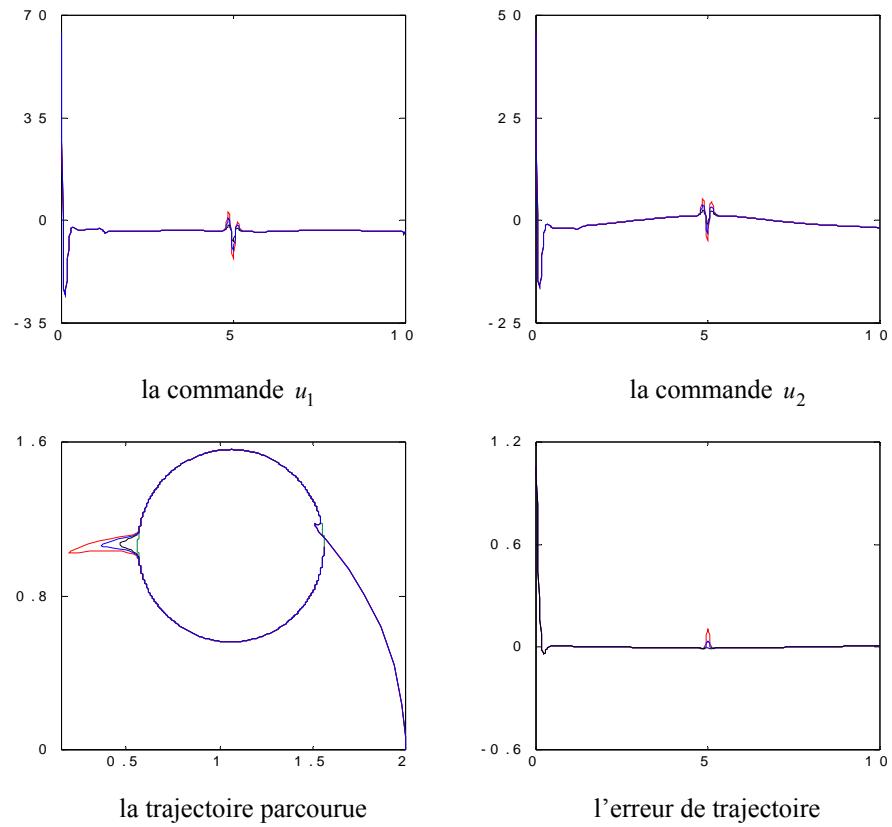
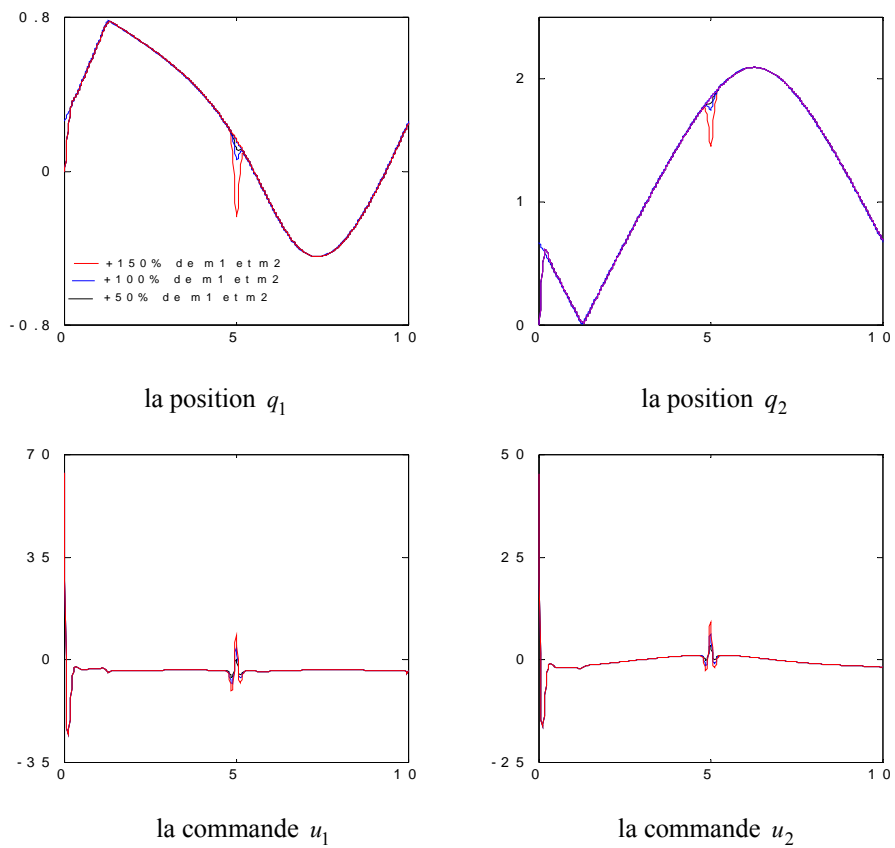


