

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE  
FACULTE D'ELECTRONIQUE ET D'INFORMATIQUE



MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER en : ELECTRONIQUE  
Spécialité : Contrôle de Processus et Robotique

Par : Soumia Wassila KHIAT

Sujet

**CONTRIBUTION DE LA COMMANDE  
PREDICTIVE ET DE LA LOGIQUE  
FLOUE A L'EXECUTION D'UNE TÂCHE  
PAR UN ROBOT MANIPULATEUR**

Soutenu publiquement le 29 Septembre 2012 devant le jury composé de :

---

Mme N. ACHOUR	Professeur	à l'USTHB	Présidente
Mr R. TOUMI	Professeur	à l'USTHB	Directeur de mémoire
Mr F. FERGUENE	Maître de conférences /A	à l'USTHB	Examineur
Mr R. MELLAH	Maître de conférences /A	à l'UMMTO	Examineur

---

---

---

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

---

---

---

---

# DEDICACES

---

---

*A mes très chers parents,  
A la mémoire de ma regrettée belle mère,  
A mon mari et à mon fils,  
A mes frères et à mes belles-sœurs,  
A mes neveux et nièces,  
A ma belle famille.*

---

---

# REMERCIEMENTS

---

---

*Je tiens tout d'abord à remercier Monsieur Redouane Toumi, Professeur à l'U.S.T.H.B. qui a proposé et dirigé ce travail .Qu'il soit assuré de ma profonde reconnaissance.*

*Je remercie tout particulièrement Mme Noura Achour, Professeur à l'U.S.T.H.B. qui me fait l'honneur de présider le jury de ce magistère.*

*Je tiens également à exprimer ma gratitude à Mr Ferguene, Maître de conférences à l'U.S.T.H.B. et Mr Mellah, Maître de conférences à l'université Mouloud Mammeri de Tizi Ouzou pour avoir accepté de juger ce travail et pour l'intérêt et le temps qu'ils y accordent.*

*Merci, enfin, à toute personne qui a contribué de près ou de loin à l'aboutissement de ce travail.*

---



---

# TABLE DES MATIERES

---



---

<b>Introduction générale</b> .....	12
------------------------------------	----

## **CHAPITRE 1**

### MODELISATION DES MANIPULATEURS RIGIDES

1.1. Introduction .....	16
1.2. Modèle géométrique direct .....	18
1.3. Modèle géométrique inverse .....	19
1.4. Modèle cinématique direct .....	19
1.5. Modèle cinématique inverse .....	20
1.6. Modèle dynamique (Formalisme de Lagrange) .....	20
1.7. Modèle dynamique sous forme compacte .....	22
1.8. Commande du robot basée sur la dynamique inverse .....	23
1.9. Conclusion .....	25

## **CHAPITRE 2**

### MODELISATION DES MANIPULATEURS FLEXIBLES

2.1. Introduction .....	27
2.2. Modélisation d'un axe flexible .....	27
2.3. Dérivation de l'équation de mouvement .....	30
2.4. Commande par inversion .....	36
2.5. Conclusion .....	37

## **CHAPITRE 3**

### COMMANDE PREDICTIVE GENERALISEE

3.1. Introduction .....	40
3.2. Modélisation CARIMA .....	40
3.3. Recherche du prédicteur optimal .....	41
3.4. Critère de commande .....	43
3.5. Loi de commande .....	45
3.6. Régulateur polynomial RST équivalent .....	45
3.7. Paramètres de synthèse de la commande GPC .....	46

## TABLE DES MATIERES

3.7.1.	Horizon d'initialisation «N1» .....	47
3.7.2.	Horizon de prédiction «HP = N2».....	48
3.7.3.	Horizon de commande «Nu».....	48
3.7.4.	Pondération sur les incréments de commande « $\lambda$ ».....	49
3.8.	Résultats et simulations .....	49
3.9.	Conclusion .....	55

## CHAPITRE 4

### COMMANDE PREDICTIVE GENERALISEE SUPERVISEE PAR LA LOGIQUE FLOUE

4.1.	Introduction .....	57
4.2.	Étapes de la supervision floue.....	57
4.2.1.	S'informer sur le comportement du système commandé.....	58
4.2.1.1.	La variance « VarX » .....	59
4.2.1.2.	Le degré de stabilité « DegSta ».....	60
4.2.2.	Contraintes et normalisation.....	60
4.2.3.	Stratégie de réglage flou.....	61
4.3.	Superviseur flou.....	61
4.3.1.	Superviseur <b>N2</b> .....	62
4.3.2.	Superviseur $\lambda$ .....	66
4.4.	Résultats et simulation .....	68
4.5.	Conclusion .....	71

## CHAPITRE 5

### LINEARISATION DES ROBOTS MANIPULATEURS AVEC PRISE EN COMPTE DES INCERTITUDES: APPROCHE FLOUE

5.1.	Introduction .....	73
5.2.	Formulation du problème.....	73
5.3.	Compensateur flou.....	76
5.3.1.	Conception du compensateur flou.....	76
5.3.1.1.	Description du système d'inférence floue utilisé .....	76
	Pour plus de détails voir annexe I. ....	77
5.3.1.2.	Description du compensateur flou .....	77
5.4.	Résultats et simulation .....	83
5.4.1.	Augmentation de la charge.....	89
5.4.2.	Augmentation de la vitesse de travail.....	93
5.5.	Conclusion .....	96

## CHAPITRE 6

### METHODE DES PERTURBATIONS SINGULIERES

6.1	Introduction .....	98
6.2	Manipulateur flexible singulièrement perturbé.....	98
6.3	Commande du manipulateur flexible avec perturbation singulière.....	103
6.3.1.	Sous-commande lente .....	103
6.3.2.	Sous-commande rapide .....	104

## TABLE DES MATIERES

6.4	Résultats et simulations .....	105
6.5	Conclusion .....	106
	<b>Conclusion et perspectives</b> .....	108

### ANNEXE I

#### Références bibliographiques

---



---

# LISTE DES FIGURES

---



---

Figure 1.1. Diagramme représentant un robot manipulateur à structure ouverte simple ayant n degrés de liberté [2].	16
Figure 1.2. Représentation schématique du robot manipulateur à 2ddl.	18
Figure 1.3. Représentation entrée-sortie du robot.	23
Figure 2.1. Axe souple en flexion[14].	28
Figure 2.2. Manipulateur plan à deux axes flexibles et deux liaisons rotoïdes [14].	31
Figure 3.1. Régulateur polynomial RST équivalent	46
Figure 3.4. Suivi de trajectoire pour $q_1$ et $q_2$	52
Figure 3.5. Erreurs de poursuite lors du suivi de trajectoire $e_1$ et $e_2$	53
Figure 3.7. Position articulaire $q_i$ , vitesse articulaire $\dot{q}_i$ , commande GPC lors du suivi de trajectoire.	53
Figure 3.8. Diagramme de Bode de la boucle GPC1	54
Figure 3.9. Diagramme de Bode de la boucle GPC2	54
Figure 4.1. Cycle de supervision floue	58
Figure 4.2. Schéma fonctionnel du superviseur flou	62
Figure 4.3.a. Fonctions d'appartenance relatives à la variable d'entrée «VarUn»	64
Figure 4.3.b. Fonctions d'appartenance relatives à la variable d'entrée «VarYn»	64
Figure 4.3.c. Fonctions d'appartenance relatives à la variable de sortie «N2n»	64
Figure 4.4.a. Fonctions d'appartenance relatives à la variable d'entrée «VarUn»	66
Figure 4.4.b. Fonctions d'appartenance relatives à la variable d'entrée «DegStan»	67
Figure 4.4.c. Fonctions d'appartenance relatives à la variable de sortie «λn»	67
Figure 4.5. Suivi de trajectoire pour $q_1$ et $q_2$	69
Figure 4.6. Evolution de la variance de signaux $q_1$ et $q_2$ en présence de supervision	69
Figure 4.7. Evolution de la variance de signaux GPC1 et GPC2 en présence de supervision	70
Figure 4.8. Evolution du degré de stabilité des boucles de commande GPC1 et GPC2.	70
Figure 5.1. Diagramme du système linéarisé [35]	82
Figure 5.2. Commande GPC du système linéarisé.	82
Figure 5.3. Suivi et erreurs de suivi sans compensation floue pour $m_1=11$ kg, $m_2= 6.5$ kg et $f_q, q = 2[\sin t \cos (t)]^T$	84
Figure 5.4. Commandes GPC1 et GPC2 et commandes actionneurs sans compensation floue pour $m_1=11$ kg, $m_2= 6.5$ kg et $f_q, q = 2[\sin t \cos (t)]^T$	84
Figure 5.5. Suivi sans compensation floue pour $m_1=11$ kg, $m_2= 6.5$ kg et $f_q, q = 2[\sin t \cos (t)]^T$	84
Figure 5.6. Suivi et erreurs de suivi pour $m_1 = 8$ kg; $m_2 = 6$ kg	85
Figure 5.7. Commandes GPC1 et GPC2 et commandes actionneurs $m_1 = 8$ kg; $m_2 = 6$ kg	86
Figure 5.8. Suivi et erreurs de suivi pour $m_1 = 24$ kg; $m_2 = 18$ kg	86

## LISTE DES FIGURES

Figure 5.9. Commandes GPC1 et GPC2 et commandes actionneurs $m_1=24$ kg; $m_2=18$ kg .....	86
Figure 5.10. Suivi et erreurs de suivi pour $\mathbf{f}_q, \mathbf{q} = 2[\text{sintcost}]^T$ .....	87
Figure 5.11. Commandes GPC1 et GPC2 et commandes actionneurs $\mathbf{f}_q, \mathbf{q} = 2[\text{sintcost}]^T$ .....	87
Figure 5.12. Suivi et erreurs de suivi sans compensation floue pour $m_p=0.2$ kg .....	90
Figure 5.13. Commandes GPC1 et GPC2 et commandes actionneurs sans compensation floue pour $m_p=0.2$ kg .....	90
Figure 5.14. Suivi et erreurs de suivi pour $m_p=0.1$ kg .....	91
Figure 5.15. Commandes GPC1 et GPC2 et commandes actionneurs pour $m_p=0.1$ kg .....	91
Figure 5.16. Suivi et erreurs de suivi pour $m_p=0.4$ kg .....	91
Figure 5.17. Commandes GPC1 et GPC2 et commandes actionneurs pour $m_p=0.4$ kg .....	91
Figure 5.18. Déviation des extrémités des axes 1 et 2 pour $m_p= 0.2$ kg et $m_p=0.4$ kg.....	92
Figure 5.19. Les déviations maximales $y_{1\_max}$ et $y_{2\_max}$ des extrémités des axes 1 et 2 en fonction de $m_p$ .....	92
Figure 5.20. Suivi et erreurs de suivi pour $\omega_1= 5$ rad/s et $\omega_2= 6$ rad/s .....	93
Figure 5.21. Commandes GPC1 et GPC2 et commandes actionneurs pour $\omega_1= 5$ rad/s et $\omega_2= 6$ rad/s.....	93
Figure 5.22. Suivi et erreurs de suivi pour $\omega_1= 8$ rad/s et $\omega_2= 10$ rad/s.....	94
Figure 5.23. Commandes GPC1 et GPC2 et commandes actionneurs pour $\omega_1= 8$ rad/s et $\omega_2= 10$ rad/s.....	94
Figure 5.24. Déviation des extrémités des axes 1 et 2 pour différentes valeurs de $\omega_1$ et $\omega_2$ .....	95
Figure 5.25. Les déviations maximales $y_{1\_max}$ et $y_{2\_max}$ des extrémités des axes 1 et 2 en fonction de $\omega_1$ et $\omega_2$ .....	95
Figure 6.1. Déviation des extrémités des axes 1 et 2 avec commande rigide (à gauche) et commande composite (à droite). 105	105
Figure 6.2. Erreurs de suivi $e_1$ et $e_2$ avec commande rigide (à gauche) et commande composite (à droite).....	105
Figure A.1. Forme les plus courantes des fonctions d'appartenance .....	111
Figure A.2. Méthode de fuzzification singleton .....	113
Figure A.3. Méthode d'inférence « product inference ».....	114
Figure A.4. défuzzification par le centre de gravité.....	115

---

---

# LISTE DES TABLEAUX

---

---

Tableau 3.1. Influence de $N2$ sur quelques performances de la commande en boucle fermée [26].	48
Tableau 3.2. Influence de $\lambda$ sur les performances du système en boucle fermée[26]	49
Tableau 3.3. Paramètres du robot manipulateur rigide	51
Tableau 4.1. Table des règles floues reliant $N2n$ aux indices de performance $VarYn$ et $VarUn$ .	65
Tableau 4.2. Table des règles définissant $\lambda n$ en fonction de $Degstan$ et $VarUn$ .	68
Tableau 5.1. $e1\_max$ et $e2\_max$ en fonction des masses $m1$ et $m2$	85
Tableau 5.2. Erreurs de suivi maximales pour différentes valeurs de $f(q, q)$	87
Tableau 5.3. Paramètre du manipulateur flexible	88
Tableau 5.4. Fréquences de vibration en fonction de la charge $mp$	89
Tableau 5.5. Performances de la commande pour $mp=0.4$ .	93
Tableau 5.6. Performances pour $\omega1 = 8$ rad/s, $\omega2 = 9$ rad/s	96

# **Introduction**

## **Générale**

---

## Introduction générale

La commande prédictive s'est considérablement développée durant les dernières années, tant dans le domaine de la recherche que dans le domaine de l'industrie. Ce succès peut être attribué au fait que la commande prédictive est la méthode la plus générale de poser le problème de la commande de processus dans le domaine temporel.

La commande prédictive a prouvé ses performances à travers de nombreuses applications industrielles, tout en s'avérant simple d'implantation. La littérature sur ce sujet foisonne depuis de nombreuses années d'exemples illustrant la capacité de ce type de structures. De fait, le succès de cette stratégie de commande avancée réside principalement dans sa capacité à faire intervenir la connaissance d'une trajectoire à suivre dans le futur, sur un horizon fini. Sa description temporelle facilite de plus la prise en compte des contraintes physiques des systèmes. Toutefois, cette démarche de commande n'a cessé d'évoluer et, partant de structures classiques reposant sur des idées relativement intuitives, de multiples avancées ont été réalisées, principalement dans l'optique d'améliorer d'une part la robustesse et les performances temps réel et de permettre d'autre part l'extension à d'autres classes de système.

La logique floue est aujourd'hui utilisée dans tous les domaines de l'automatique. Dans la modélisation, la commande, l'approximation, la supervision...etc. Elle permet d'aborder de façon simple des problèmes complexes.

Le travail présenté dans ce mémoire a pour but de démontrer les avantages apportés par la commande prédictive à l'exécution d'une tâche par un robot manipulateur et d'exhiber les améliorations que peut amener la logique floue à cette commande.

Nous entamons le travail par la modélisation dynamique des robots manipulateurs rigides et flexibles. Nous avons choisi de travailler sur deux robots manipulateurs plans à deux degrés de liberté rotoïdes. Le premier manipulateur possède des axes rigides et le second est un manipulateur léger à axes flexibles. Les modèles dynamiques obtenus ont ensuite été linéarisés.

Après linéarisation, chaque système à commander se présente sous forme d'un système découplé à deux entrées et deux sorties, composé de deux doubles intégrateurs indépendants entre eux. La commande de chaque robot linéarisé est assurée par deux boucles

de commande prédictive généralisée GPC indépendantes. Une pour chaque double intégrateur échantillonné.

Les performances de la commande GPC dépendent beaucoup du choix des paramètres de synthèse : horizon de commande  $N_u$ , horizon de prédiction  $N_2$  ainsi que de la pondération sur les incréments de commande  $\lambda$ . Un réglage approprié de ces paramètres est donc nécessaire. Une couche de supervision floue est ajoutée à la boucle de commande GPC, et permet de régler les paramètres de synthèse de la commande GPC.

La linéarisation du modèle dynamique d'un robot manipulateur repose sur l'hypothèse que notre connaissance du modèle du robot est parfaite. Ce qui dans la pratique est impossible. Les dynamiques non modélisées, qu'elles soient dues à des perturbations structurées ou non structurées, font que la linéarisation est irréalisable dans la pratique. Une solution combinant la linéarisation à un contrôleur flou est utilisée afin de compenser les erreurs dues aux perturbations. La méthode proposée repose sur l'idée que les incertitudes peuvent être estimées par un système flou.

Afin de pouvoir tester, les stratégies de commande décrites précédemment sur le manipulateur flexible, nous appliquons à ce dernier la théorie des perturbations singulières. Celle-ci permet de décomposer le modèle du robot en un sous-modèle rigide et un sous-modèle flexible. Une commande composite nous permet de commander séparément les deux sous systèmes.

Des tests réalisés en simulation permettent d'évaluer les performances des différentes stratégies de commande appliquées aux manipulateurs rigide et flexible.

#### - **Organisation de la thèse :**

Le chapitre 1 a pour objet la modélisation des robots manipulateurs rigides. Il comprend une brève présentation des différents modèles : géométrique, cinématique et dynamique. L'intérêt sera porté particulièrement sur les manipulateurs plans à deux degrés de liberté rotoïdes. Le chapitre abordera brièvement la linéarisation des robots manipulateurs (commande par couple calculé).

Le chapitre 2 aborde le problème de la modélisation des robots manipulateurs à axes flexibles. Le modèle de la poutre d'Euler-Bernoulli est utilisé pour modéliser les axes flexibles. Nous nous intéresserons particulièrement au cas d'un manipulateur plan à deux axes

flexibles et deux liaisons rotoïdes (liaisons rigides). La commande par inversion est abordée à la fin du chapitre.

Le Chapitre 3 est dédié à la présentation théorique de la commande prédictive généralisée GPC. La commande GPC sera présentée sous une approche polynomiale. La méthode de synthèse de la loi de commande GPC sera présentée ainsi que le régulateur polynomial RST équivalent déduit à partir de cette loi de commande. Le chapitre abordera la question du réglage des paramètres de la commande GPC.

Le chapitre 4 propose une méthode de réglage des paramètres de la GPC. Cette méthode est basée sur la mesure des performances de la commande. L'influence des paramètres de la GPC sur ces différentes performances mesurées est connue et a été abordée au chapitre I. Sur cette base, une stratégie floue permettant le réglage en ligne des paramètres de la commande GPC est développée.

Le chapitre 5 reprend de façon critique la question de la linéarisation (commande par couple calculé) des robots manipulateurs présentée au chapitre I, en démontrant les limites de celle-ci et la nécessité de la prise en compte des incertitudes (incertitudes de modélisation et incertitudes dues à l'environnement) lors d'une implémentation réelle. Ce chapitre présente une approche qui combine la logique floue et la commande par couple calculé afin de compenser l'effet des incertitudes.

Dans le Chapitre 6, nous présentons la théorie des perturbations singulières. Cette dernière est appliquée au manipulateur flexible dans le but de découpler la dynamique flexible de la dynamique rigide. Une commande composite permet de commander séparément les sous-systèmes rigide et flexible. Nous pouvons ainsi appliquer au sous-système rigide, les stratégies de commande proposées dans les chapitres précédents.

À la fin des chapitres 3, 4, 5 et 6, nous pouvons retrouver les résultats de simulation relatifs aux stratégies de commande présentées.

Une conclusion générale rappelle les principaux résultats obtenus. Elle présente les limites de ce travail et ouvre des perspectives vers d'éventuelles améliorations.

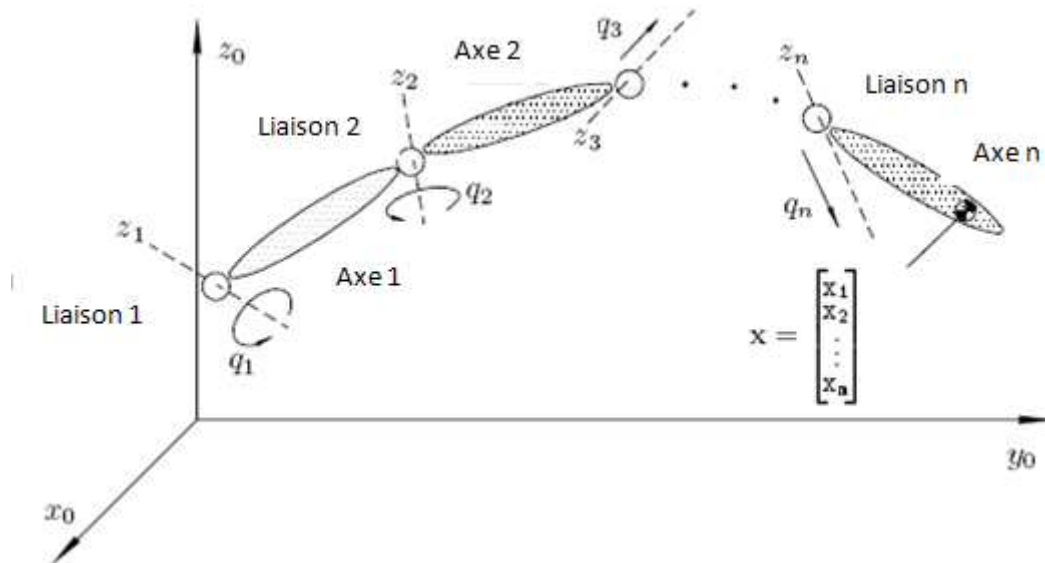
# Chapitre 1

---

## **Modélisation des Robots Manipulateurs Rigides**

### 1.1. Introduction

Les robots manipulateurs sont des systèmes mécaniques articulés composés d'axes (ou liens) et d'articulations (ou liaisons). Les liaisons sont principalement de deux types : prismatiques et rotoïdes[1]. Dans le présent travail, nous nous intéressons aux manipulateurs à structure ouverte simple.



**Figure 1.1. Diagramme représentant un robot manipulateur à structure ouverte simple ayant  $n$  degrés de liberté [2].**

Considérons l'exemple du bras articulé à  $n$  degrés de liberté représenté sur la figure 1.1.

Soit un repère de référence tridimensionnel  $\{x_0, y_0, z_0\}$  situé à la base du robot. Les axes du robot sont numérotés respectivement à partir de la base ou axe (0) jusqu'à l'axe ( $n$ ) lié à l'effecteur terminal. Les liaisons entre les axes sont numérotées de sorte que la  $i^{\text{ème}}$  liaison relie l'axe ( $i$ ) à l'axe ( $i - 1$ ). Un actionneur est placé au niveau de chaque liaison( $i$ ) et produit les mouvements de l'axe( $i$ ).

La coordonnée généralisée  $q_i$  correspond au déplacements angulaires de l'axe  $i$ .  $q_i$  est mesurée par rapport au prolongement de l'axe ( $i - 1$ ). Un robot manipulateur à  $n$  degrés de liberté (ddl) possède un vecteur  $q$  de  $n$  coordonnées généralisées ou coordonnées articulaires [2]:

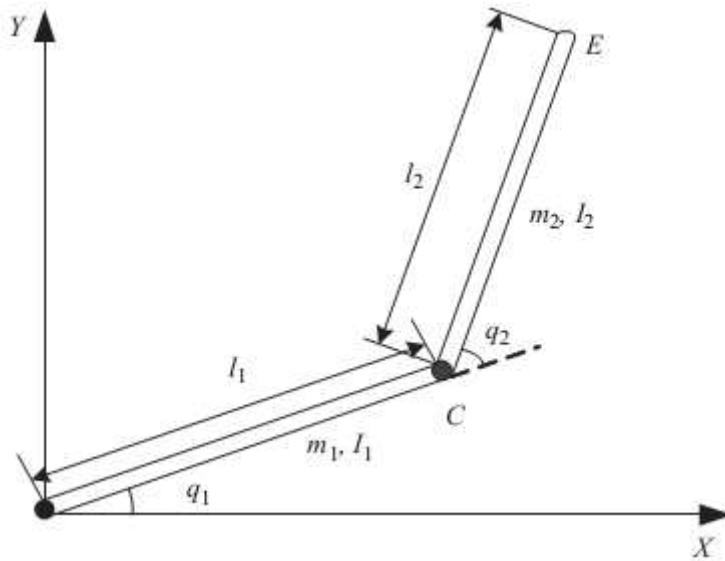
$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \quad (1.1)$$

La position et l'orientation de l'effecteur sont exprimées dans le repère de référence placé à la base du robot, à l'aide des coordonnées opérationnelles  $x_i$  composant le vecteur  $X[2]$  :

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \quad m \leq n. \quad (1.2)$$

Dans ce premier chapitre, nous allons aborder la modélisation des robots manipulateurs rigides. Nous allons également présenter le robot manipulateur rigide qui sera utilisé au cours de la simulation.

Comme nous pouvons le voir sur la figure 1.2., le robot en question est un manipulateur plan comportant deux liens rigides et deux liaisons rotoïdes. Le robot comprend un actionneur à chaque articulation. Les liens du robot sont de longueurs  $l_1$  et  $l_2$ , et de masses  $m_1$  et  $m_2$ . Les distances entre les axes de rotation et les centres de masses sont notées respectivement  $l_{c1}$  et  $l_{c2}$ .  $I_1$  et  $I_2$  sont les moments d'inertie des liens du robot relativement aux axes qui passent par leurs centres de masses respectifs et sont parallèles à l'axe  $z$  du repère de référence. Le robot évolue dans le plan vertical  $xy$ .



**Figure 1.2. Représentation schématique du robot manipulateur à 2ddl.**

Le vecteur des positions articulaires  $q$  correspondant au robot à 2 ddl s'écrit comme suit :

$$q = [q_1 \ q_2]^T = [\theta_1 \ \theta_2]^T \quad (1.3)$$

## 1.2. Modèle géométrique direct

Le modèle géométrique direct permet de déterminer la position et l'orientation de l'organe terminal du manipulateur dans le repère de référence en fonction des variables articulaires[3].

Il s'écrit sous la forme :

$$X = f(q) \quad (1.4)$$

$q = [q_1, q_2, \dots, q_n]^T \in R^n$  est le vecteur des variables articulaires.

$X = [x_1, x_2, \dots, x_m]^T \in R^m$  est le vecteur des variables opérationnelles  $m \leq n$ .

Les coordonnées  $x_1$  et  $x_2$  qui définissent la position de l'effecteur du robot à 2 ddl dépendent des coordonnées articulaires  $q_1$  et  $q_2$ .

Les relations suivantes sont immédiates :

$$x_1 = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \quad (1.5)$$

$$x_2 = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \quad (1.6)$$

### 1.3. Modèle géométrique inverse

Le modèle géométrique inverse permet de déterminer la posture du robot et donc le vecteur des variables articulaires à partir du vecteur de coordonnées opérationnelles[3], le modèle s'écrit sous la forme suivante :

$$q = f^{-1}(X) \quad (1.7)$$

Il est plus intuitif de spécifier la position désirée du robot dans l'espace des coordonnées opérationnelles. Nous pouvons ensuite à l'aide du modèle cinématique inverse trouver la ou les configurations correspondantes dans le système de coordonnées articulaires. Pour le cas de notre robot manipulateur plan à 2ddl, le modèle géométrique inverse peut être représenté par les équations suivantes :

$$q_1 = \tan^{-1}\left(\frac{x_1}{x_2}\right) - \tan^{-1}\left(\frac{l_2 \sin(q_2)}{l_1 + l_2 \cos(q_2)}\right), \quad (1.8)$$

$$q_2 = \cos^{-1}\left(\frac{x_1^2 + x_2^2 - l_1^2 - l_2^2}{2l_1 l_2}\right). \quad (1.9)$$

Dans le cas du robot étudié, pour une même position de l'effecteur, il existe deux configurations possibles pour les axes du robot, i.e. deux valeurs possibles pour le vecteur de coordonnées articulaires  $q$ .

### 1.4. Modèle cinématique direct

Le modèle cinématique direct permet de déterminer la vitesse de l'organe terminal en fonction des vitesses articulaires[1]. Le modèle est décrit par l'équation :

$$\dot{X} = J(q)\dot{q} \quad (1.10)$$

Où  $J(q) \in R^{m \times n}$  est la matrice jacobienne.

A partir du modèle géométrique direct, les relations suivantes entre la vitesse de l'effecteur et les vitesses articulaires de notre manipulateur à 2ddl peuvent être obtenues [2] :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (1.11)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = J(q) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (1.12)$$

### 1.5. Modèle cinématique inverse

Le modèle cinématique inverse permet de déterminer les vitesses articulaires en fonction de la vitesse de l'organe terminal [2]. Pour les manipulateurs non redondants ( $n=m$ ), le modèle s'écrit :

$$\dot{q} = J^{-1}(q)\dot{X} \quad (1.13)$$

La solution de l'équation existe si J est de rang plein, cela est valable tant que le manipulateur ne passe pas par une configuration singulière. Pour les manipulateurs redondants, le modèle cinématique inverse admet plusieurs solutions possibles. Le choix d'une solution particulière est guidé par l'optimisation d'une fonction objective.

La relation entre les vitesses articulaires et la vitesse de l'effecteur de notre robot s'écrit comme suit[2]:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \frac{1}{l_1 l_2 \sin(q_2)} \begin{bmatrix} l_2 \cos(q_1 + q_2) & l_2 \sin(q_1 + q_2) \\ -l_1 \cos(q_1) - l_2 \cos(q_1 + q_2) & -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} \quad (1.14)$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \frac{1}{l_1 l_2 \sin(q_2)} \begin{bmatrix} l_2 \cos(q_1 + q_2) & l_2 \sin(q_1 + q_2) \\ -l_1 \cos(q_1) - l_2 \cos(q_1 + q_2) & -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} \quad (1.15)$$

Notons que  $\sin(q_2)$  apparaît dans le dénominateur de tous les termes de la matrice  $J^{-1}(q_d)$ . Par conséquent, les configurations  $q_2 = k\pi$ , ( $k = 1, 2, \dots$ ) avec  $q_1$  quelconque sont des configurations singulières.

Dans le présent travail, il n'y a pas à se soucier des configurations singulières. La commande étant entièrement effectuée dans l'espace articulaire et donc le Jacobien n'est jamais utilisé.

Nous allons, à présent, nous intéresser à la modélisation dynamique du manipulateur.

### 1.6. Modèle dynamique (Formalisme de Lagrange)

Le modèle dynamique d'un robot consiste en un ensemble d'équations différentielles ordinaires dont les variables sont les vecteurs de positions, de vitesses et d'accélération[2].

Les équations dynamiques d'un robot manipulateur peuvent être obtenues en utilisant le formalisme de Newton ou via celui de Lagrange. Ces deux formalismes sont détaillés dans bon nombre d'ouvrages de robotique. Citons à titre d'exemples [1, 3, 4]

Dans cette section, nous nous intéressons à la modélisation dynamique du robot manipulateur plan à deux degrés de liberté rotoïdes représenté sur la figure 1.2. Nous allons calculer l'équation de Lagrange

Soit  $K_i$  et  $P_i$  respectivement l'énergie cinétique et potentielle relatives à l'axe  $i$ .

Commençons par calculer les énergies cinétique et potentielle de l'axe 1:

$$K_1 = \frac{1}{2} m_1 l_{c1}^2 \dot{q}_1^2 + \frac{1}{2} I_1 \dot{q}_1^2 \quad (1.16)$$

$$P_1 = m_1 g l_{c1} \sin(q_1) \quad (1.17)$$

Pour l'axe 2, nous allons d'abord écrire l'expression de la position du centre de masses  $s_2 = [s_{2x}, s_{2y}]^T$  par rapport au repère de référence :

$$s_{2x} = l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2) \quad (1.18)$$

$$s_{2y} = l_1 \sin(q_1) + l_{c2} \sin(q_1 + q_2) \quad (1.19)$$

D'où la relation :

$$\dot{s}_2^T \dot{s}_2 = l_1^2 \dot{q}_1^2 + l_{c2}^2 (\dot{q}_1 + \dot{q}_2)^2 + 2l_1 l_{c2} \cos(q_2) (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) \quad (1.20)$$

Nous obtenons :

$$K_2 = \frac{1}{2} m_2 \dot{s}_2^T \dot{s}_2 + \frac{1}{2} I_1 (\dot{q}_1 + \dot{q}_2)^2 \quad (1.21)$$

$$P_2 = m_2 g (l_1 \sin(q_1) + l_{c2} \sin(q_1 + q_2)) \quad (1.22)$$

En calculant le Lagrangien du système  $\mathcal{L} = K_1 + K_2 - P_1 - P_2$  et en remplaçant dans l'équation de Lagrange :

$$\frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right] - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i, \quad i = 1, 2 \quad (1.23)$$

Nous obtenons les équations de mouvement du manipulateur plan à 2ddl rotoïdes [2] :

$$\tau_1 = [m_1 \ell_{c1}^2 + I_1 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_2] \ddot{q}_1 + [m_2(l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_2] \ddot{q}_2 - m_2 l_1 l_{c2} \sin(q_2) (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2), \quad (1.24)$$

$$\tau_2 = [m_2(l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_2] \ddot{q}_1 + [m_2 l^2 + I_2] \ddot{q}_2 + m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1^2 + m_2 g l_{c2} \cos(q_1 + q_2) \quad (1.25)$$

## 1.7. Modèle dynamique sous forme compacte

Dans ce paragraphe, Le modèle dynamique du robot sera présenté sous une nouvelle forme appelée forme compacte[2].

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (1.26)$$

Avec :

$$C(q, \dot{q})\dot{q} = \dot{M}(q)\dot{q} - \frac{1}{2} \frac{\partial}{\partial q} [\dot{q}^T M(q) \dot{q}] \quad (1.27)$$

$$g(q) = \frac{\partial U(q)}{\partial q} \quad (1.28)$$

L'équation (1.26) est l'équation du modèle dynamique du robot manipulateur à n degrés de liberté. C'est une équation différentielle vectorielle non linéaire.

$C(q, \dot{q})\dot{q}$  est un vecteur de dimension n appelé vecteur des forces centrifuges et de Coriolis.  $g(q)$  est un vecteur de dimension n regroupant les forces et les couples gravitationnels.  $\tau$  est un vecteur de dimension n constitué des couples et forces appliqués par les actionneurs au niveau des liaisons.

Le fait que la matrice  $M(q)$  soit définie positive et donc inversible, nous permet d'exprimer le modèle dynamique (1.26) du robot dans l'espace d'état. Avec  $[q^T \dot{q}^T]^T$  en vecteur d'état, et  $\tau$  en vecteur d'entrée, nous parvenons à l'équation d'état suivante :

$$\frac{d}{dt} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M(q)^{-1} [\tau(t) - C(q, \dot{q})\dot{q} - g(q)] \end{bmatrix} \quad (1.29)$$

Le modèle (1.29) peut être vu comme un système dynamique ayant pour entrée le vecteur  $\tau$  et comme sortie le vecteur  $[q^T \dot{q}^T]^T$ . Ceci est illustré par la figure suivante :



Figure 1.3. Représentation entrée-sortie du robot.

Pour le calcul de la commande, il est plus pratique d'écrire le modèle dynamique du robot à deux degrés de liberté obtenu au paragraphe précédent sous forme matricielle[2] :

$$\underbrace{\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix}}_{M(q)} \ddot{q} + \underbrace{\begin{bmatrix} C_{11}(q) & C_{12}(q) \\ C_{21}(q) & C_{22}(q) \end{bmatrix}}_{c(q)} \dot{q} + \underbrace{\begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix}}_{g(q)} = \tau \quad (1.30)$$

$$M_{11}(q) = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_1 + I_2$$

$$M_{12}(q) = m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_2$$

$$M_{21}(q) = m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_2$$

$$M_{22}(q) = m_2 l_{c2}^2 + I_2$$

$$C_{11}(q, \dot{q}) = -m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2$$

$$C_{12}(q, \dot{q}) = -m_2 l_1 l_{c2} \sin(q_2) (\dot{q}_1 + \dot{q}_2)$$

$$C_{21}(q, \dot{q}) = m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1$$

$$C_{22}(q, \dot{q}) = 0$$

$$g_1(q) = [m_1 l_{c1} + m_2 l_1] g \cos(q_1) + m_2 l_{c2} g \cos(q_1 + q_2)$$

$$g_2(q) = m_2 l_{c2} g \cos(q_1 + q_2)$$

## 1.8. Commande du robot basée sur la dynamique inverse

A partir de l'équation (1.26) décrivant le comportement dynamique du robot, nous pouvons constater que le robot manipulateur est un système non linéaire.

Le but de cette section est de développer une commande linéarisante pour le modèle dynamique du robot manipulateur. Pour retrouver plus de détails sur les méthodes de linéarisation, il est possible de consulter les ouvrages[5] [6].

Nous commençons par extraire l'expression de l'accélération articulaire à partir de l'équation (1.26). Nous obtenons :

$$\ddot{q} = M^{-1}(q)[\tau - (C(q, \dot{q})\dot{q} + g(q))] \quad (1.31)$$

Si nous appliquons à présent, la commande suivante :

$$\tau = M(q)u + C(q, \dot{q})\dot{q} + g(q) \quad (1.32)$$

Nous aboutissons à l'équation représentant le système en boucle ouverte :

$$u = \ddot{q} \quad (1.33)$$

Le système obtenu est un double intégrateur. Le système est linéarisé et découplé.