figure 5.4 positions  $q_1$  et  $q_2$  commandes  $U_1$  et  $U_2$  pour  $N_1[i]=1, N_2[i]=5, N_u[i]=1, \lambda_i = 0$  et application de perturbation



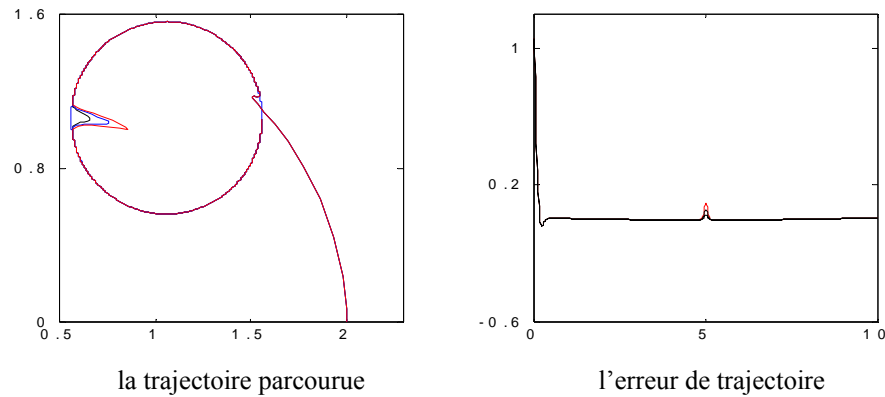


figure 5.5 positions  $q_1$  et  $q_2$  commandes  $U_1$  et  $U_2$  pour  $N_1[i]=1$ ,  $N_2[i]=5$ ,  $N_u[i]=1$ ,  $\lambda_i=0$  et variation paramétrique.

### 5.3 Synthèse et discussion

L'étude de l'influence des paramètres de la GPC sur le robot manipulateur PUMA nous a montré l'efficacité de la GPC. Son efficacité réside dans le choix des paramètres de synthèse  $\lambda$ ,  $N_1$ ,  $N_2$  et  $N_u$  qui demeure délicat.

Le choix de ces horizons et du coefficient de pondération à un aspect fondamental dans le succès de la GPC, plusieurs auteurs ont traités ce problème, ce qui nous amène à faire une discussion sur chaque paramètre.

- **Horizon initial de prédiction  $N_1$**

Le paramètre  $N_1$  est toujours déterminé par  $N_1 T_e \geq \text{retard}$  ( $T_e$  : période d'échantillonnage du système). En effet, si le retard du système est connu il n'y a pas de raisons à imposer sur  $N_1$  une valeur inférieure au retard, autrement des calculs inutile seront effectués.

donc

$$N_1 = \frac{\text{retard de système}}{\text{période d'échantillonnage}}$$

- **Horizon maximal de prédiction  $N_2$**

Généralement  $N_2$  doit être supérieur au degré de  $B$ , ce qui est évident relativement au choix de  $N_1$ . En effet, si  $N_1$  est supérieur au retard du système qui lui même en rapport avec le degré de  $B$ , ce qui vous dire que  $N_2$  est choisi supérieur ou égale à  $N_1$ .