Dans le cas idéal, il suffirait donc de donner comme signal de commande les dérivées secondes des positions articulaires désirées pour que le système suive la trajectoire désirée.

Comme il est impossible d'avoir le modèle exact du robot, il y aura toujours entre les positions désirées et réelles une différence qui va s'accroître avec le temps. L'erreur de position  $e$  est définie par :

$$e = q_d - q \quad (1.34)$$

Où  $q_d$  est la position de référence. De façon similaire, Nous définissons l'erreur de vitesse comme étant la différence entre la vitesse désirée et la vitesse réelle :

$$\dot{e} = \dot{q}_d - \dot{q} \quad (1.35)$$

Le vecteur de commande (proportionnelle dérivée) :

$$u = \ddot{q}_d + K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) \quad (1.36)$$

Si nous appliquons au robot manipulateur la commande (1.36) en cascade avec la commande linéarisante (1.32), Nous obtenons l'équation différentielle suivante, décrivant la dynamique du robot en boucle fermée :

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \quad (1.37)$$

La dynamique de la réponse est déterminée par les gains proportionnel  $K_p$  et dérivé  $K_d$ .

## 1.9. Conclusion

Nous avons consacré ce chapitre à la modélisation des robots manipulateurs rigides et ce dans le but de modéliser le robot manipulateur plan à deux degrés de liberté rotoïdes et deux segments rigides qui sera utilisé dans la suite du travail.

Une brève définition des modèles, géométrique, cinématique et dynamique directs et inverses a été introduite. Pour chaque définition, le modèle correspondant au manipulateur plan à 2ddl a été calculé.

A la fin du chapitre, une commande linéarisante basée sur la dynamique inverse a été proposée afin de linéariser le modèle dynamique du robot manipulateur.

# Chapitre 2

---

## **Modélisation des Robots Manipulateurs flexibles**

## 2.1. Introduction

Les bras manipulateurs flexibles sont légers et offrent bien des avantages par rapport aux manipulateurs rigides. Ils consomment moins d'énergie, sont plus rapides et fournissent un meilleur rapport charge utile/masse du bras manipulateur[7].

Pour ce type de manipulateurs, la flexibilité de la structure ne peut être négligée lors de la modélisation. Si nous voulons positionner l'effecteur du robot avec précision malgré les vibrations dues à la flexibilité, des modèles performants et des stratégies de commande efficaces doivent être développés.

Les méthodes les plus utilisées pour la modélisation des manipulateurs flexibles sont la méthode des modes supposés [8, 9] où les déformations élastiques peuvent être approchées par une combinaison linéaire de coordonnées modales et la méthode des éléments finis [10] [7] où un élément flexible est représenté par plusieurs éléments rigides reliés par des ressorts. Dans les deux cas la méthode de Lagrange est utilisée pour le calcul du modèle dynamique du robot.

Dans le présent travail, nous nous intéressons à la modélisation des robots manipulateurs à axes flexibles. Le modèle dynamique obtenu sera utilisé comme modèle interne pour la commande prédictive et pour modéliser le robot lors de la simulation. Nous commençons par étudier le modèle relatif à un axe flexible. La méthode des modes supposés est utilisée.

## 2.2. Modélisation d'un axe flexible

Pour modéliser un axe flexible, nous allons utiliser le modèle de la poutre d'Euler-Bernoulli [7, 11-13]. Considérons l'axe flexible représenté sur la figure 2.1 et sujet à des déformations transversales. Tout autre type de déformation est supposé négligeable [14].

L'équation de la poutre d'Euler-Bernoulli s'écrit comme suit :

$$EI \frac{\partial^4 w(x, t)}{\partial x^4} + \rho \frac{\partial^2 w(x, t)}{\partial t^2} = 0 \quad (2.1)$$

$EI$  est la rigidité en flexion, et  $\rho$  la masse linéique.

En utilisant la méthode de séparation des variables et en prenant  $w(x, t) = \phi(x)\delta(t)$ , Nous obtenons :

$$\frac{EI}{\rho\phi(x)} \frac{\partial^4 \phi(x)}{\partial x^4} = \frac{-1}{\delta(t)} \frac{\partial^2 \delta(t)}{\partial t^2} = \omega^2 \quad (2.2)$$

$\omega$  est une constante.

L'équation (2.2) aboutit aux deux équations suivantes :

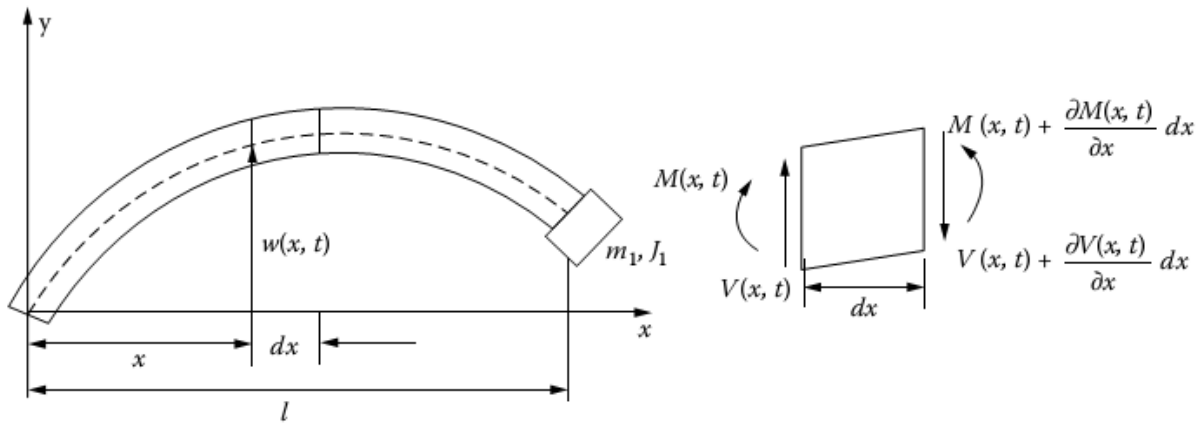
$$\frac{d^4 \phi(x)}{dx^4} - \frac{\rho\omega^2}{EI} \phi(x) = 0 \quad (2.3)$$

$$\frac{d^2 \delta(t)}{dt^2} + \omega^2 \delta(t) = 0 \quad (2.4)$$

La solution de l'équation différentielle de la variable  $t$  (2.4) est de la forme :

$$\delta(t) = A \cos(\omega t) + B \sin(\omega t) \quad (2.5)$$

Où les constantes A et B sont déterminées à partir des conditions initiales.



**Figure 2.1. Axe souple en flexion[14].**

La solution de l'équation différentielle de la variable  $x$  (2.3) s'écrit :

$$\phi(x) = C_1 \cosh(\beta x) + C_2 \sinh(\beta x) + C_3 \cos(\beta x) + C_4 \sin(\beta x) \quad (2.6)$$

Avec  $C_1, C_2, C_3, C_4$  constantes et  $\beta^4 = \frac{\rho\omega^4}{EI}$ .

Les constantes  $C_1, C_2, C_3, C_4$  sont déterminées à partir des conditions aux limites.

La fonction  $\phi(x)$  est la fonction des déformées modales de la poutre.

Les conditions aux limites pour un axe flexible de robot manipulateur sont généralement choisies comme étant « extrémité encastree – extrémité libre » : l'axe flexible est encastree à l'extrémité  $x = 0$  avec zéro déformation. A l'extrémité  $x = l$ , l'axe est libre avec un moment de flexion et un effort tranchant nuls. Toutefois, nous considérons ici le cas où une masse  $m_L$ , de moment d'inertie  $J_L$ , est fixée à l'extrémité  $x = l$  de l'axe et va ainsi générer un effort tranchant et un moment de flexion non nuls. Ce dernier cas est plus proche de la réalité [14].

En  $x = 0$ , les conditions aux limites sont :

Déplacement dû à la flexion :

$$w(0, t) = 0, \quad (2.7)$$

Rotation due à la flexion :

$$\frac{\partial w(0, t)}{\partial x} = 0 \quad (2.8)$$

En  $x = l$ ,

Moment fléchissant :

$$EI \frac{\partial^2 w(x, t)}{\partial x^2} \Big|_{x=l} = -J_L \frac{d^2}{dt^2} \left( \frac{\partial w(x, t)}{\partial x} \Big|_{x=l} \right) \quad (2.9)$$

Effort tranchant :

$$EI \frac{\partial^3 w(x, t)}{\partial x^3} \Big|_{x=l} = m_L \frac{d^2}{dt^2} (w(x, t)|_{x=l}) \quad (2.10)$$

L'application des conditions aux limites à l'équation des déformées modales (2.6) donne l'équation fréquentielle suivante :

$$\begin{aligned} 1 + \cos(\beta l) \cosh(\beta l) - \frac{m_L \beta}{\rho} (\sin(\beta l) \cosh(\beta l) - \cos(\beta l) \sinh(\beta l)) \\ - \frac{J_L \beta^3}{\rho} (\sin(\beta l) \cosh(\beta l) + \cos(\beta l) \sinh(\beta l)) \\ + \frac{m_L J_L \beta^4}{\rho^2} (1 - \cos(\beta l) \cosh(\beta l)) = 0 \end{aligned} \quad (2.11)$$

$$\begin{aligned}
 1 + \cos(\beta l) \cosh(\beta l) - \frac{m_L \beta}{\rho} (\sin(\beta l) \cosh(\beta l) - \cos(\beta l) \sinh(\beta l)) \\
 - \frac{J_L \beta^3}{\rho} (\sin(\beta l) \cosh(\beta l) + \cos(\beta l) \sinh(\beta l)) \\
 + \frac{m_L J_L \beta^4}{\rho^2} (1 - \cos(\beta l) \cosh(\beta l)) = 0
 \end{aligned} \tag{2.12}$$

Cette équation a un nombre infini de solutions  $\{\beta_i, i = 1, \dots, \infty\}$ . Chaque solution  $\beta_i$  correspond à une pulsation naturelle  $\omega_i$  et à une déformée modales  $\Phi_i(x)$  telle que :

$$\phi_i(x) = a_i [\cosh(\beta_i x) - \cos(\beta_i x) - b_i (\sinh(\beta_i x) - \sin(\beta_i x))] \tag{2.13}$$

$$b_i = \frac{\cosh(\beta_i l) + \cos(\beta_i l) - \frac{J_L \beta_i^3}{\rho} (\sinh(\beta_i l) + \sin(\beta_i l))}{\sinh(\beta_i l) + \sin(\beta_i l) - \frac{J_L \beta_i^3}{\rho} (\cosh(\beta_i l) - \cos(\beta_i l))} \tag{2.14}$$

La constante  $a_i$  est calculée à partir de l'équation de normalisation :

$$\int_0^l \phi_i^2(x) dx = l \tag{2.15}$$

Les déformées modales remplissent la condition d'orthogonalité.

La solution de l'équation de la poutre d'Euler-Bernoulli s'écrit :

$$w(x, t) = \sum_{i=1}^{\infty} \phi_i(x) \delta_i(t) \tag{2.16}$$

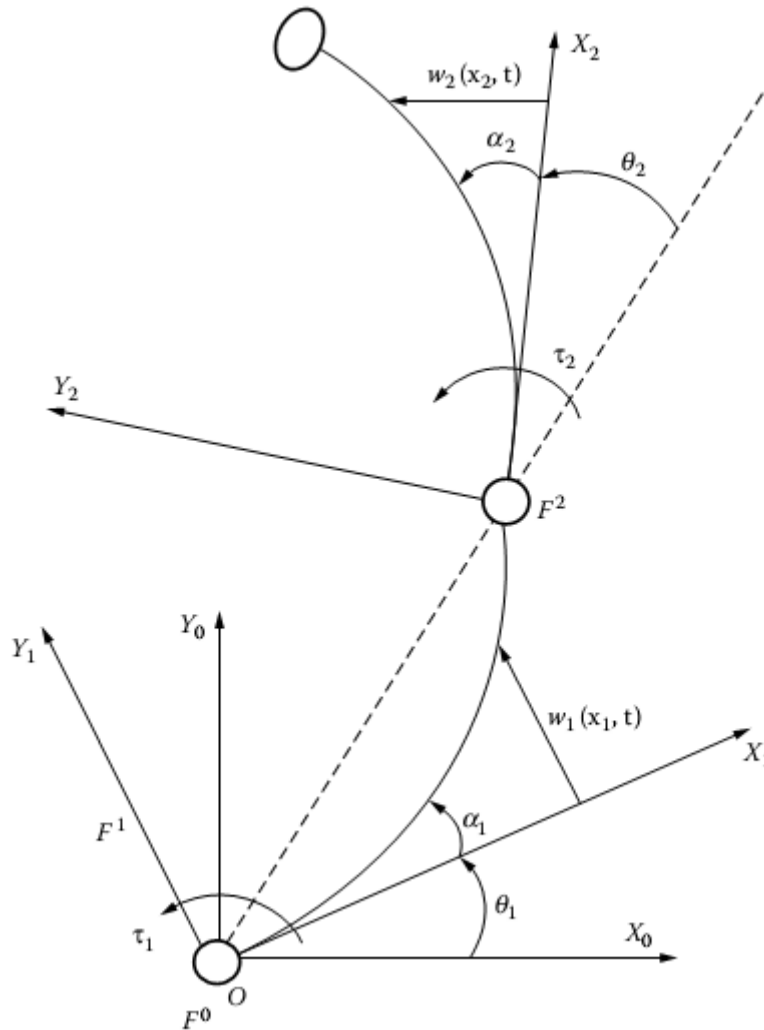
### 2.3. Dérivation de l'équation de mouvement

Considérons le cas d'un manipulateur plan à chaîne ouverte simple composé de  $n$  axes flexibles et  $n$  degrés de liberté rotoïdes (liaisons rigides). Les axes sont sujets à des vibrations de flexion dans le plan de travail du robot. Tout autre type de déformation est considéré négligeable. Chaque axe est modélisé comme une poutre d'Euler-Bernoulli encastree à un actionneur à sa première extrémité et ayant une masse fixée à l'autre. Cette masse possède une inertie de rotation. Afin d'obtenir un modèle à dimension finie de l'axe flexible, Nous allons employer la méthode des modes supposés (assumed modes). La déformation de chaque axe  $i$  est décrite par la somme pondérée de  $m_i$  déformées modales :

$$w_i(x_i, t) = \sum_{j=1}^{m_i} \phi_{ij}(x_i) \delta_{ij}(t) \tag{2.17}$$

Ici,  $x_i$  représente la position le long de la ligne neutre de l'axe  $i$  (la ligne neutre est la position qu'occuperait l'axe  $i$  en l'absence de déformation).  $\phi_{ij}$  et  $\delta_{ij}$  sont respectivement la déformée modale et la coordonnée modale associées à l'axe  $i$  et à la pulsation naturelle  $\omega_j$ .

Prenons l'exemple du manipulateur flexible à 2ddl représenté sur la figure 2.2. Le repère  $F^0$  est le repère fixe lié à l'environnement.  $F^i$  est le repère lié à l'axe  $i$ .



**Figure 2.2. Manipulateur plan à deux axes flexibles et deux liaisons rotoïdes [14].**

Le passage d'un repère vers un autre est représenté par une matrice homogène. La matrice de passage entre les repères  $F^1$  et  $F^0$  s'écrit comme suit [14]:

$$H_{0,1}(t) = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

Où  $s_1 = \sin(\theta_1(t))$  et  $c_1 = \cos(\theta_1(t))$ .

Les transformations de rotation et de translation rigides (sans prendre en compte la déformation des axes) entre  $F^2$  et  $F^1$  sont représentées par la matrice :

$$\hat{H}_{1,2}(t) = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

Où  $s_2 = \sin(\theta_2(t))$  et  $c_2 = \cos(\theta_2(t))$ .

A ces déplacements rigides, nous devons rajouter la rotation et la translation dues à la déformation de l'axe 1. La déformation de l'axe 1 par rapport à la ligne neutre est décrite par la matrice de passage ci-dessous :

$$D_{1,2}(t) = \begin{bmatrix} c_{\alpha 1} & -s_{\alpha 1} & 0 & l_1 \\ s_{\alpha 1} & c_{\alpha 1} & 0 & w_1(l_1, t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.20)$$

$\alpha_i$  est l'angle de rotation de l'extrémité de l'axe  $i$  dû à la déformation.  $l_i$  est la longueur de l'axe  $i$ ,  $s_{\alpha i} = \sin(\alpha_i(t))$  et  $c_{\alpha i} = \cos(\alpha_i(t))$ . La déformation de l'axe flexible est supposée petite par rapport à sa longueur. La variation de la longueur due à la déformation est négligée.

$$\alpha_i(t) = \left. \frac{\partial w_i(x_i, t)}{\partial x_i} \right|_{x_i=l_i} \quad (2.21)$$

La matrice de transformation homogène entre les repères  $F^2$  et  $F^1$  s'écrit :

$$H_{1,2}(t) = D_{1,2}(t)\hat{H}_{1,2}(t) = \begin{bmatrix} c_{\alpha 1,2} & -s_{\alpha 1,2} & 0 & l_1 \\ s_{\alpha 1,2} & c_{\alpha 1,2} & 0 & w_1(l_1, t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.22)$$

$$H_{1,2}(t) = D_{1,2}(t)\hat{H}_{1,2}(t) = \begin{bmatrix} c_{\alpha 1,2} & -s_{\alpha 1,2} & 0 & l_1 \\ s_{\alpha 1,2} & c_{\alpha 1,2} & 0 & w_1(l_1, t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.23)$$

Avec  $s_{\alpha 1,2} = \sin(\alpha_1(t) + \theta_2(t))$  et  $c_{\alpha 1,2} = \cos(\alpha_1(t) + \theta_2(t))$ .

Nous obtenons la matrice globale de passage entre  $F^2$  et le repère  $F^0$ :

$$H_{0,2}(t) = H_{0,1}(t)H_{1,2}(t) = \begin{bmatrix} c_{1,\alpha 1,2} & -s_{1,\alpha 1,2} & 0 & c_1 l_1 - s_1 w_1(l_1, t) \\ s_{1,\alpha 1,2} & c_{1,\alpha 1,2} & 0 & s_1 l_1 + c_1 w_1(l_1, t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

Où  $s_{1,\alpha 1,2} = \sin(\theta_1(t) + \alpha_1(t) + \theta_2(t))$  et  $c_{1,\alpha 1,2} = \cos(\theta_1(t) + \alpha_1(t) + \theta_2(t))$ .

En utilisant les relations précédentes, la position et la vitesse linéaire de n'importe quel point se trouvant sur les liens du robot peuvent être calculées.

La position par rapport à  $F^1$  d'un point  $r_1$  situé sur l'axe 1 s'écrit :

$$r_1^1(t) = [x_1 \quad w_1(x_1, t) \quad 0 \quad 1]^T \quad (2.25)$$

L'expression de la position absolue de  $r_1$  dans le repère  $F^0$  est la suivante :

$$r_1^0(t) = [c_1 x_1 - s_1 w_1(x_1, t), \quad s_1 x_1 + c_1 w_1(x_1, t) \quad 0 \quad 1]^T \quad (2.26)$$

La vitesse absolue de ce point  $r_1$  dans le repère  $F^0$  :

$$v_1^0(t) = \frac{dr_1^0(t)}{dt} = \begin{bmatrix} -[s_1 x_1 + c_1 w_1(x_1, t)]\theta_1 - s_1 w_1(x_1, t) \\ [c_1 x_1 - s_1 w_1(x_1, t)]\theta_1 + c_1 w_1(x_1, t) \\ 0 \\ 0 \end{bmatrix} \quad (2.27)$$

Pour le point  $r_2$  se trouvant sur l'axe 2, la position par rapport à  $F^2$  s'écrit :

$$r_2^2(t) = [x_2 \quad w_2(x_2, t) \quad 0 \quad 1]^T \quad (2.28)$$

La position absolue de  $r_2$  dans  $F^0$  est :

$$r_2^0(t) = H_0^2 r_2^2(t) = \begin{bmatrix} l_1 c_1 - w_1(l_1, t) s_1 + x_2 c_{1,\alpha 1,2} - w_2(x_2, t) s_{1,\alpha 1,2} \\ l_1 s_1 + w_1(l_1, t) c_1 + x_2 s_{1,\alpha 1,2} + w_2(x_2, t) c_{1,\alpha 1,2} \\ 0 \\ 1 \end{bmatrix} \quad (2.29)$$

La vitesse absolue de ce point dans  $F^0$  est exprimée comme suit :

$$\begin{aligned} v_2^0(t) &= \frac{dr_2^0(t)}{dt} \\ &= \begin{bmatrix} -(l_1 s_1 + w_1(l_1, t)c_1)\dot{\theta}_1 - w_1(l_1, t)s_1 - x_2 s_{1,\alpha 1,2} - w_2(x_2, t)c_{1,\alpha 1,2} - w_2(x_2, t)s_{1,\alpha 1,2} \\ (l_1 c_1 - w_1(l_1, t)s_1)\dot{\theta}_1 + w_1(l_1, t)c_1 + x_2 c_{1,\alpha 1,2} - w_2(x_2, t)s_{1,\alpha 1,2} + w_2(x_2, t)c_{1,\alpha 1,2} \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (2.30)$$

Afin de calculer le modèle dynamique du robot manipulateur à axes flexibles, nous allons utiliser la méthode de Lagrange. Les vecteurs position,  $r_1^0$  et  $r_2^0$ , et les vecteurs vitesse  $v_1^0$  et  $v_2^0$ , serviront à calculer les expressions des l'énergie cinétique et potentielle. Les expressions qui suivent sont valables pour un manipulateur à n degrés de liberté.

L'énergie cinétique de la liaison  $i$  est exprimée par l'équation suivante [14]:

$$T_{hi} = \frac{1}{2} m_{hi} (v_i^0)^T v_i^0 \Big|_{x_i=0} + \frac{1}{2} J_{hi} \dot{\theta}_i^0 \quad (2.31)$$

Où  $m_{hi}$ ,  $J_{hi}$  et  $\theta_i^0$  sont respectivement la masse, le moment d'inertie et la position angulaire absolue de la liaison  $i$ .

$$\theta_i^0 = \sum_{j=1}^i (\theta_j + \alpha_{j-1}) \quad (2.32)$$

L'énergie cinétique de l'axe  $i$  est décrite par l'équation :

$$T_{li} = \frac{1}{2} \int_0^{l_i} \rho_i(x_i) (v_i^0)^T v_i^0 dx_i \quad (2.33)$$

L'expression de l'énergie cinétique de la charge de masse  $m_p$  et de moment d'inertie  $J_p$  localisée sur l'effecteur terminal s'écrit comme suit :

$$T_p = \frac{1}{2} m_p (\dot{v}_n^0)^T \dot{v}_n^0 \Big|_{x_n=l_n} + \frac{1}{2} J_p (\dot{\theta}_n^0 + \dot{\alpha}_n)^2 \quad (2.34)$$

L'énergie cinétique totale du robot est la somme des énergies cinétiques des différentes parties qui le composent :

$$T = \sum_{i=1}^n T_{hi} + \sum_{i=1}^n T_{li} + T_p \quad (2.35)$$

L'énergie potentielle totale du système est donnée par l'expression :

$$V = \sum_{i=1}^n \frac{1}{2} \int_0^{l_i} (EI)_i \left[ \frac{\partial^2 w_i(x_i, t)}{\partial x_i^2} \right]^2 dx_i \quad (2.36)$$

Les équations (2.35) et (2.36) sont utilisées pour former le Lagrangien  $L = T - V$ . Le modèle dynamique satisfait les équations de Lagrange :

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = f_i, \quad i = 1, \dots, n \quad (2.37)$$

Où les variables  $\{q_i(t)\}, i = 1, \dots, n$  sont les coordonnées généralisées ( $q_i = \theta_i$ ) et  $\{f_i(t)\}, i = 1, \dots, n$  sont les forces généralisées correspondantes.

En calculant l'équation de Lagrange, nous obtenons le modèle dynamique du robot manipulateur plan à  $n$  axes flexibles et  $n$  degrés de liberté rotoïdes (liaisons rigides).

Le modèle dynamique s'exprime comme suit [8] :

$$\begin{bmatrix} B_{\theta\theta}(\theta, \delta) & B_{\theta\delta}(\theta, \delta) \\ B_{\theta\delta}^T(\theta, \delta) & B_{\delta\delta}(\theta, \delta) \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} h_{\theta}(\theta, \delta, \dot{\theta}, \dot{\delta}) \\ h_{\delta}(\theta, \delta, \dot{\theta}, \dot{\delta}) \end{bmatrix} + \begin{bmatrix} 0 \\ K\delta \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix} \quad (2.38)$$

Où  $\theta$  est le vecteur  $n \times 1$  des variables articulaires,  $\delta$  le vecteur  $m \times 1$  des coordonnées modales. Le modèle dynamique du robot manipulateur flexible se compose de  $n + m$  équations différentielles non-linéaires du second ordre.

$B$  est la matrice d'inertie,  $h_{\theta}$  et  $h_{\delta}$  sont deux termes dus aux forces centrifuge et de Coriolis,  $K$  est la matrice de raideur.  $\tau$  est un vecteur de longueur  $n$  composé des couples actionneurs appliqués au niveau des articulations.

La longueur du vecteur de coordonnées modales  $\delta$  est égale à  $m$  tel que

$$m = \sum_{i=1}^n m_i.$$

$m_i$  est le nombre de coordonnées modales prises en compte pour modéliser la déformation de l'axe  $i$ .

Si les axes sont rigides,  $\delta \equiv 0$ , et l'équation (2.38) devient :

$$B_{\theta\theta}(\theta, \delta)\ddot{\theta} + h_{\theta}(\theta, \dot{\theta}) = \tau \quad (2.39)$$

## 2.4. Commande par inversion

Si le vecteur des variables articulaires  $\theta$  est défini comme sortie de notre système, la commande par inversion entrée-sortie, s'applique directement à l'équation (2.38) représentant le modèle dynamique [8, 15].

Nous allons extraire l'expression du vecteur des accélérations flexibles  $\ddot{\delta}$  de l'équation (2.38) :

$$\ddot{\delta} = -B_{\delta\delta}^{-1}[(h_{\delta} + K\delta) + B_{\theta\delta}^T\ddot{\theta}] \quad (2.40)$$

En remplaçant  $\ddot{\delta}$  par son expression dans les  $n$  (dans notre cas  $n=2$ ) premières équations du modèle (2.38), Nous obtenons :

$$\tau = (B_{\theta\theta} - B_{\theta\delta}B_{\delta\delta}^{-1}B_{\theta\delta}^T)\ddot{\theta} + h_{\theta} - B_{\theta\delta}B_{\delta\delta}^{-1}(h_{\delta} + K\delta) \quad (2.41)$$

Le degrés relatif est égal à deux pour toutes les composantes du vecteur de sortie  $\theta$ .

La matrice  $(B_{\theta\theta} - B_{\theta\delta}B_{\delta\delta}^{-1}B_{\theta\delta}^T)$  de dimension  $n \times n$  est de rang plein. L'équation (2.41) permet donc de définir complètement le vecteur d'entrée  $\tau$ .

Soit  $a$  un vecteur d'accélérations articulaires. Posons  $\ddot{\theta} = a$  dans l'équation (2.41). Définissons l'équation de la boucle de retour :

$$\tau = (B_{\theta\theta} - B_{\theta\delta}B_{\delta\delta}^{-1}B_{\theta\delta}^T)a + h_{\theta} - B_{\theta\delta}B_{\delta\delta}^{-1}n_{\delta} \quad (2.42)$$

Avec

$$n_{\delta} = h_{\delta} + K\delta \quad (2.43)$$

L'implémentation de la boucle de retour décrite par l'équation (2.42) requiert la mesure de  $\theta$ ,  $\dot{\theta}$ ,  $\delta$  et  $\dot{\delta}$ .

La commande (2.42) appliquée au robot manipulateur flexible représenté par l'équation (2.38) nous donne en boucle fermée le système linéarisé suivant :

$$\ddot{\theta} = a \quad (2.44)$$

$$\ddot{\delta} = -B_{\delta\delta}^{-1}(B_{\theta\delta}^T a + n_{\delta}) \quad (2.45)$$

Pour effectuer un suivi de trajectoire, la commande peut être complétée de la manière suivante :

$$a = \ddot{\theta}_{des} + K_D(\dot{\theta}_{des} - \dot{\theta}) + K_P(\theta_{des} - \theta) \quad (2.46)$$

Où les matrices  $K_p > 0$  et  $K_D > 0$  permettent de réaliser un placement de pôles dans le demi-plan gauche du plan complexe.

L'application de la commande linéarisante (2.42) dépend de la stabilité des dynamiques non observables introduites par l'équation (2.45). Afin d'étudier la dynamique des zéros associés au système d'équations (2.44) et (2.45), nous fixons la valeur de la sortie  $\theta = constante = \dot{\theta} = \ddot{\theta} = 0$  dans l'équation (2.42). Nous obtenons :

$$\ddot{\delta} = -B_{\delta\delta}^{-1} (h_{\delta} + K\delta) \quad (2.47)$$

Une condition suffisante pour garantir la stabilité de tout le système en boucle fermée est que la dynamique des zéros soit asymptotiquement stable.

Théorème [8] :

L'état  $\delta = \dot{\delta} = 0$  est un point d'équilibre asymptotiquement stable pour le système (2.47).

## 2.5. Conclusion

Dans ce chapitre nous avons abordé la modélisation des manipulateurs flexibles. La rotation d'un segment flexible entraîne la déformation de ce dernier. La connaissance de la coordonnée articulaire ne permet plus de calculer avec précision la position de l'extrémité du segment du robot. La déformation du segment ne peut être négligée et doit, par conséquent être modélisée.

Afin de modéliser un segment flexible, nous avons utilisé le modèle de la poutre d'Euler-Bernoulli. Ce dernier est largement utilisé dans la littérature pour modéliser les bras à chaîne ouverte simple. Les déformations élastiques peuvent être approchées par une combinaison linéaire d'un nombre fini de coordonnées modales, dont les amplitudes sont considérées comme coordonnées généralisées. Cette technique permet de traiter la déformation d'un segment flexible avec un nombre réduit de coordonnées.

La formulation des équations dynamiques s'effectue par la méthode de Lagrange. Nous aboutissons à une structure dynamique représentée par des équations différentielles ordinaires.

Nous avons vu que si le vecteur des variables articulaires est défini comme sortie de notre système, la commande par inversion entrée-sortie, s'applique directement à l'équation représentant le modèle dynamique du robot.

# Chapitre 3

---

## Commande Prédicative Généralisée

### 3.1. Introduction

Dans le présent chapitre, nous allons présenter la commande prédictive généralisée GPC. La commande GPC, proposée par Clarke et al. en 1985, est devenue l'une des commandes les plus populaires, aussi bien en industrie qu'en recherche académique [16, 17].

La commande prédictive prend en compte, à l'instant présent, le comportement futur du système à commander, en utilisant explicitement le modèle mathématique de ce dernier afin de prédire la sortie dans le futur sur un horizon fini. Cette méthode apparaît toute indiquée dans les problèmes de suivi de trajectoire et notamment en robotique [18].

L'idée de base de la GPC est de calculer une séquence de commandes futures qui minimise une fonction de coût à travers un horizon de prédiction [19]. Le critère à minimiser est la somme d'une fonction quadratique mesurant la distance entre la séquence de sorties prédites du système et une séquence de référence plus une fonction quadratique mesurant l'effort de commande.

La commande GPC sera présentée sous forme polynomiale. Cette dernière a deux principaux avantages, le premier est que les blocs qui constituent la boucle de commande sont calculés hors ligne. Le deuxième avantage est que la boucle de commande obtenue peut être analysée par les méthodes classiques de l'automatique notamment en ce qui concerne la stabilité.

### 3.2. Modélisation CARIMA

La commande prédictive repose sur le modèle. C'est le modèle du système à commander qui va nous permettre de prédire son comportement futur. La commande GPC est une commande numérique. La modélisation CARIMA sera utilisée pour représenter le modèle du système à commander.

La modélisation CARIMA permet de raisonner sur un modèle incrémental. Cette modélisation introduit un intégrateur dans la fonction de transfert afin d'annuler l'erreur statique. L'équation correspondante se présente comme suit [20] :

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\frac{\xi(t)}{\Delta(q^{-1})} \quad (3.1)$$

Avec :

$y$  la sortie du système ;

$u$  le signal de commande ;

$C(q^{-1})\xi(t)$  terme lié aux perturbations ;

$\xi$  séquence aléatoire centrée non corrélée avec l'entrée ;

$q^{-1}$  opérateur retard ;

$\Delta(q^{-1}) = 1 - q^{-1}$  est l'opérateur différence.

$A$  et  $B$  sont deux polynômes définis comme suit :

$$\begin{cases} A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_{n_a}q^{-n_a} \\ B(q^{-1}) = 1 + b_1q^{-1} + b_2q^{-2} + \dots + b_{n_b}q^{-n_b} \end{cases}$$

### 3.3. Recherche du prédicteur optimal

La présente étape est fondamentale pour la synthèse de la commande GPC. Le but est de trouver une équation de prédiction afin d'estimer le comportement futur du système à commander.

Nous recherchons l'expression de  $y(t + j/t)$  qui est l'estimation de la sortie à l'instant  $t + j$  à partir des données disponibles à l'instant  $t$ . Elle peut être décomposée de la manière suivante [16, 20] :

$$y(t + j/t) = \sum_{i=1}^j g_i \Delta u(t + j - i) + \rho_j \quad (3.2)$$

$$\rho_i = \sum_{i=j+1}^{\infty} g_i \Delta u(t + j - i) \quad (3.3)$$

Le premier terme correspond à la réponse forcée et le second à la réponse libre.

Afin d'obtenir une loi de commande polynomiale, Nous allons écrire l'expression du prédicteur sous forme polynomiale :

$$y(t+j) = \underbrace{F_j(q^{-1})y(t) + H_j(q^{-1})\Delta u(t-1)}_{\text{réponse libre}} + \underbrace{G_j(q^{-1})\Delta u(t+j-1) + J_j(q^{-1})\xi(t+j)}_{\text{réponse forcée}} \quad (3.4)$$

Nous devons définir les polynômes  $G_j, F_j, H_j, J_j$  afin de trouver l'expression du prédicteur.

$G_j$ : est un terme représentant le futur.

$F_j, H_j$ : sont deux termes correspondant au futur et au passé.

$J_j$ : est le terme lié aux perturbations.

L'équation CARIMA combinée avec l'équation polynomiale du prédicteur donne le système d'équations suivant :

$$\begin{cases} A(q^{-1})\Delta(q^{-1})y(t+j) = B(q^{-1})\Delta u(t+j-1) + \xi(t+j) \\ [1 - q^{-j}F_j(q^{-1})]y(t+j) = [G_j(q^{-1}) + q^{-j}H_j(q^{-1})]\Delta u(t+j-1) + J_j(q^{-1})\xi(t+j) \end{cases}$$

L'équivalence entre les deux équations aboutit aux équations diophantiennes suivantes :

$$\Delta(q^{-1})A(q^{-1})J_j(q^{-1}) + q^{-j}F_j(q^{-1}) = 1 \quad (3.6)$$

$$G_j(q^{-1}) + q^{-j}H_j(q^{-1}) = B(q^{-1})J_j(q^{-1}) \quad (3.7)$$

La résolution de ce système d'équations diophantiennes donne de façon explicite et unique les polynômes  $F_j(q^{-1}), G_j(q^{-1}), H_j(q^{-1})$  et  $J_j(q^{-1})$ .

Plusieurs ouvrages abordent les méthodes de résolution des équations diophantiennes. Citons à titre d'exemple [21, 22].

En supposant que la meilleure prédiction pour le terme lié aux perturbations est nulle, le prédicteur optimal s'écrit [16, 20] :

$$\begin{aligned}\hat{y}(t+j) &= F_j(q^{-1})y(t) + G_j(q^{-1})\Delta u(t+j-1) \\ &\quad + H_j(q^{-1})\Delta u(t-1)\end{aligned}\tag{3.8}$$

Avec :

$$\deg[J_j(q^{-1})] = j - 1;$$

$$\deg[G_j(q^{-1})] = j - 1;$$

$$\deg[F_j(q^{-1})] = \deg[A(q^{-1})];$$

$$\deg[H_j(q^{-1})] = \deg[B(q^{-1})] - 1.$$

Le prédicteur optimal est nécessaire au calcul du critère de commande GPC que nous allons décrire au paragraphe suivant.

### 3.4. Critère de commande

La commande GPC repose sur la minimisation d'un critère quadratique de la forme suivante [16, 20] :

$$J = \sum_{j=N_1}^{N_2} (w(t+j) - \hat{y}(t+j))^2 + \lambda \sum_{j=1}^{N_u} \Delta u^2(t+j-1)\tag{3.9}$$

Avec :

$w(t)$  la consigne à l'instant  $t$ .

$N_1$  horizon d'initialisation.

$N_2$  horizon de prédiction.

$N_u$  horizon de commande.

$\lambda$  pondération sur les incréments de commande.

Nous supposons que  $\Delta u(t+j) = 0$  pour  $j \geq N_u$ .