Dans la commande des systèmes, on privilégie le régime établi. En effet, le régime peut être atteint par augmentation de  $N_2$  avec des commandes douces, mais au détriment du temps de réponse. En règle générale  $N_2$  est choisie de façon qu'il correspond au temps de montée du système.

donc

$$N_2 = \frac{\text{temps de monté}}{\text{période d'échantillonnage}}$$

- **Horizon de commande  $N_u$**

Pour les différents systèmes étudiés, on a constaté que la valeur de  $N_u=1$  a donnée de bons résultats, l'augmentation de  $N_u$  ne fait que rendre la commande plus active et importante, une

autre augmentation ne fera presque aucune différence. Cette augmentation introduit une énorme complexité de calcul d'où un temps de calcul très important. Dans le cas de la commande numérique à paramètre connus, ce temps de calcul ne nous intéresse pas puisqu'ils se feront hors ligne.

- **Coefficient de pondération de la commande  $\lambda$**

L'atténuation des variations brusques de commande que l'on peut constater avec cet algorithme de GPC est possible par l'introduction de paramètre  $\lambda$ . Il permet de donner plus ou moins de poids sur la commande de façon à assuré de bonne réponse lorsque le système au départ présente un risque d'instabilité.

Le coefficient  $\lambda$  ne peut être lié qu'au gain du système a réglé, parce que plus le gain d'un système est grand plus la commande doit être pondérée, c'est à dire plus  $\lambda$  est important et inversement. Donc  $\lambda$ . doit être choisi d'une manière a amplifié ou a réduire l'effort de commande

## 5.4 Comparaison entre la commande CDM et GPC

D'après les résultats de simulation on constate que le temps de réponse  $t_s$  est imposé initialement par l'algorithme de commande CDM, tandis que le temps de réponse de système commandé par GPC dépend de tous les paramètres de synthèse du GPC. Le paramètre  $\lambda$  de GPC est semblable a celui  $\gamma$  de CDM lorsque il est inférieur à la limite de stabilité  $\gamma^*$  et  $\lambda$  loin de zéro, mais pour les valeurs de  $\gamma$  supérieur de  $\gamma$  standard, la valeur de  $\lambda$  correspondante est la valeur proche de zéro.

Pour la commande GPC la détermination de la loi de commande est basée sur la minimisation d'un critère quadratique, mais pour la CDM, les performances sont imposées par l'algorithme de commande. Pour la mise en œuvre de l'algorithme de GPC, il faut disposer d'un calculateur numérique, cependant pour la CDM, il faut avoir seulement un circuit analogique. Donc, la mise en œuvre de l'algorithme de la commande CDM est mois coûteuse que celle de la GPC.

Les résultats de simulation montre que les deux méthodes sont capables de rejeter les perturbations et les variations paramétriques qui influent sur le robot manipulateur PUMA.

## 5.5 Conclusion

Ce chapitre a été consacré à l'application de la GPC au robot manipulateur PUMA. Il est claire, d'après les résultats obtenus, qu'une valeur supérieure ou égale à cinq de l'horizon de prédiction est nécessaire, Nous tenons à signaler qu'un compromis existe entre l'horizon de commande et la commande. En effet, l'augmentation de  $N_u$  rend les commande plus actives, Les coefficients de pondération supérieur à un ou égal ou légèrement inférieur à un rendent les commandes insuffisantes à subvenir au besoin des robots pour la poursuite de trajectoire, d'où la nécessité d'assigner à  $\lambda_i$  des valeurs très petites par rapport à un ou 'carrément' égale à zéro.

Enfin on peut conclure q'une bonne poursuite de trajectoires désirées pour le Robot manipulateur PUMA ne peut être réalisée que par un choix adéquat des paramètres de synthèse de la GPC.

*CONCLUSION*  
*GENERALE*



## **Conclusion générale**

Dans ce travail, on s'est intéressé à l'étude des performances de la méthode de diagramme des coefficients, la simulation de quatre types de système à dynamique difficile, présentant chacun une particularité spécifique, Ceci grâce aux paramètres de synthèse de la commande CDM, qui constitue la puissance de l'algorithme.

L'utilisation de la forme standard de CDM nous à permet de choisir les paramètre de commande, Ceci veut dire qu'on peut spécifier d'avance une dynamique de sortie vu qui' il y'a un lien concret entre les paramètres de synthèse et l'équation caractéristique.

L'application de la CDM au robot manipulateur PUMA a montré que malgré la dynamique imposée par le modèle de robot en boucle ouverte, de très bons résultats ont été obtenus et ceci par un choix spécifique des paramètres de synthèse.

L'étude et l'application de GPC au robot manipulateur a donnée des performances acceptable par le choix de paramètre de synthèse, un autre paramètre paraît très important pour l'obtention d'une bonne poursuite de trajectoire, mais il faut prendre une attention particulière doit être donné au choix des paramètres de synthèse puisque' ils représentent la puissance de GPC.

La recherche de ces paramètres est une tache pénible, l'utilisation d'un algorithme se base sur un critère optimale, permet de trouver systématiquement les paramètres optimaux, la spécification du critère et la méthode d'optimisation offre de nouvelle voies pour la synthèse d'un algorithme de commande GPC, dont l'utilisation spécifie un double critères celui de la commande et un autre pour les paramètres de synthèse.

*Annexe*

## ANNEXE I

## Calcul de modèle dynamique de robot par la méthode d'E-L

Les matrices de transformations sont :

$$A_1^0 = \begin{bmatrix} c1 & -s1 & 0 & l1c1 \\ s1 & c1 & 0 & l1s1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2^1 = \begin{bmatrix} c2 & -s2 & 0 & l2c2 \\ s2 & c2 & 0 & l2s2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_0^2 = \begin{bmatrix} c12 & -s12 & 0 & l(c1+c12) \\ s12 & c12 & 0 & l(s1+s12) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{ann1.1})$$

avec

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{ann1.2})$$

On obtient les matrices  $U_{ij}$  en utilisons :

$$U_{ij} = \frac{\partial A_i^0}{\partial q_j} = \begin{cases} 0 & \text{si } j \geq i \\ A_{j-1}^0 Q_j A_i^{j-1} & \text{si } j \leq i \end{cases} \quad (\text{ann1.3})$$

On obtient les matrices  $U_{ij}$

$$U_{11} = A_0^1 Q_1 A_1^0 = \begin{bmatrix} -s1 & -c1 & 0 & -l1s1 \\ -c1 & -s1 & 0 & l1c1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad U_{21} = Q_1 A_2^1 = \begin{bmatrix} -s12 & -c12 & 0 & -l(s12-s1) \\ -c12 & -s12 & 0 & l(c12+c1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad U_{22} = A_0^2 Q_2 A_0^2 = \begin{bmatrix} -s12 & -c12 & 0 & -l1s12 \\ -c12 & -s12 & 0 & l1c12 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{ann1.4})$$

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial (\dot{q}_i / dt)} \right] - \frac{\partial L}{\partial q_i} = \tau_i \quad (\text{ann1.5})$$

Avec L est le Lagrangien.

$$\tau_i = \sum_{j=i}^n \sum_{k=1}^i \text{tr}(U_{ik} \cdot J_j \cdot U_{jk}^T) \cdot \frac{d^2 q_k}{dt^2} + \sum_{j=i}^n \sum_{k=1}^i \sum_{m=1}^j \text{tr}(U_{jkm} \cdot J_j \cdot U_{ji}^T) \dot{q}_k \cdot \dot{q}_m - \sum_{j=i}^n m_j g \cdot U_{ji} \cdot r_j^j \quad (\text{ann1.6})$$

On mettait

$$\tau_i = \sum_{j=i}^2 D_{ik} \ddot{q}_k + \sum_{k=1}^2 \sum_{m=1}^2 h_{ikm} \dot{q}_k \dot{q}_m + C_i \quad (\text{ann1.7})$$