Dans la suite du chapitre, nous allons utiliser la notation matricielle suivante afin de simplifier l'écriture des équations :

$$if = [F_{N_1}(q^{-1}), \dots, F_{N_2}(q^{-1})]^T \quad (3.10)$$

$$ih = [H_{N_1}(q^{-1}), \dots, H_{N_2}(q^{-1})]^T \quad (3.11)$$

$$\tilde{u} = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u-1)]^T \quad (3.12)$$

$$\hat{y} = [\hat{y}(t+N_1), \dots, \hat{y}(t+N_2)]^T \quad (3.13)$$

Avec :

$$\dim(if) = (N_2 - N_1 + 1) \times (n_a + 1);$$

$$\dim(ih) = (N_2 - N_1 + 1) \times n_b;$$

$$\dim(\tilde{u}) = N_u \times 1;$$

$$\dim(\hat{y}) = (N_2 - N_1 + 1) \times 1.$$

Avec cette nouvelle notation, le prédicteur et le critère de commande s'écrivent comme suit :

$$\hat{y} = G\tilde{u} + ify(t) + ih\Delta u(t-1) \quad (3.14)$$

$$J = [G\tilde{u} + ify(t) + ih\Delta u(t-1) - W]^T [G\tilde{u} + ify(t) + ih\Delta u(t-1) - W] + \lambda \tilde{u}^T \tilde{u} \quad (3.15)$$

Avec :

$$W = [w(t+N_1) \dots w(t+N_2)]^T;$$

$$G = \begin{bmatrix} g_{N_1}^{N_1} & g_{N_1-1}^{N_1} & \dots & \dots \\ g_{N_1+1}^{N_1+1} & g_{N_1}^{N_1+1} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ g_{N_2}^{N_2} & g_{N_2-1}^{N_2} & \dots & g_{N_2-N_u+1}^{N_2} \end{bmatrix}$$

Les termes  $\{g_i^j\}$  sont les coefficients des polynômes  $G_j$ .

Il faut à présent minimiser le critère de commande  $J$  afin d'obtenir la loi de commande GPC.

### 3.5. Loi de commande

Dans le but de trouver l'expression du signal de commande, nous minimisons le critère quadratique  $J$  décrit au paragraphe précédent. L'annulation de la dérivée  $\frac{\partial J}{\partial \tilde{u}} \equiv 0$ , nous donne l'expression du vecteur  $\tilde{u}_{opt}$  [16, 20] :

$$\tilde{u}_{opt} = -M[ify(t) + ih\Delta u(t-1) - W] \quad (3.16)$$

Où

$$M = [G^T G + \lambda I_{N_u}]^{-1} G^T = \begin{bmatrix} m_1 \\ \vdots \\ m_{N_u} \end{bmatrix}$$

Dans le cadre de la commande GPC, seule le premier élément du vecteur  $\tilde{u}_{opt}$  est appliqué au système commandé :

$$u_{opt}(t) = u_{opt}(t-1) - m_1[ify(t) + ih\Delta u(t-1) - W] \quad (3.17)$$

$$\dim(m_1) = 1 \times (N_2 - N_1 + 1)$$

### 3.6. Régulateur polynomial RST équivalent

Nous souhaitons écrire la loi de commande GPC sous forme d'une boucle de commande RST. Le but du présent paragraphe est de trouver les expressions des polynômes  $R, S$  et  $T$ . Pour cela, nous allons procéder par identification en comparant la loi de commande GPC et l'équation de la commande RST [23-25].

Réécrivons l'expression de la loi de commande GPC trouvée précédemment afin de procéder à l'identification :

$$\Delta u_{opt}(t) = -m_1 ify(t) - m_1 ih\Delta u(t-1) + m_1 W \quad (3.18)$$

$$\Delta u_{opt}(t)[1 + m_1 i h q^{-1}] = -m_1 i f y(t) + m_1 [q^{N_1} \dots q^{N_2}]^T w(t) \quad (3.19)$$

Nous allons comparer cette dernière expression à l'équation de la commande RST :

$$S(q^{-1})\Delta(q^{-1})u(t) = -R(q^{-1})y(t) + T(q^{-1})w(t) \quad (3.20)$$

L'identification aboutit aux égalités suivantes :

$$S(q^{-1}) = 1 + m_1 i h q^{-1} \quad (3.21)$$

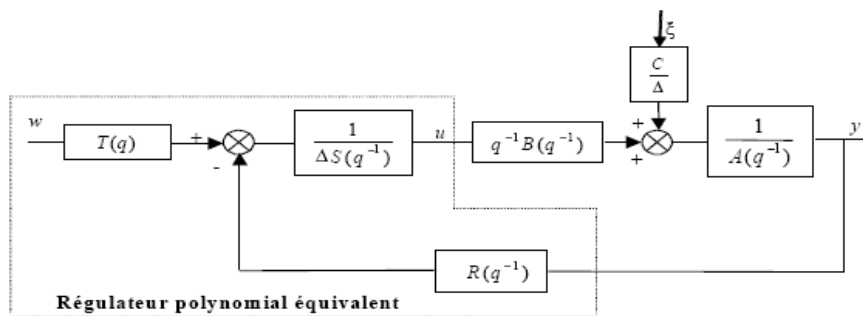
$$R(q^{-1}) = m_1 i f \quad (3.22)$$

$$T(q^{-1}) = m_1 [q^{N_1} \dots q^{N_2}]^T \quad (3.23)$$

Les polynômes  $R, S$  et  $T$  peuvent être calculés hors ligne.

Le polynôme  $T$  est non causal, ceci est caractéristique de la commande GPC. C'est ce polynôme qui va produire l'effet anticipatif de la commande prédictive.

Nous avons écrit notre loi de commande GPC sous forme d'une boucle de commande RST. Ceci va nous permettre d'étudier la stabilité de notre commande GPC avant de l'implémenter.



**Figure 3.1. Régulateur polynomial RST équivalent**

### 3.7. Paramètres de synthèse de la commande GPC

L'horizon d'initialisation  $N_1$ , l'horizon de prédiction  $N_2$ , l'horizon de commande  $N_u$  et la pondération sur les incréments de commande  $\lambda$ , sont appelés paramètres de synthèse de la commande GPC. Ces paramètres sont présents dans le critère  $J$  de la commande :

$$J = \sum_{j=N_1}^{N_2} (w(t+j) - \hat{y}(t+j))^2 + \lambda \sum_{j=1}^{N_u} \Delta u^2(t+j-1) \quad (3.24)$$

L'un des problèmes liés à l'implémentation de la commande GPC est le réglage des paramètres de synthèse. Le choix de ces paramètres détermine le comportement du système en boucle fermée.

Il n'existe pas de règles analytiques qui permettent de définir les paramètres de synthèse de la commande GPC en fonction du système commandé, des performances désirées, de la stabilité de la boucle de commande...etc.

Plusieurs études théoriques et expérimentales se sont intéressées à ce problème et ont permis d'arriver à des indications permettant de guider le concepteur dans le choix des paramètres de la commande GPC.

Le concepteur peut également effectuer des essais sur son propre système. En variant un à un les paramètres de la commande GPC, des relations qualitatives entre ces derniers et les différentes performances de la commande GPC peut être trouvées.

### 3.7.1. Horizon d'initialisation « $N_1$ »

L'horizon d'initialisation  $N_1$  doit appartenir à l'intervalle suivant [26] :

$$d \leq N_1 \leq n_B + d \quad (3.25)$$

Le retard du système à commander peut être choisi comme horizon d'initialisation [20] :

$$N_1 = d \quad (3.26)$$

Plus  $N_1$  est petit, plus la commande est rapide.

Choisir  $N_1$  inférieur à  $d$  n'a aucun intérêt. Cela introduit à des calculs superflus puisque la sortie à l'instant  $t + i$  avec  $i < d$  ne peut être affectée par la commande à l'instant  $t$ .

**3.7.2. Horizon de prédiction «HP =  $N_2$ »**

Il convient de choisir le domaine de variation de l'horizon de prédiction conformément à la règle suivante [26] :

$$(N_2)_{min} \leq N_2 \leq (N_2)_{max} \tag{3.27}$$

Avec :

$$(N_2)_{min} = 2Sup\{n_B + d; n_A + 1\}; \tag{3.28}$$

$$(N_2)_{max} = 2Sup\{n_B + d; n_A + 1\} + 10. \tag{3.29}$$

Le choix de la valeur de  $N_2$  à l'intérieur de son domaine de variation sera déterminé par les performances souhaitées pour le système en boucle fermée. En augmentant  $N_2$  la commande devient plus agressive, les variations du signal de commande sont plus importantes et la réponse de la boucle de commande plus rapide. Cela conduit à un dépassement plus important et peut conduire à des oscillations dans la réponse du système commandé. La stabilité de la boucle de commande s'en trouve ainsi fragilisée.

Nous pouvons voir sur le tableau suivant, l'influence de  $N_2$  sur quelques performances de la boucle fermée.

Variation $N_2$	Temps de montée	Degré de Stabilité	Variance U	Variance Y
>0	AUGMENTE	AUGMENTE	DIMINUE	DIMINUE
<0	DIMINUE	DIMINUE	AUGMENTE	AUGMENTE

**Tableau 3.1. Influence de  $N_2$  sur quelques performances de la commande en boucle fermée [26].**

**3.7.3. Horizon de commande « $N_u$ »**

Le paramètre  $N_u$  a une faible influence sur la marge de gain et la marge de phase [19, 20]. Par contre  $N_u$  influence significativement sur la charge de calcul :  $N_u$  fixe les dimensions des matrices à inverser. Augmenter  $N_u$  accroît rapidement la charge de calcul. En général, une valeur de  $N_u = 1$  est jugée adéquate pour les systèmes utilisés fréquemment en industrie [26].

### 3.7.4. Pondération sur les incréments de commande « $\lambda$ »

Il existe une valeur optimale pour  $\lambda$  notée  $\lambda_{opt}$  procurant une marge de phase maximale et une bonne marge de gain. Elle est définie par [26] :

$$\lambda_{opt} = trace(G^T G) \quad (3.30)$$

Cette relation fondamentale permet de réduire considérablement la plage de recherche des jeux de paramètres stables autour des valeurs de  $\lambda$  « admissibles ».

$G$  est la matrice formée des coefficients de la réponse indicielle.

Le choix de  $\lambda$  peut se faire à l'intérieur de l'intervalle suivant [26] :

$$\lambda = k\lambda_{opt}, \quad (3.31)$$

$$k \in [0.1, 10]. \quad (3.32)$$

Dans le tableau qui suit, nous pouvons voir l'influence de la variation de  $\lambda$  sur les performances du système contrôlé :

Variation $\lambda$	Temps de montée	Degré de Stabilité	Variance U	Variance Y
>0	AUGMENTE	AUGMENTE	DIMINUE	DIMINUE
<0	DIMINUE	DIMINUE	AUGMENTE	AUGMENTE

**Tableau 3.2. Influence de  $\lambda$  sur les performances du système en boucle fermée[26]**

L'augmentation de  $\lambda$  rend la commande plus lisse (et plus lente) et plus stable.

### 3.8. Résultats et simulations

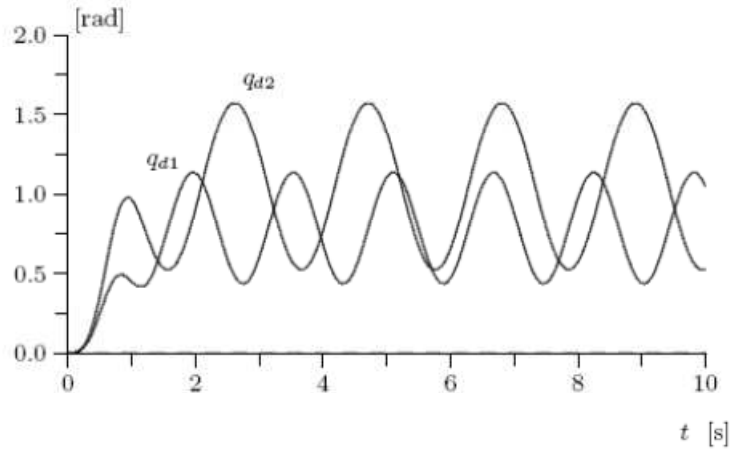
Dans le but de tester les performances des commandes étudiées. Nous avons choisi de travailler dans l'espace articulaire. Les trajectoires de référence suivantes ont été choisies :

$$\begin{bmatrix} q_{d1} \\ q_{d2} \end{bmatrix} = \begin{bmatrix} b_1 [1 - e^{-2t^3}] + c_1 [1 - e^{-2t^3}] \sin(\omega_1 t) \\ b_2 [1 - e^{-2t^3}] + c_2 [1 - e^{-2t^3}] \sin(\omega_2 t) \end{bmatrix} [rad] \quad (3.33)$$

$$b_1 = \pi/4 [rad], c_1 = \pi/9 [rad], b_2 = \pi/3 [rad], c_2 = \pi/6 [rad], \omega_1 = 4 [rad/s],$$

$\omega_2 = 3 [rad/s]$ , sont les paramètres de la trajectoire de référence.

Ces trajectoires sont proposées dans[1].



**Figure 3-2: Trajectoires de références dans l'espace articulaire.**

Notons quelques points importants concernant la trajectoire de référence choisie :

- La trajectoire de référence contient un terme sinusoïdal. Ce terme sert à évaluer la performance de la commande durant des mouvements périodiques relativement rapides. Ceci est important puisque ce type de mouvements excite les non linéarités du système.
- Elle contient également un terme croissant qui amène lentement le robot vers son régime définitif sans saturer les actionneurs.

L'amplitude et la fréquence du signal périodique doivent être choisies avec soin afin d'éviter toute saturation en couple ou en vitesse des actionneurs.

En dérivant les expressions des trajectoires de référence, on trouve les équations suivantes exprimant les vitesses articulaires en fonction du temps :

$$\begin{aligned} \dot{q}_{d1} = & 6b_1t^2e^{-2t^3} + 6c_1t^2e^{-2t^3} \sin(\omega_1t) \\ & + [c_1 - c_1e^{-2t^3}] \cos(\omega_1t) \omega_1 [rad/s] \end{aligned} \quad (3.34)$$

$$\begin{aligned} \dot{q}_{d2} = & 6b_2t^2e^{-2t^3} + 6c_2t^2e^{-2t^3} \sin(\omega_2t) \\ & + [c_2 - c_2e^{-2t^3}] \cos(\omega_2t) \omega_2 [rad/s] \end{aligned} \quad (3.35)$$

On procède de même pour calculer les accélérations articulaires correspondant à la trajectoire de référence :

$$\begin{aligned} \ddot{q}_{d1} = & 12b_1te^{-2t^3} - 36b_1t^4e^{-2t^3} + 12c_1te^{-2t^3} \sin(\omega_1t) \\ & - 36c_1t^4e^{-2t^3} \sin(\omega_1t) + 12c_1t^2e^{-2t^3} \cos(\omega_1t) \omega_1 \\ & - [c_1 - c_1e^{-2t^3}] \sin(\omega_1t) \omega_1^2 [rad/s^2], \end{aligned} \quad (3.36)$$

$$\begin{aligned} \ddot{q}_{d2} = & 12b_2te^{-2t^3} - 36b_2t^4e^{-2t^3} + 12c_2te^{-2t^3} \sin(\omega_2t) \\ & - 36c_2t^4e^{-2t^3} \sin(\omega_2t) \\ & + 12c_2t^2e^{-2t^3} \cos(\omega_2t) \omega_2 \\ & - [c_2 - c_2e^{-2t^3}] \sin(\omega_2t) \omega_2^2 [rad/s^2] \end{aligned} \quad (3.37)$$

Afin de pouvoir juger des performances de la boucle de commande GPC. Nous avons effectué plusieurs simulations en utilisant le modèle du robot manipulateur rigide plan à deux degrés de liberté rotoïdes présenté dans le *chapitre 1*.

Le tableau qui suit regroupe les valeurs des paramètres du robot manipulateur rigide. Ces valeurs seront supposées être les valeurs réelles au cours de la simulation[27].

Description	Valeur	Unité
Longueur de l'axe 1	$l_1 = 0.432$	m
Longueur de l'axe 2	$l_2 = 0.432$	m
Masse de l'axe 1	$m_1 = 15.91$	Kg
Masse de l'axe 2	$m_2 = 11.36$	kg
Inertie relative au centre de masses (axe	$I_1 = 0.248$	Kg.m <sup>2</sup>
Inertie relative au centre de masses (axe	$I_2 = 0.18$	Kg.m <sup>2</sup>
Accélération de la gravité	$g = 9.81$	m/s <sup>2</sup>

**Tableau 3.3. Paramètres du robot manipulateur rigide**

Comme nous avons pu le voir dans le présent chapitre, la commande GPC est une commande linéaire. Un robot manipulateur rigide est un système non linéaire. Pour pouvoir appliquer la commande GPC, le modèle du manipulateur a été linéarisé suivant la méthode présentée au *chapitre 1*. Le modèle linéarisé se présente sous forme de deux doubles intégrateurs indépendants entre eux. Le premier ayant pour sortie la coordonnée articulaire  $q_1$

et le second  $q_2$ . Chaque articulation sera commandée à l'aide d'une boucle de commande GPC mono-entrée mono-sortie indépendante.