Sachant que

$$U_{ijk} = \begin{cases} A_0^{k-1} Q_k A_{k-1}^{j-1} Q_j A_{j-1}^i & ; k \leq j \leq i \\ A_0^{j-1} Q_j A_{j-1}^{k-1} Q_k A_{k-1}^i & ; j \leq k \leq i \\ 0 & ; i < j \text{ ou } i < k \end{cases} \quad (\text{ann1.8})$$

On obtient

$$U_{212}=U_{221}=Q_1 A_0^1 Q_2 A_1^2 = \begin{bmatrix} -c_{12} & s_{12} & 0 & -l_{c12} \\ -s_{12} & -c_{12} & 0 & -l_{s12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; U_{211}=Q_1 A_0^2 = \begin{bmatrix} -c_{12} & s_{12} & 0 & -l(c_{12}+c_1) \\ -s_{12} & -c_{12} & 0 & -l(s_{12}+s_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{ann1.9})$$

$$U_{222}=A_0^1 Q_2 A_1^1 Q_2 A_1^2 = U_{212} \quad (\text{ann1.10})$$

Les matrices d'inertie sont donnée par :

$$J_i = \begin{bmatrix} I_i & -1 & 0 & \bar{x}_{im_i} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \bar{x}_{im_i} & 0 & 0 & m_i \end{bmatrix} \quad (\text{ann1.11})$$

On obtient finalement et après quelque calculs le modèle dynamique du robot manipulateur :

$$\tau(t) \equiv D(\theta) \cdot \ddot{\theta} + h(\theta, \dot{\theta}) + c(\theta) \quad (\text{ann1.12})$$

$$\begin{bmatrix} \tau_1(\theta) \\ \tau_2(\theta) \end{bmatrix} = \begin{bmatrix} \frac{1}{3}m_1 l^2 + \frac{4}{3}m_2 l^2 + m_2 l^2 c_2 & \frac{1}{3}m_2 l^2 + \frac{1}{2}m_2 l^2 c_2 \\ \frac{1}{3}m_2 l^2 + \frac{1}{2}m_2 l^2 c_2 & \frac{1}{3}m_2 l^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \quad (\text{ann1.13})$$

$$+ \begin{bmatrix} -\frac{1}{2}m_2 l^2 s_2 \dot{\theta}_2^2 - m_2 l^2 s_2 \dot{\theta}_1 \dot{\theta}_2 \\ \frac{1}{2}m_2 l^2 s_2 \dot{\theta}_1^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}m_1 g l c_1 + m_2 g l (\frac{1}{2}c_{12} + c_1) \\ \frac{1}{2}m_2 g l c_{12} \end{bmatrix}$$

## ANNEXE 2

## Analyse des performances de la commande CDM

La forme générale de polynôme caractéristique de système en boucle fermée est comme suit :

$$P(s) = a_n s^n + \dots + a_1 s + a_0 = \sum_{i=0}^n a_i s^i \quad (\text{ann2.1})$$

Les indices de stabilité  $\gamma_i$  la constante de temps équivalent et les limites de stabilité  $\gamma_i$  sont exprimées en forme de l'équation (ann2.1)

$$\gamma_i = \frac{a_i}{a_{i+1} a_{i-1}} \quad i=1, \dots, n-1, \quad \gamma_0 = \gamma_n = \infty \quad (\text{ann2.3})$$

$$\tau = \frac{a_1}{a_0} \quad (\text{ann2.4})$$

$$\gamma_i^* = \frac{1}{\gamma_{i-1}} + \frac{1}{\gamma_{i+1}} \quad (\text{ann2.5})$$

La relation entre ces paramètres est défini par l'équation (ann2.6) et (ann2.7)

$$\frac{a_{i+1}}{a_i} = \frac{a_j}{a_{j-1}} \frac{1}{\gamma_i \gamma_{i-1} \gamma_{j+1} \gamma_j} \quad i > j \quad (\text{ann2.6})$$

$$a_i = \frac{a_0 \tau^i}{\gamma_{i-1} \gamma_{i-2}^2 \dots \gamma_1^{i-1}} \quad (\text{ann2.7})$$

Le polynôme caractéristique donné par l'équation (ann2.1) doit être exprimé en fonction des paramètres  $a_0, \tau$  et  $\gamma_i$  comme suit :

$$P(s) = a_0 \left[ \sum_{i=2}^n \left( \prod_{j=1}^{i-1} \frac{1}{\gamma_{i-j}^j} \right) (\tau s)^i \right] + \tau s + 1 \quad (\text{ann2.8})$$

Le constant de temps équivalent de l'ordre  $i$  dépend de l'ordre zéro comme suit :

$$\tau_i = \frac{a_{i+1}}{a_i} = \frac{\tau}{\gamma_i \dots \gamma_2 \gamma_1} \quad (\text{ann2.9})$$

Les indices de stabilité de quatrième ordre sont donnés par rapport de première ordre est comme suit :

$$\gamma_{ij} = \frac{a_i^2}{a_{i+j} a_{i-j}} = \prod_{k=1}^{j-1} (\gamma_{i+j-k} \gamma_{i-j+k})^k \gamma_i^j \quad (\text{ann2.10})$$

## ANNEXE 3

## LA Résolution Itérative de L'équation de DIOPHANTINE

L'implémentation de la commande prédictive généralisée nécessite la résolution de l'équation de DIOPHANTINE, sa résolution ne présente aucune difficulté du point de vue mathématique.

Nous allons présenter une méthode itérative qui a été développée récemment par CLARCK.

Considérons l'équation suivante :

$$1 = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}) \quad (\text{ann3.1})$$

Supposons que  $E_j(q^{-1})$  et  $F_j(q^{-1})$  sont connus et cherchons à calculer  $E_{j+1}(q^{-1})$  et  $F_{j+1}(q^{-1})$

$E_{j+1}(q^{-1})$  et  $F_{j+1}(q^{-1})$  doivent vérifier l'équation (ann4.1) pour  $j+1$

$$1 = E_{j+1}(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j-1}F_{j+1}(q^{-1}) \quad (\text{ann3.2})$$

Soustrayons l'équation (ann3.1) de l'équation (ann3.2)

$$0 = A(q^{-1})\Delta(q^{-1})(E_{j+1}(q^{-1}) - E_j(q^{-1})) + q^{-j}(q^{-1}F_{j+1}(q^{-1}) - F_j(q^{-1})) \quad (\text{ann3.3})$$

$E_j(q^{-1})$  et  $F_j(q^{-1})$  étant respectivement de degrés  $j-1$  et  $j$ , la différence  $E_{j+1}(q^{-1}) - E_j(q^{-1})$  est de degré  $j$ , Ecrivons cette différence sous la forme :

$$E_{j+1}(q^{-1}) - E_j(q^{-1}) = R(q^{-1}) + r_j q^{-j} \quad (\text{ann3.4})$$

L'équation (ann4.4) s'écrira :

$$0 = A(q^{-1})\Delta(q^{-1})R(q^{-1}) + q^{-j}(q^{-1}F_{j+1}(q^{-1}) - F_j(q^{-1}) + A(q^{-1})\Delta(q^{-1})r_j) \quad (\text{ann3.5})$$

Pour  $R(q^{-1})=0$  et  $F_{j+1}(q^{-1})=q(F_j(q^{-1}) + A(q^{-1})\Delta(q^{-1})r_j)$ , l'équation (ann4.5) se trouve vérifiée, d'où les récursions qui permettent de calculer les coefficients des polynômes  $E_{j+1}(q^{-1})$  et  $F_{j+1}(q^{-1})$ .

Connaissions les polynômes  $E_j(q^{-1})$  et  $F_j(q^{-1})$

et posons :

$$F_j(q^{-1}) = f_{j,0} + f_{j,1}q^{-1} + \dots + f_{j,n}q^{-n} \quad (\text{ann3.6})$$

$$A(q^{-1})\Delta(q^{-1}) = a_0^* + a_1^*q^{-1} + \dots + a_{n+1}^*q^{-n-1} \quad (\text{ann3.7})$$

Les équations (ann3.4) et (ann3.7) définissent les récurrences qui permettent de calculer  $E_{j+1}(q^{-1})$  et  $F_{j+1}(q^{-1})$  à partir de la connaissance de  $E_j(q^{-1})$  et  $F_j(q^{-1})$ .