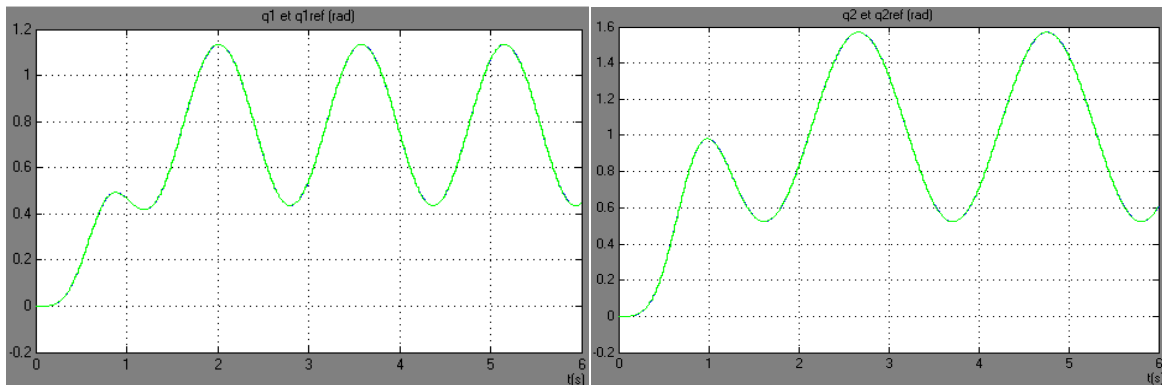
Pour la boucle GPC1, les paramètres suivants ont été choisis :  $N_1 = 1, N_2 = 8, N_u = 1, \lambda = \lambda_{opt} = 2.193 \times 10^{-5}$ . L'algorithme de commande GPC RST, nous donne les polynômes  $R, S$  et  $T$  correspondants :

$$\begin{aligned} R(z^{-1}) &= 10^4(0.7449 - 1.2955 z^{-1} + 0.5739 z^{-2}). \\ \Delta S(z^{-1}) &= 1 - 0.713 z^{-1} - 0.2869 z^{-2}. \\ T(z^{-1}) &= 72.9594 z^8 + 55.8595 z^7 + 41.03967 z^6 + 28.4997 z^5 + 18.2398 z^4 \\ &\quad + 10.2599 z^3 + 4.5599 z^2 + 1.1399 z. \end{aligned}$$

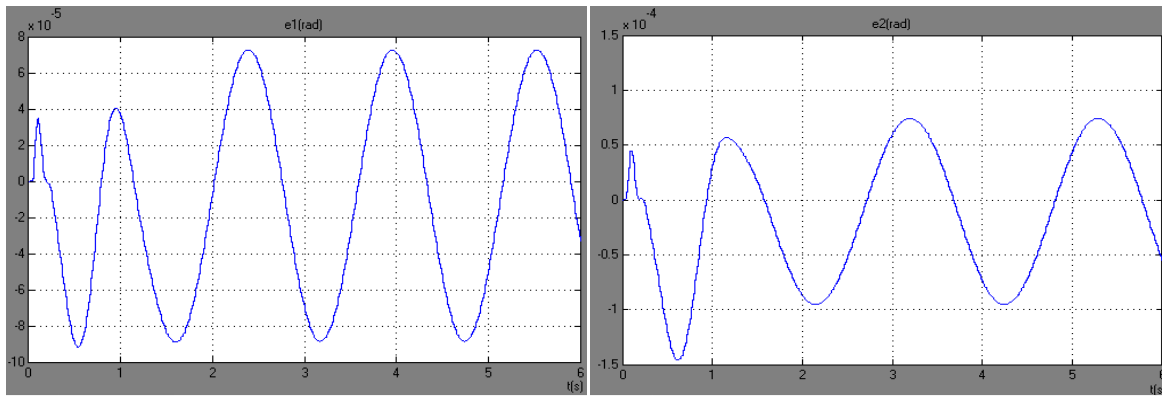
Pour la boucle GPC2 :  $N_1 = 1, N_2 = 15, N_u = 1, \lambda = \lambda_{opt} = 4.457 \times 10^{-4}$ . Les polynômes  $R, S$  et  $T$  correspondants sont :

$$\begin{aligned} R(z^{-1}) &= 10^4(0.6281 - 1.1615 z^{-1} + 0.5404 z^{-2}). \\ \Delta S(z^{-1}) &= 1 - 0.7298 z^{-1} - 0.2702 z^{-2}. \\ T(z^{-1}) &= 12.61 z^{15} + 10.99 z^{14} + 9.48 z^{13} + 8.0757 z^{12} + 6.7858 z^{11} + 5.6081 z^{10} \\ &\quad + 4.5426 z^9 + 3.5892 z^8 + 2.7479 z^7 + 2.0189 z^6 + 1.402 z^5 + 0.8973 z^4 \\ &\quad + 0.5047 z^3 + 0.2243 z^2 + 0.5061 z. \end{aligned}$$

Nous pouvons voir sur les figures suivantes la variation des variables articulaires  $q_1$  et  $q_2$  ainsi que l'erreur de poursuite pour le jeu de paramètres choisi.



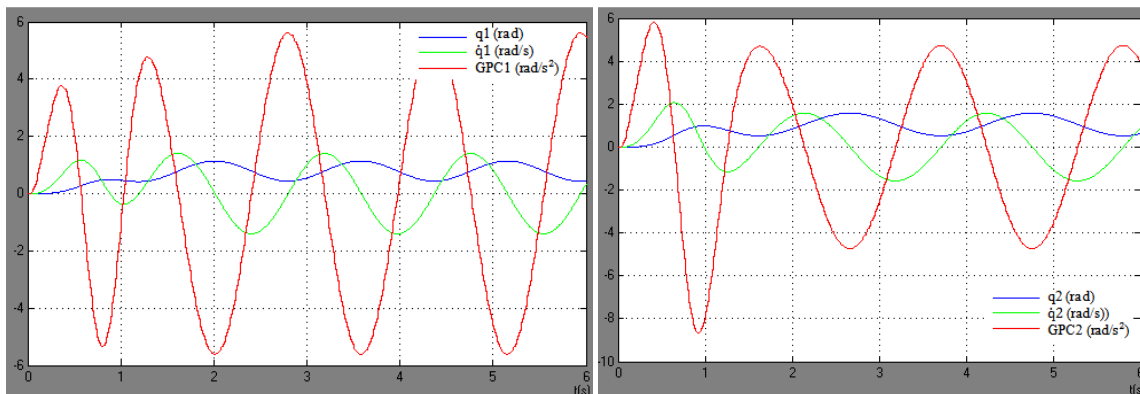
**Figure 3.4. Suivi de trajectoire pour  $q_1$  et  $q_2$**



**Figure 3.5. Erreurs de poursuite lors du suivi de trajectoire  $e_1$  et  $e_2$**

Nous pouvons constater que les coordonnées articulaires  $q_1$  et  $q_2$  suivent de près leurs trajectoires de référence respectives sur le suivi d’une trajectoire sinusoïdale.

Nous présentons ci-après les figures représentant sur une même fenêtre la position articulaire  $q_i$ , la vitesse articulaire  $\dot{q}_i$ , ainsi que le signal de la commande GPC noté  $GPC_i$  et correspondant selon le méthode de linéarisation à l’accélération articulaire  $\ddot{q}_i$ . Le paramètre  $i$  ( $i = 1,2.$ ) correspond à la première ou à la deuxième articulation.

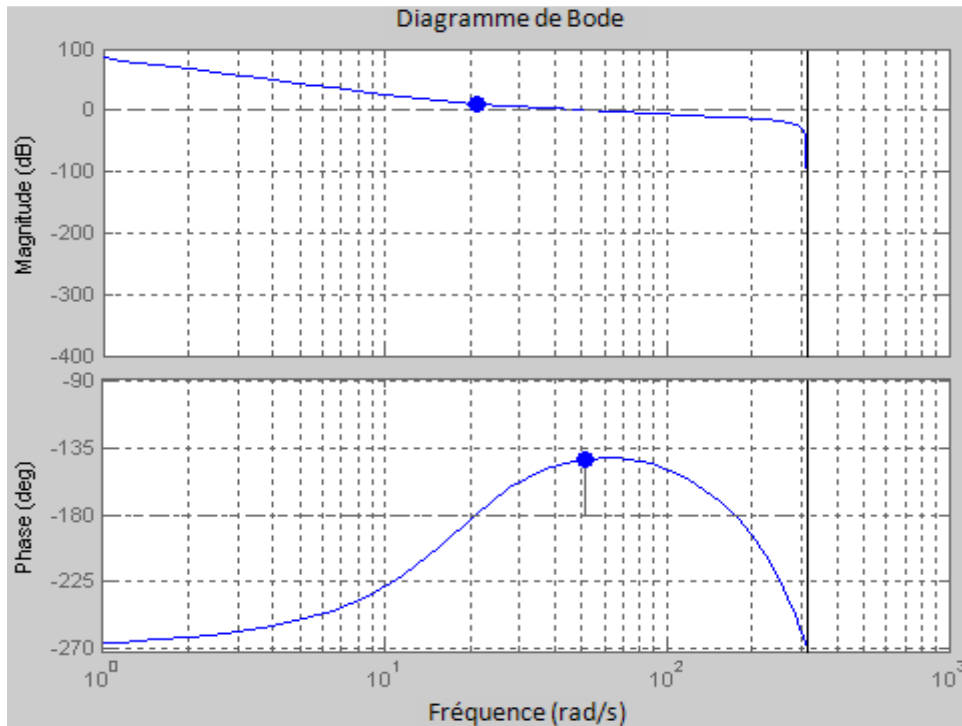


**Figure 3.7. Position articulaire  $q_i$ , vitesse articulaire  $\dot{q}_i$ , commande GPC lors du suivi de trajectoire.**

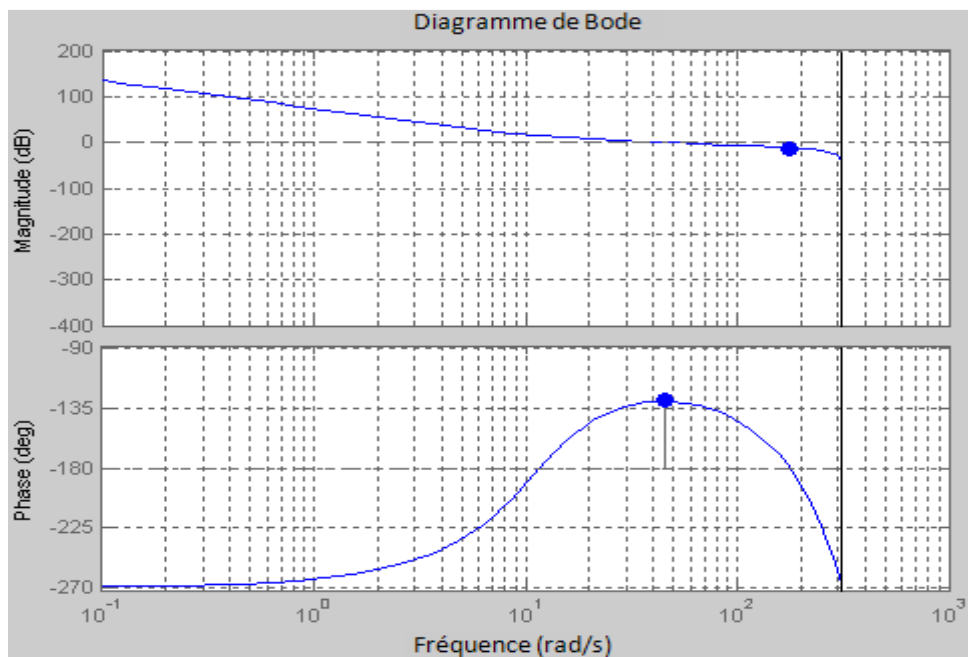
Nous pouvons voir sur les différentes figures que pour chaque articulation le signal de commande GPC coïncide bien avec l’accélération articulaire.

Nous allons nous intéresser maintenant à la stabilité de la boucle de commande. Nous avons vu dans la partie théorique de ce chapitre que l’un des principaux avantages de la commande GPC polynomiale est la possibilité d’étudier la stabilité de la boucle de commande avant de l’implémenter et de pouvoir pour cela utiliser les méthodes fréquentielles.

Nous allons dans les figure suivantes, présenter les diagrammes de Bode correspondant aux boucles de commande GPC1 et GPC2.



**Figure 3.8. Diagramme de Bode de la boucle GPC1**



**Figure 3.9. Diagramme de Bode de la boucle GPC2**

Les deux boucles de commandes sont stables. La première présente une marge de gain = -9.61 dB et une marge de phase = 37.16 deg. La seconde présente une marge de gain =12.39dB et une marge de phase = 50.38 deg.

### **3.9. Conclusion**

Ce chapitre a été consacré à la présentation de la commande prédictive généralisée GPC. Nous avons choisi d'utiliser la commande GPC polynomiale. Cette dernière offre l'avantage d'être calculée hors ligne pour un jeu donné des paramètres de synthèse. L'un de ses principaux avantages reste la possibilité d'étudier la stabilité de la boucle de commande obtenue en utilisant les méthodes fréquentielles.

Nous avons abordé le problème du réglage des paramètres de synthèse de la commande GPC en présentant quelques relations qualitatives reliant ces derniers aux performances en boucle fermée.

La commande GPC RST a été testée en simulation sur le modèle du robot manipulateur rigide plan à 2 degrés de liberté rotoïdes. Des résultats satisfaisants ont pu être obtenus pour un suivi de trajectoire.

La stabilité de la boucle de commande a pu être vérifiée en utilisant le diagramme de Bode.

# Chapitre 4

---

## **Commande Prédicative Généralisée GPC Supervisée par la Logique Floue**

## 4.1. Introduction

Les performances de la commande GPC dépendent du choix de ses paramètres de synthèse : horizon de commande  $N_u$ , horizon d'initialisation  $N_1$ , horizon de prédiction  $N_2$  ainsi que la pondération sur les incréments de commande  $\lambda$ . Un réglage approprié de ces paramètres est nécessaire pour garantir le bon fonctionnement de la commande.

Face à un modèle peu précis ou si le système travaille en présence de perturbations, par exemple, il est délicat de choisir les paramètres de la commande GPC. Les paramètres de commande choisis hors ligne peuvent ne pas garantir les performances souhaitées lors de l'implémentation de la commande.

Ce chapitre expose une méthode de supervision capable d'ajuster ces paramètres en ligne et ceci en fonction des performances de la commande. Le but est d'atteindre les performances désirées et d'exploiter au maximum les possibilités de la commande GPC.

La supervision a pour rôle de surveiller en ligne le fonctionnement de la commande. Elle intervient en cas de dégradation des performances de celle-ci. C'est un organe de surveillance, de prise de décision et d'intervention. Les informations concernant la commande lui arrivent à partir des capteurs ou éventuellement d'un observateur d'état. Il n'a pas pour rôles de les estimer. Il recueille les informations dont il a besoin, les analyse et intervient si nécessaire en changeant les paramètres de la commande

Cette couche de supervision peut également être utilisée lors de la phase de conception, en simulation par exemple, afin de régler définitivement les paramètres de la commande GPC à implémenter.

La méthode d'ajustement proposée ici est basée sur la logique floue.

## 4.2. Étapes de la supervision floue

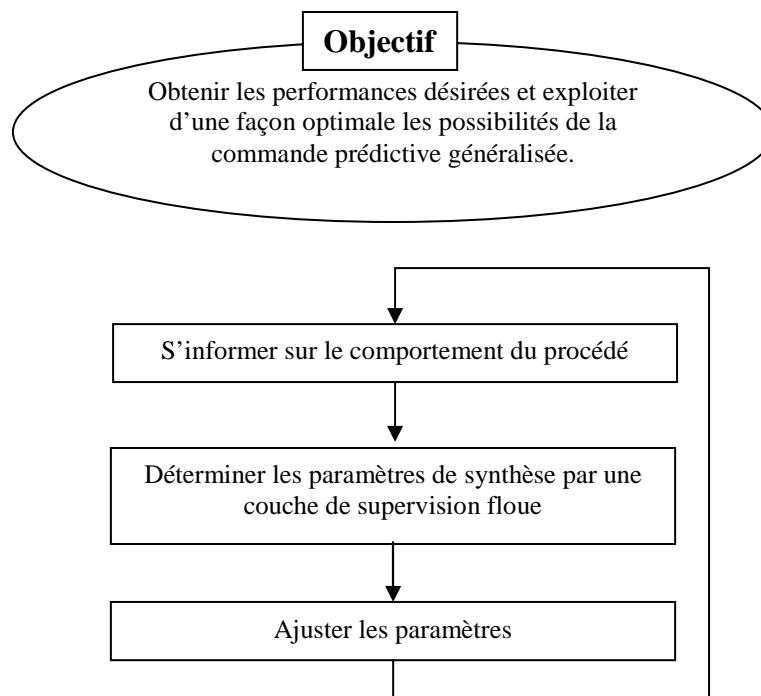
La couche floue que nous allons ajouter à la boucle de commande GPC doit être capable de superviser le travail de celle-ci. Si les performances de la commande GPC se dégradent, la couche de supervision floue doit réagir en ajustant les paramètres de synthèse de façon à améliorer la qualité de la commande.

Le superviseur flou présenté est inspiré du livre [26]. La supervision par logique floue est abordée dans plusieurs ouvrages [28, 29] et plusieurs articles de recherche [30-32].

La première étape de la supervision est de récolter des informations sur le fonctionnement de la commande et de les envoyer à la couche de supervision floue. Le rôle de cette dernière est de calculer les nouveaux paramètres de la commande GPC.

Une fois les nouveaux paramètres calculés par la couche de supervision floue, il faut déterminer la nouvelle boucle de commande GPC RST.

La figure 4.1 résume le travail du superviseur.



**Figure 4.1. Cycle de supervision floue**

#### 4.2.1. S'informer sur le comportement du système commandé

Il faut commencer par choisir les performances que nous désirons surveiller. Ces performances jugées importantes seront mesurées ou estimées tout au long du fonctionnement du système commandé.

Nous allons surveiller la rapidité de la réponse et la robustesse de la stabilité de la boucle de commande.

Nous choisissons le paramètre  $N_2$  pour agir sur la rapidité de la réponse du système commandé et le paramètre  $\lambda$  pour agir sur la robustesse de la stabilité en boucle fermée.

En augmentant la rapidité de la réponse du système, nous devons faire attention à ce que la variance du signal de commande ne devienne pas trop importante. Cette remarque est confirmée par le tableau 3.1. En effet, pour augmenter la rapidité de la réponse, il faut réduire la valeur de  $N_2$ . Ceci a pour conséquence logique une augmentation de la variance du signal de commande.

En améliorant la robustesse de la boucle de commande, nous allons également surveiller la variance du signal de commande. En effet, afin d'améliorer la robustesse de la commande il faut augmenter la valeur de  $\lambda$  (voir tableau 3.2). En parallèle, l'augmentation de  $\lambda$  aura pour effet de diminuer la variance du signal de commande. Une trop grande valeur de  $\lambda$  va étouffer la commande GPC en l'empêchant de varier suffisamment.

Les grandeurs suivantes vont nous renseigner sur le comportement de la commande :

- La rapidité (variance de la sortie « VarY ») ;
- La variance du signal de commande « VarU » ;
- La robustesse de la stabilité de la commande (le degré de stabilité « DegSta »).

#### 4.2.1.1. La variance « VarX »

La variance d'un signal nous renseigne sur le degré de fluctuation de celui-ci. Nous nous intéressons à la variance du signal de commande VarU.

Notons  $\bar{X}(k)$  la moyenne du signal  $X$  calculée sur une fenêtre de  $k$  échantillons[26].

$$\bar{X}(k) = \frac{1}{k} \left[ \sum_{i=1}^{k-1} X(i) + X(k) \right] = \frac{1}{k} [(k-1)\bar{X}(k-1) + X(k)] \quad (4.1)$$

$$\bar{X}(k) = \bar{X}(k-1) + \frac{1}{k} [X(k) - \bar{X}(k-1)] \quad (4.2)$$

$$VARX = \overline{X^2}(k) - [\bar{X}(k)]^2 \quad (4.3)$$

La variance  $VARX$  est calculée sur une fenêtre glissante de 25 échantillons ( $k=25$ ).