Les conditions initiale étant :

$$1 = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}) \quad (\text{ann3.8})$$

Le coefficient  $a_0^*$  étant à 1, pour que cette identité (équation (ann3.7)) soit vérifiée, il faut que le coefficient  $e_{1,0}$  de  $E_1(q^{-1})$  soit égale à 1. Comme le degré de  $q^{-1}F_1(q^{-1})$  soit égale au degré de polynôme  $A(q^{-1})\Delta(q^{-1})$ , le degré de polynôme  $E_1(q^{-1})$  doit être égale à zéro.

Il revient:

$$E_1(q^{-1})=1, F_1(q^{-1})=q(1-A(q^{-1})\Delta(q^{-1})) \text{ et } E_{j+1}(q^{-1})=E_j + q^{-j}r_j \quad (\text{ann3.9})$$

## Bibliographies

- [1] K.S.Fu & Al, "Robotics: Control, Sensing, Vision and Intelligence", Mc Graw Hill, 1987.
  - [2] D.Stoten, "Modelling and control of elastic joint robot", ASME, Journal of dynamics system Measurement and control, Vol.109, pp 310-319, 1987.
  - [3] P.Lopez & J.N.Foulc, "Introduction à la robotique", Peter Peregrinus, IEEE controls Engineering, series 35, 1988.
  - [4] De Luca.A and Ulivi G, "Full linearization of induction motors nonlinear state-feedback", In Proc. of the 26 IEEE CDC (Los Angeles, 1987).
  - [5] Amestegui, M., Ortega, R, and Ibarra, J.M, "Adaptive linearizing-decoupling robot control: A comparative study", Proc. 5<sup>th</sup> Yale Workshop on Applications of Adaptive Systems Theory, Newhaven, CT, 1987.
  - [6] A.Uçar, S.E.Hammamci, "A Controller Based On Coefficient Diagram Method for the Robotic Manipulators", 0-7803-6542-9/00/ 2000 IEEE.
  - [7] Noriyuki Komine, Kazuki Shibata, "Derivative State Constraint Controller Via CDM for Oscillation Systems", SICE 2001 July 25-27, 2001, Nagoya
  - [8] Samy Hassany Ould Ouali, Hamid Meziane, "Commande Robuste d'une Machine Asynchrone à Flux Orienté Basée sur la Méthode de Diagramme des Coefficients", Université Mouloud Mammeri de Tizi-Ouzou, Faculté de Génie électrique et d'informatique, Département Electrotechnique.
  - [9] Shunji Manabe, "Importance of Coefficient Diagram in Polynomial Method", Proceedings of the 42nd IEEE Conference on Decision and Control Maui, Hawaii USA, December 2003
  - [10] S.Manabe, "Early Developpement of Fractional Order Control", proceedings of DETC'03 ASME 2003 Design Engineering Technical Conference and Computer and Information in Engineering Conference Chicago, Illinois, USA, September 2-6, 2003.
  - [11] S. Manabe, "Controller design of two-mass resonant system by coefficient diagram method", T: IEEE japaan, 118-D, vol.1, 1, 1998, pp.58-66.
  - [12] S.E. Hamamci and M. Koksals, "Robust Controller Design for TITO Processes with Coefficient Diagram Method", 0-7803-7729-X/03/17.00 02003 IEEE.
  - [13] T. Benjanarasuthl, D. Kumpanya, "FFC Incorporating PI Controller Designed by CDM for Temperature Control Systems", 0-7803-7657-9/02/% 17.00 0 2002 IEEE.
  - [14] N. Komine, "FFC Designed for PI Flow Control System Designed by CDM", Proceedings of the 2001 International Conference on Control, Automation and Systems, Jeju, Korea, 2001,pp.1052-105
  - [15] D. Kumpanya, P. Photong, T. Benjanarasuth, N.Komine and J. Ngamwiwit, "Industrial Controller Design by CDM", The Proceedings of ICMT'99, Pusan, Korea, pp.140-144, Oct 1999
  - [16] Patrick Boucher, Didier Dumer (1996), "La Commande Prédictive", Edition Technip Paris.
  - [17] J.M Macijowski (2002), "Predictive Control with Constraints", prentice Hall.
  - [18] Yu Shiming, Wang Haiqing Wang Xingang, "Novel Generalized Predictive Control Algorithm with Application", Proceeding of IEEE TENCON'02.
  - [19] D.W.Clarck & al," Generalized predictive control-part II: Extended and interpretation", Automatica, Vol.23, N°2, 1987, pp 148-160.
-