#### 4.2.1.2. Le degré de stabilité « DegSta »

Le degré de stabilité DegSta est défini comme étant la distance minimale séparant le lieu de la fonction de transfert en boucle ouverte du point critique Xc (-1,0)[33].

Nous commençons par définir la fonction de sensibilité :

$$SENS(j\omega) = \frac{1}{(1 + TSC(j\omega))} \quad (4.4)$$

Où TSC(z) est la fonction de transfert en boucle ouverte du système compensé avec :

$$z = \exp(j\omega T_e).$$

DegSta sera exprimé de la manière suivante :

$$DegSta = \frac{1}{\text{Sup}(\omega) ||SENS(j\omega)||} \quad (4.5)$$

$$\omega \in [0, \pi/T_e].$$

La diminution de DegSta est un signe de dégradation de la robustesse de la boucle de commande.

#### 4.2.2. Contraintes et normalisation

Pour le bon fonctionnement de la boucle de commande GPC, des contraintes seront définies sur les paramètres de synthèse ainsi que sur les performances surveillées.

Les performances de la commande sont les entrées de notre superviseur flou. Les paramètres de synthèse en sont les sorties.

Pour chaque performance mesurée ou calculée *PERX*, un intervalle [*PERXmin*, *PERXmax*] exprimant la plage de variation autorisée sera défini.

Cet intervalle sera normalisé et ramené à un intervalle [0,1]. La variable normalisée *PERXn* correspondant à la valeur *PERX* appartenant à l'intervalle [*PERXmin*, *PERXmax*] est calculée de la manière suivante [26]:

$$PERX_n = \frac{(PERX - PERX_{min})}{(PERX_{max} - PERX_{min})} \quad (4.6)$$

Chaque paramètre de synthèse  $PRY$  de la commande GPC appartient à une plage de valeurs autorisées, comprise entre  $PRY_{min}$  et  $PRY_{max}$ . Le paramètre  $PRY$  sera également normalisé[26].

$$PRY_n = \text{l'entier le plus proche de } \left[ \frac{(PRY - PRY_{min})}{(PRY_{max} - PRY_{min})} \right] \quad (4.7)$$

### 4.2.3. Stratégie de réglage flou

L'étape la plus importante dans la conception de notre superviseur flou est l'établissement des relations qui existent entre les paramètres de synthèse de la commande GPC et les performances de la boucle de commande.

Le but est d'établir des relations qualitatives. Ces relations seront traduites en règles floues qui constitueront le cœur du superviseur.

Afin d'établir ces relations, nous pouvons modifier les valeurs des paramètres de synthèse de la commande GPC un par un et en surveillant à chaque fois l'évolution des performances de la boucle de commande correspondante.

Dans le présent travail, nous utilisons les relations données dans les tableaux 4.1 et 4.2.

## 4.3. Superviseur flou

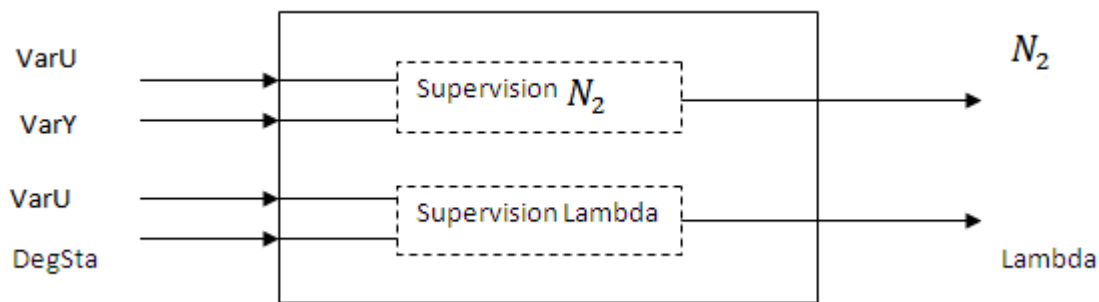
Nous allons à présent dérouler la méthode utilisée pour la supervision de la commande GPC.

Nous allons manipuler les paramètres de synthèse de la commande GPC (horizon de prédiction  $N_2$  et pondération sur les incréments de commande  $\lambda$ ) pour améliorer les performances de la commande.

Le paramètre de synthèse  $N_2$  va nous permettre d'agir sur la rapidité de la réponse du système commandé (variance de la sortie) et la variance du signal de commande.

Le paramètre  $\lambda$  quant à lui, va agir sur la robustesse de la commande ainsi que sur la variance du signal de commande.

La figure suivante est un schéma fonctionnel du superviseur flou présenté dans ce chapitre. Celui-ci se compose de deux superviseurs flous internes : le superviseur  $N_2$  et le superviseur  $\lambda$ .



**Figure 4.2. Schéma fonctionnel du superviseur flou**

#### 4.3.1. Superviseur $N_2$

Le paramètre  $N_2$  va nous permettre d'agir sur la rapidité du système commandé et sur la variance du signal de commande.

Si nous diminuons  $N_2$ , cela induira augmentation de la rapidité du système en même temps qu'une augmentation de la variance du signal de commande  $VarU$  (voir tableau 3.1.). Il faudra donc régler  $N_2$  de manière à assurer un compromis entre ces deux performances.

La première étape est de normaliser les indices de performance mesurés :  $VarY$  devient  $VarY_n$  et  $VarU$  devient  $VarU_n$ .

Nous allons associer  $2N+1$  ensembles flous à chaque entrée de supervision (indices de performances  $VarY_n$  et  $VarU_n$ ) et  $4N+1$  ensembles flous pour la sortie du superviseur  $N_2$ .

En prenant  $N=5$ . Nous obtenons  $2N+1=11$  ensembles flous pour chaque entrée normalisée et  $4N+1=21$  ensembles flous pour la sortie  $N_2$ . La sortie du superviseur est obtenue sous forme normalisée.

Le nombre d'ensembles flous a été choisi au cours de la simulation, en effectuant plusieurs essais avec des valeurs de  $N$  différentes.

Pour chaque variable d'entrée, les attributs des ensembles flous vont de très très petit (TTP) à très très grand (TTG). Pour la sortie, ils varient de très très très petit (TTTP) à très très très grand (TTTG).

Des fonctions d'appartenance triangulaires définies sur l'intervalle  $[0,1]$  ont été utilisées pour les variables d'entrée et la variable de sortie de notre superviseur  $N_2$  (voir figure 4.3).

Le choix de la forme des fonctions d'appartenance est une difficulté inhérente à la conception d'un système flou. L'une des approches les plus employées par les chercheurs est l'utilisation de fonctions d'appartenance triangulaires uniformément espacées.

La forme triangulaire permet une implémentation facile et requiert peu de temps de calcul lors de l'évaluation en temps réel. Elle consiste en un premier choix qui peut être modifié si besoin lors de la phase de conception. Libre au concepteur du système de choisir une fonction d'appartenance plus complexe s'il estime que le gain de performance est non négligeable.

Des études comparatives ont montré qu'avec les différentes formes des fonctions d'appartenance, les résultats sont pratiquement similaires en boucle fermée. La forme la plus fréquemment utilisée en commande floue est la forme triangulaire.

Dans le présent cas, le choix des fonctions d'appartenance triangulaires a donné des résultats satisfaisants en terme de commande et de temps de calcul. Le temps de calcul enregistré étant déjà assez important, il n'est pas justifié de passer à des fonctions plus complexes.

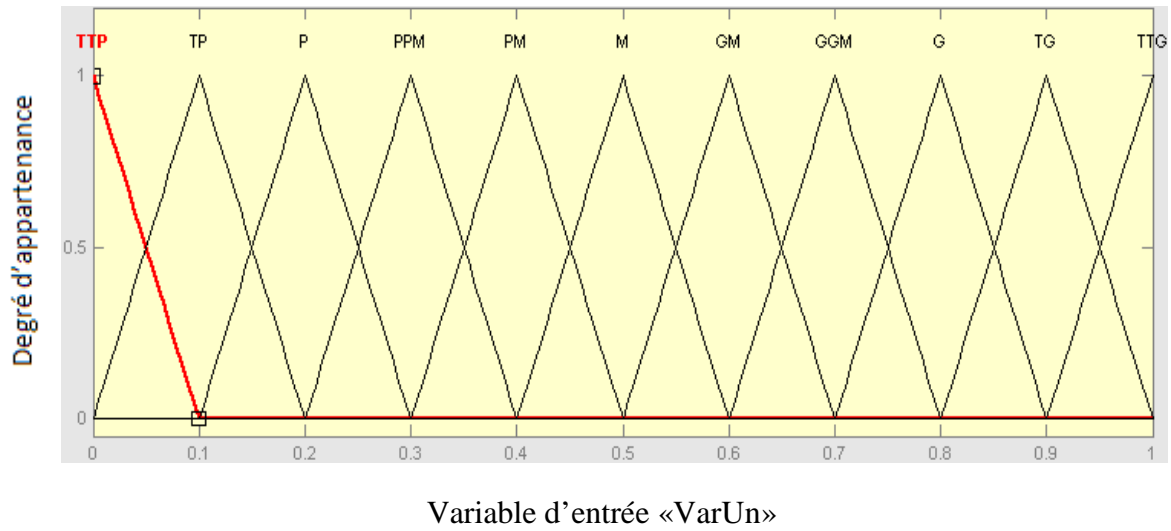


Figure 4.3.a. Fonctions d'appartenance relatives à la variable d'entrée «VarUn»

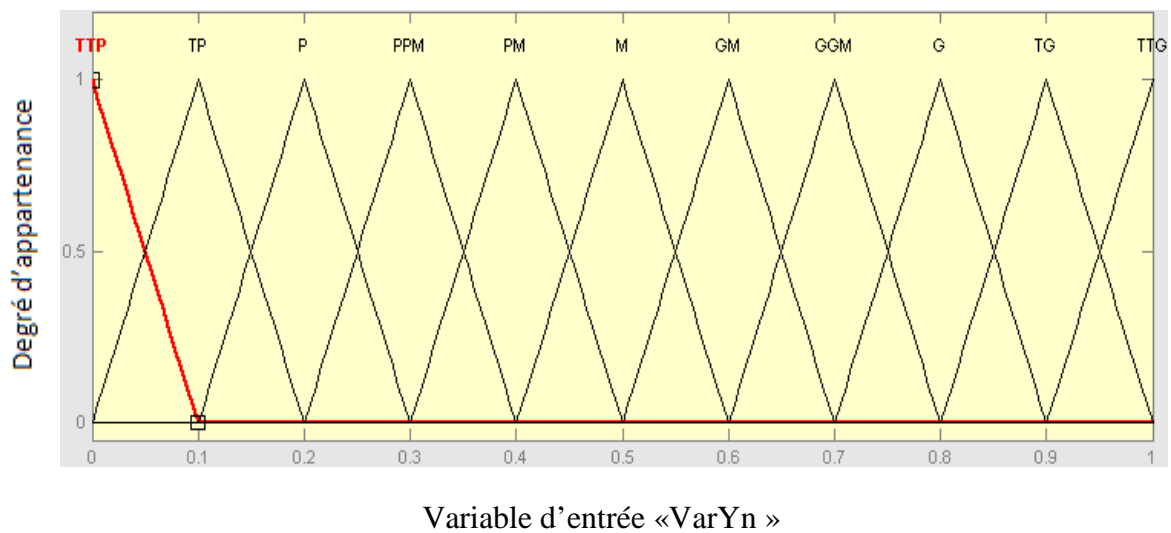


Figure 4.3.b. Fonctions d'appartenance relatives à la variable d'entrée «VarYn»

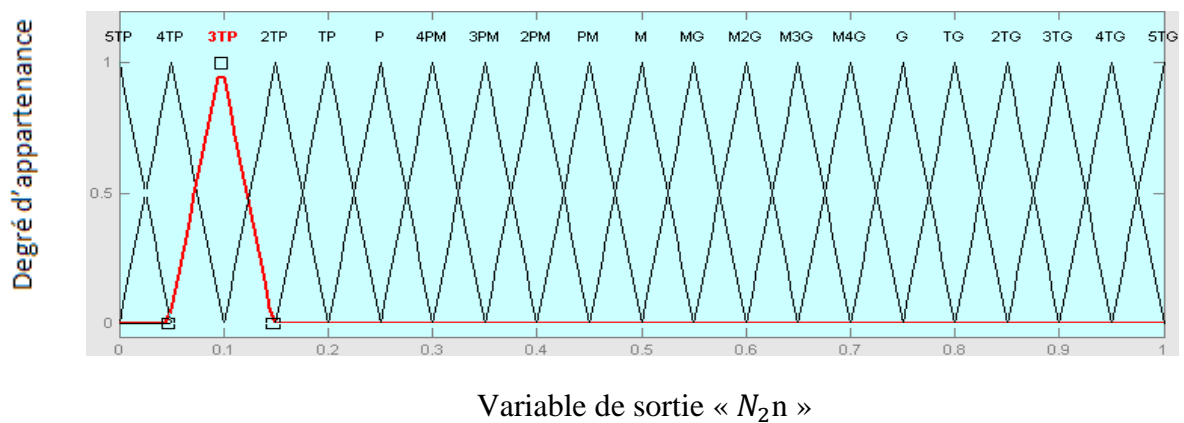


Figure 4.3.c. Fonctions d'appartenance relatives à la variable de sortie « $N_{2n}$ »

Les règles floues qui relient le paramètre de synthèse normalisé  $N_2n$  aux indices de performance VarYn et VarUn sont données par le tableau 4.1.

La méthode d'inférence MAX-MIN a été appliquée afin de calculer les fonctions d'appartenance aux règles floues.

Pour déterminer la sortie normalisée du superviseur  $N_2n$ , nous avons utilisé la méthode du barycentre. La valeur finale de  $N_2$  est égale à :

$$N_2 = \text{L'entier le plus proche de } [N_2n(N_2max - N_2min) + N_2min] \tag{4.8}$$

		VarYn										
		TTG	TG	G	MG	2MG	M	P2M	PM	P	TP	TTP
V a r U n	TTG	5TP	4TP	3TP	2TP	TP	P	3PM	2PM	P2M	P3M	<b>M</b>
	TG	4TP	3TP	2TP	TP	P	3PM	2PM	P2M	P3M	<b>M</b>	3MG
	G	3TP	2TP	TP	P	3PM	2PM	P2M	P3M	<b>M</b>	3MG	2MG
	MG	2TP	TP	P	3PM	2PM	P2M	P3M	<b>M</b>	3MG	2MG	M2G
	2MG	TP	P	3PM	2PM	P2M	P3M	<b>M</b>	3MG	2MG	M2G	M3G
	M	P	3PM	2PM	P2M	P3M	<b>M</b>	3MG	2MG	M2G	M3G	G
	P2M	3PM	2PM	P2M	P3M	<b>M</b>	3MG	2MG	M2G	M3G	G	TG
	PM	2PM	P2M	P3M	<b>M</b>	3MG	2MG	M2G	M3G	G	TG	2TG
	P	P2M	P3M	<b>M</b>	3MG	2MG	M2G	M3G	G	TG	2TG	3TG
	TP	P3M	<b>M</b>	3MG	2MG	M2G	M3G	G	TG	2TG	3TG	4TG
	TTP	<b>M</b>	3MG	2MG	M2G	M3G	G	TG	2TG	3TG	4TG	5TG

**Tableau 4.1. Table des règles floues reliant  $N_2n$  aux indices de performance VarYn et VarUn.**

### 4.3.2. Superviseur $\lambda$

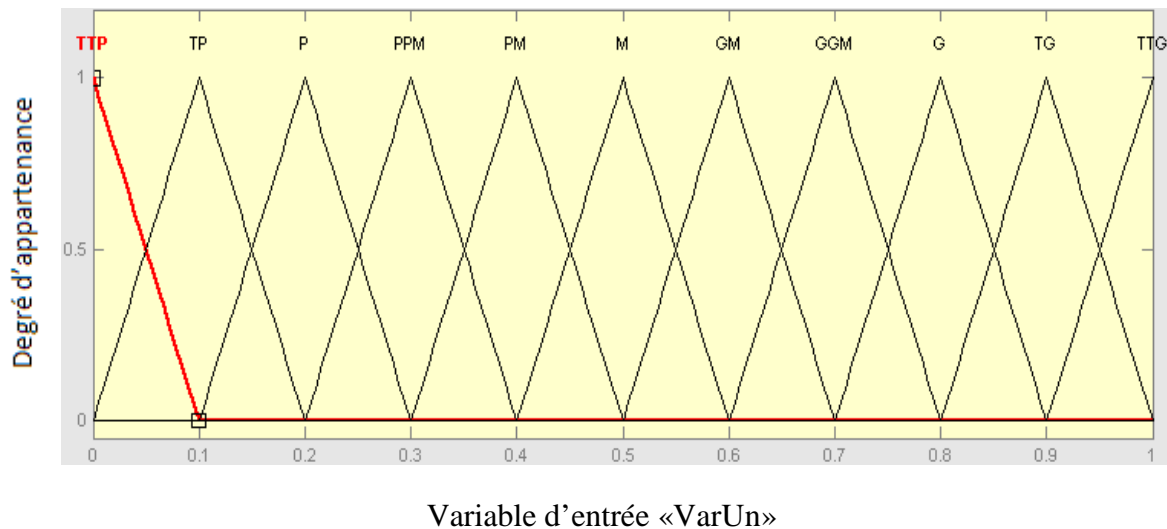
Notre second superviseur « superviseur  $\lambda$  » va nous aider à robustifier notre commande. Il surveille le degré de stabilité du système et règle la valeur de  $\lambda$  en conséquence. En contrepartie, il devra aussi faire en sorte que la variation du signal de commande ne devienne pas trop petite[34].

En effet, comme l'indique le tableau 3.2., quand  $\lambda$  augmente le degré de stabilité du système commandé augmente améliorant ainsi la robustesse de la commande. Par contre, cela va également diminuer la variation du signal de commande.