- [20] Winston Garfa-Gabin, Eduardo F. Camacho, "Application Of Multivariable GPC to a Four Tank Process With Unstable Transmission Zeros", 2002 IEEE International Symposium On Computer Aided Control System Design Proceedings September 18-20, 2002 Glasgow, Scotland, U.K.
- [21] Zheng Jiguang and Shi Ren, "Disturbance Rejection Performance of Generalized Predictive Control", proceedings of the 4<sup>th</sup> World Congress on intelligent Control and Automation June 10-14, 2002, Shanghai, P.R.China.
- [22] L.Santro, N.Peric, I.Petrovic, "Comparison of the Self-Tuning Generalised Predictive Controller and Pole Placement Controller", Proceedings of the 9<sup>th</sup> Mediterranean Electrotechnical Conference-Melecon '98: Academic Press, 1998. pp. 539-543.
- [23] N.Bali, "Etude des performances de la commande prédictive généralisée application aux robots manipulateurs PUMA et SCARA", Thèse de magister, ENP Alger, 1995.
-



## ملخص

هذا العمل يتضمن عرض نوعين من التحكم في الذراع الآلي، التحكم بطريقة مخطط المعاملات والتحكم التنبؤي المعمم، نهتم بدراسة نتائج التحكم بطريقة مخطط المعاملات وتطبيق هذه التقنية على الذراع الآلي ثم دراسة التحكم التنبؤي المعمم وتطبيقها على الذراع الآلي ثم عرض مقارنة بين التقنيتين. نتائج هذا التصنع برهنت فعالية الخوارزميين و من جهة أخرى الملاحظة التامة للمسار المفروض.

كلمات مفتاحية: الذراع الآلي؛ التحكم الآلي؛ مخطط المعاملات؛ قرائن الاستقرار، التحكم التنبؤي المعمم.

## Résumé

Ce mémoire consiste à présenter deux types de commande robuste d'un robot manipulateur PUMA, la première est la méthode de diagramme des coefficients (CDM), et la seconde est la commande prédictive généralisée (GPC). Nous nous intéressons à l'évaluation des performances de la commande CDM, puis l'application de cette technique au robot manipulateur PUMA, la deuxième technique de commande GPC est appliquée au robot manipulateur PUMA, enfin nous présentons une étude comparative des deux stratégies de commande, Les résultats de simulation ont montrés l'efficacité de ces deux algorithmes de commande par rapport à la poursuite de trajectoire.

Mot clé : robot manipulateur, commande robuste, diagramme des coefficients, indice de stabilité, commande prédictive généralisée.

## Abstract

This work consist in presenting two type of robust control for robot manipulator PUMA, the first is the coefficients diagram method (CDM) and the second is the generalized predictive control (GPC). we focus our interest to study the performances of CDM, then the application of this command to the robot manipulator PUMA, the second will studied and will applied to the robot manipulator PUMA. Finally we present a comparative study between these strategies, the simulation result have shown the efficiency of this both algorithm CDM and GPC and its perfect tracking of the imposed trajectory.

Key word: robot manipulator, robust control, coefficient diagramm method, stability index, generalized predictive control.