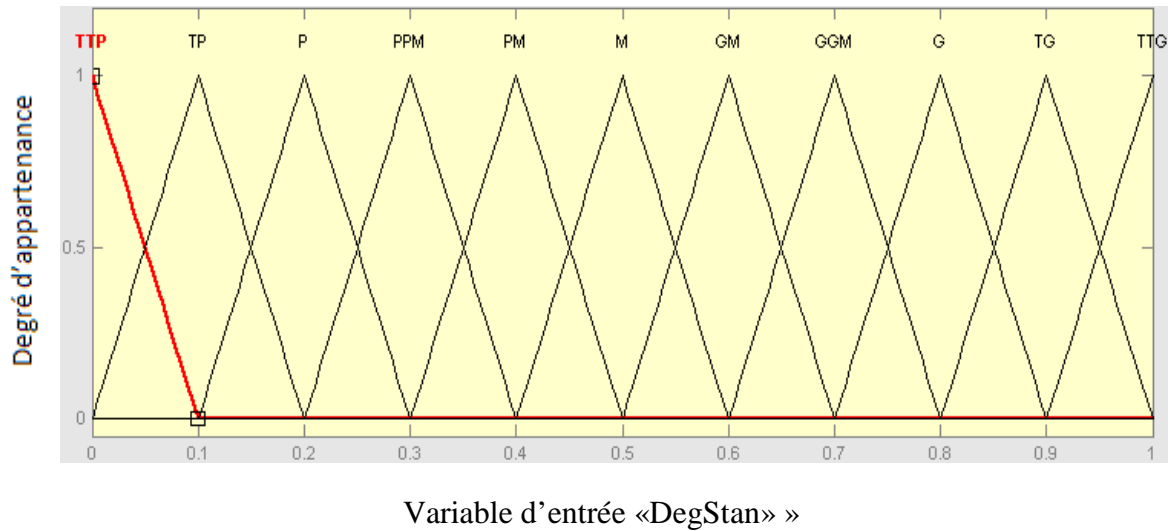
Pour rappel, les variables d'entrée seront normalisées.

Nous avons choisi pour notre « superviseur  $\lambda$  », une valeur de  $N=5$ . Nous associons donc  $2N+1=11$  ensembles flous à chacune des deux entrées de supervision (DegStan et VarUn) et  $4N+1=21$  ensembles flous pour la sortie du superviseur (paramètre de synthèse  $\lambda_n$ ), elle aussi normalisée.

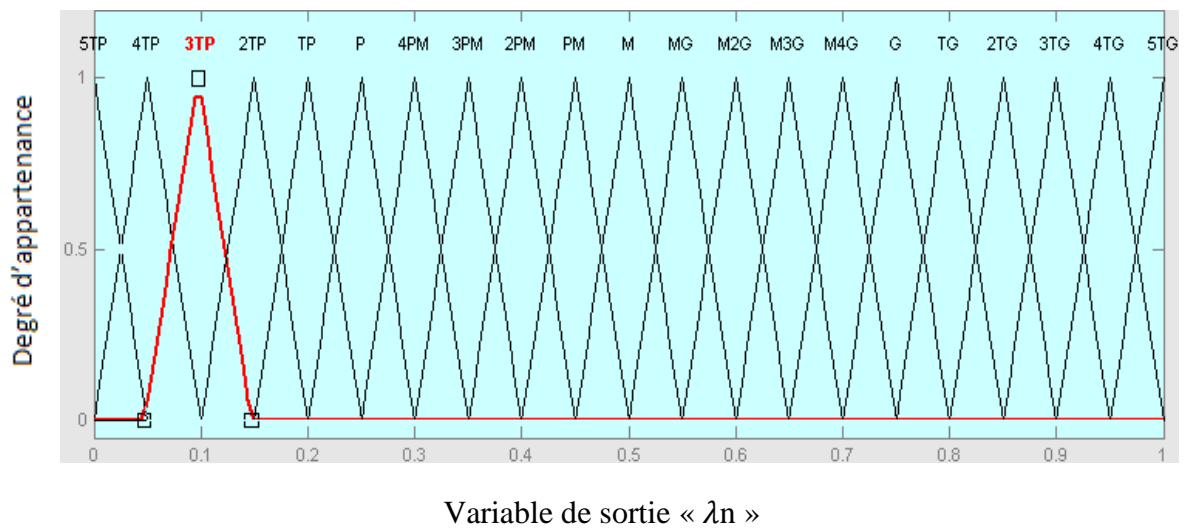
Des fonctions d'appartenance triangulaires définies sur l'intervalle  $[0,1]$  ont été utilisées pour les variables d'entrées et la variable de sortie (voir figure 4.4).



**Figure 4.4.a. Fonctions d'appartenance relatives à la variable d'entrée «VarUn»**



**Figure 4.4.b. Fonctions d'appartenance relatives à la variable d'entrée «DegStan»**



**Figure 4.4.c. Fonctions d'appartenance relatives à la variable de sortie « $\lambda_n$ »**

Pour chaque variable d'entrée, les attributs des ensembles flous vont de très très petit (TTP) à très très grand (TTG). Pour la sortie, ils varient de très très très petit (TTTP) à très très très grand (TTTG).

Les règles floues qui relient le paramètre de synthèse  $\lambda_n$  aux indices de performance DegStan et VarUn sont données par le tableau 4.2.

La méthode d'inférence MAX-MIN a été appliquée afin de calculer les fonctions d'appartenance aux règles floues.

Pour déterminer la sortie normalisée du superviseur  $\lambda_n$ , nous avons utilisé la méthode du barycentre.

$\lambda$  sera égale à :

$$\lambda = [\lambda_n(\lambda_{max} - \lambda_{min}) + \lambda_{min}] \tag{4.9}$$

		DegStan										
		TTP	TP	P	PM	P2M	M	2MG	MG	G	TG	TTG
V a r U n	TTG	5TG	4TG	3TG	2TG	TG	G	3GM	2GM	G2M	G3M	M
	TG	4TG	3TG	2TG	TG	G	3GM	2GM	G2M	G3M	M	3MP
	G	3TG	2TG	TG	G	3GM	2GM	G2M	G3M	M	3MP	2MP
	MG	2TG	TG	G	3GM	2GM	G2M	G3M	M	3MP	2MP	M2P
	2MG	TG	G	3GM	2GM	G2M	G3M	M	3MP	2MP	M2P	M3P
	M	G	3GM	2GM	G2M	G3M	M	3MP	2MP	M2P	M3P	P
	P2M	3GM	2GM	G2M	G3M	M	3MP	2MP	M2P	M3P	P	TP
	PM	2GM	G2M	G3M	M	3MP	2MP	M2P	M3P	P	TP	2TP
	P	G2M	G3M	M	3MP	2MP	M2P	M3P	P	TP	2TP	3TP
	TP	G3M	M	3MP	2MP	M2P	M3P	P	TP	2TP	3TP	4TP
	TTP	M	3MP	2MP	M2P	M3P	P	TP	2TP	3TP	4TP	5TP

Tableau 4.2. Table des règles définissant  $\lambda_n$  en fonction de Degstan et VarUn

#### 4.4. Résultats et simulation

Nous allons à présent, ajouter à la commande GPC présentée au chapitre 3 la couche de supervision floue que nous venons de concevoir.

Nous allons voir qu'à partir d'un choix initial quelconque des paramètres de synthèse de la commande GPC, la couche de supervision floue est capable de proposer de nouveaux paramètres améliorant les performances de la boucle fermée.

Nous avons lancé la simulation avec les jeux de paramètres proposés au chapitre précédent. Le jeu de paramètres  $N_1 = 1, N_2 = 8, N_u = 1, \lambda = \lambda_{opt} = 2.193 \times 10^{-5}$ , pour la boucle GPC1 et le jeu de paramètres suivant pour la boucle GPC2  $N_1 = 1, N_2 = 11, N_u = 1, \lambda = \lambda_{opt} = 4.457 \times 10^{-5}$ .

Pour un temps de simulation de 3 secondes, la durée réelle de la simulation est de 30,66 secondes avec une horloge de 800 Mhz le bloc qui modélise le robot manipulateur consomme 18 secondes, Si nous appliquons la commande à un vrai robot manipulateur, nous économiserons donc ce temps. Pour le compensateur flou, le contrôleur flou ne consomme que très peu de temps de calcul. Ce qui consomme le plus de temps, c'est le calcul de nouveaux paramètres pour la commande GPC, cela prends 9.58s. Le programme permettant le calcul de ces paramètres devra être optimisé.

Nous pouvons voir sur les figures suivantes le suivi de trajectoire pour q1 et q2 :

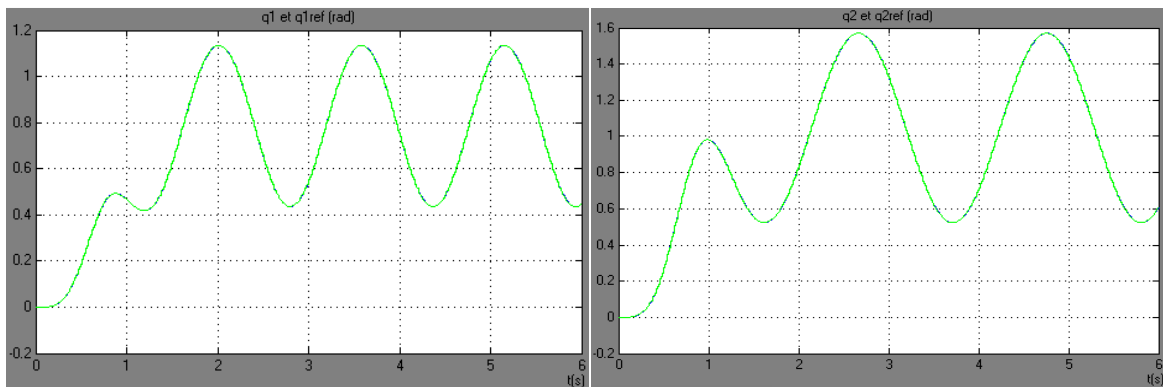


Figure 4.5. Suivi de trajectoire pour q1 et q2

Les figures suivantes montrent l'évolution des performances des boucles de commande GPC1 et GPC2.

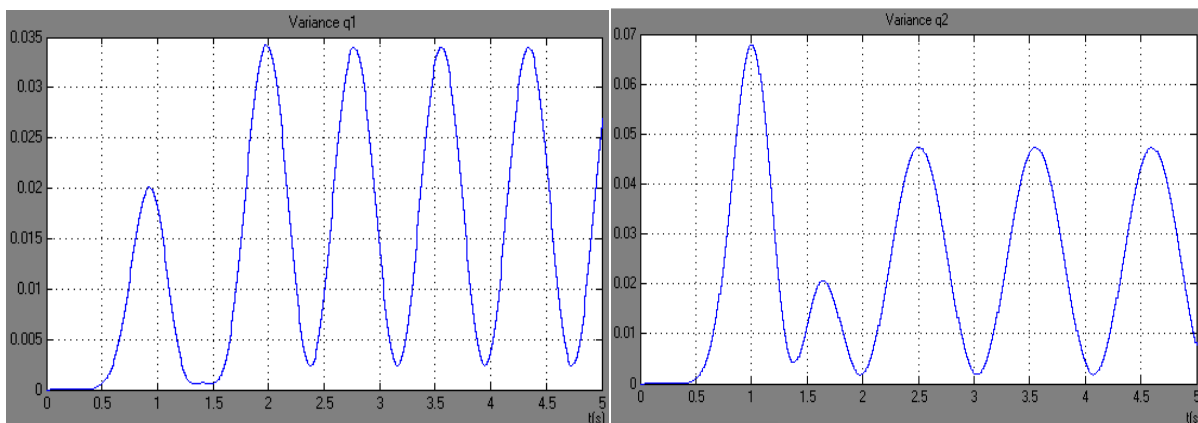
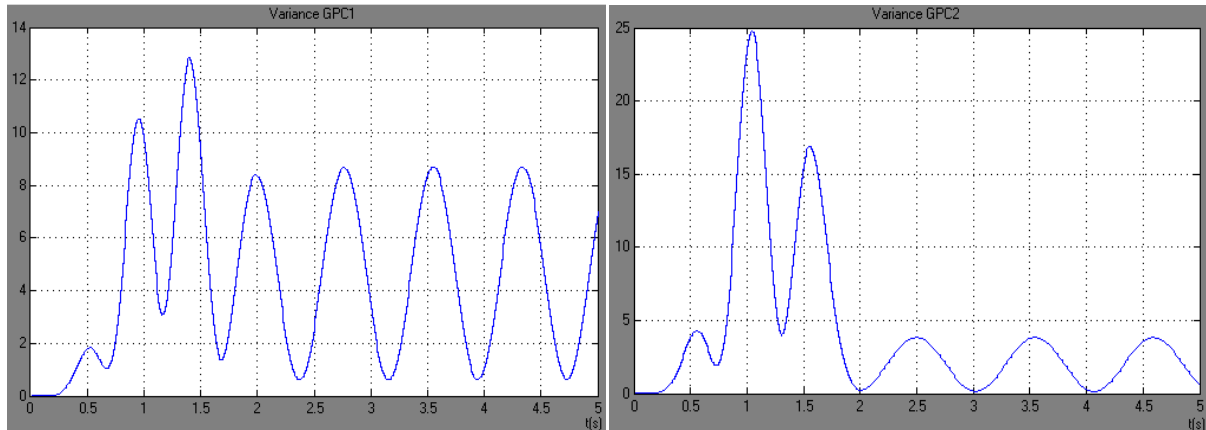
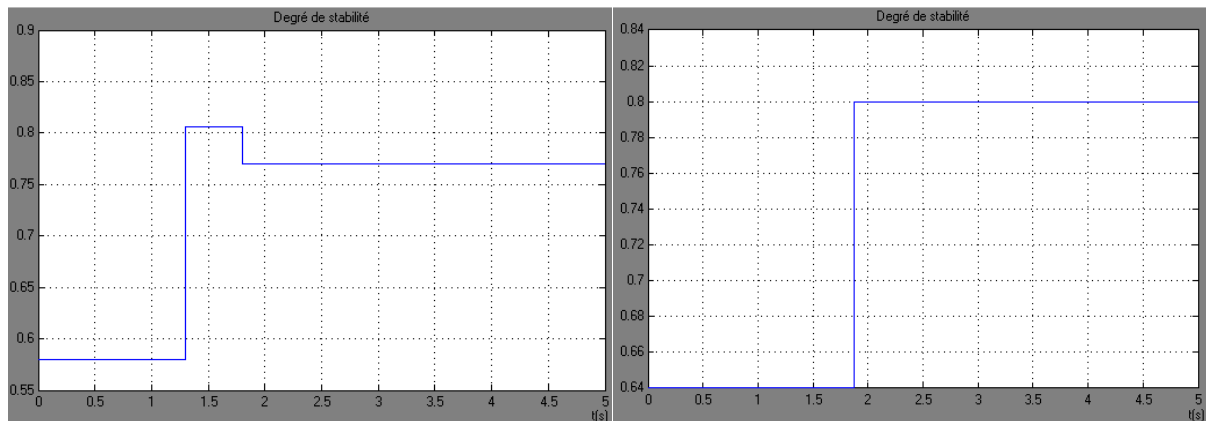


Figure 4.6. Evolution de la variance de signaux q1 et q2 en présence de supervision



**Figure 4.7. Evolution de la variance de signaux GPC1 et GPC2 en présence de supervision**



**Figure 4.8. Evolution du degré de stabilité des boucles de commande GPC1 et GPC2.**

Les valeurs de  $N_2$  et  $\lambda$  proposées par le superviseur flou sont :

Pour GPC1 :  $N_2 = 12$  et  $\lambda = 1.52 \times 10^{-4}$

Pour GPC2 :  $N_2 = 14$  et  $\lambda = 3.19 \times 10^{-4}$

Nous observons sur les figures présentées que la variance du signal  $q_1$  et donc la rapidité augmente avec la supervision floue par contre la variance de  $q_2$  diminue. Les variances des signaux de commande GPC1 et GPC2 diminuent et ont donc été améliorées par la couche de supervision floue.

La couche de supervision floue a permis d'améliorer les degrés de stabilité des deux boucles de commande. Le degré de stabilité de la boucle GPC1 passe de 0.58 à une valeur finale de 0.77. Celui de la boucle GPC2 passe de 0.64 à 0.8.

La couche de supervision floue a permis d'améliorer les performances de la commande en préservant une bonne qualité de poursuite de trajectoire.

#### **4.5. Conclusion**

Dans ce chapitre, nous avons proposé d'ajuster les paramètres de synthèse de la commande GPC à l'aide d'une couche de supervision floue.

Les relations qualitatives existant entre les paramètres de synthèse de la commande GPC et les performances de la commande en boucle fermée nous ont permis de construire un superviseur flou capable d'ajuster les paramètres de la GPC.

La couche de supervision floue proposée est composée de deux contrôleurs flous : un contrôleur chargé d'ajuster  $N_2$  et un deuxième pour ajuster  $\lambda$ .

Les simulations effectuées ont montré des résultats satisfaisants : la couche de supervision floue est capable de proposer des paramètres GPC qui améliorent les performances en boucle fermée.

La première étape de la conception a été le choix des performances à surveiller, et ensuite le choix des paramètres de la commande GPC qui vont nous permettre d'agir sur ces performances.

Ensuite, il a fallu trouver le nombre d'ensembles flous permettant d'obtenir un ajustement précis et une charge calculatoire acceptable.

# Chapitre 5

---

**Linéarisation des  
robots manipulateurs  
avec prise en compte  
des incertitudes :  
approche floue**

## 5.1. Introduction

Dans ce chapitre, Nous reprenons le problème de la linéarisation du modèle du robot manipulateur rigide. La méthode de linéarisation proposée au chapitre 1 requiert une connaissance très précise du modèle dynamique du robot. Ceci n'est jamais le cas dans la pratique à cause de différents facteurs tels que les erreurs de mesure, l'environnement, les variations de la charge ...etc.

Ce chapitre propose une solution combinant la linéarisation à un contrôleur flou chargé de compenser les erreurs dues aux incertitudes structurées et non structurées. La méthode proposée repose sur l'idée que les incertitudes peuvent être estimées par un contrôleur flou. Elle exploite la propriété d'approximateurs universels des systèmes flous.

## 5.2. Formulation du problème

Dans cette section, nous allons appliquer la commande par couple calculé à un robot manipulateur rigide sujet à des incertitudes structurées et non-structurées. Le problème exposé ici a été présenté dans[35].

Le modèle dynamique d'un robot manipulateur rigide à n degrés de liberté sujet à des perturbations externes se présente comme suit :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + f(q, \dot{q}) = \tau \quad (5.1)$$

Où  $\ddot{q}, \dot{q}, q \in R^n$  sont respectivement les vecteurs des accélérations, vitesses et positions articulaires.  $M(q) \in R^{n \times n}$  est la matrice d'inertie,  $C(q, \dot{q})\dot{q} \in R^n$  le vecteur des forces centrifuge et de Coriolis et  $G(q)$  le vecteur des forces de gravité. Le vecteur  $f(q, \dot{q}) \in R^n$  représente les dynamiques non modélisées, les termes dus au frottement, les perturbations externes ...etc.  $\tau \in R^n$  est le vecteur des couples exercés par les actionneurs au niveau des différentes articulations.

Les matrices  $M(q), C(q, \dot{q})$  et  $G(q)$  du modèle dynamique sont fonction des paramètres physiques du manipulateur, tels que les masses, les dimensions et les moments d'inertie des différents éléments composant la structure du robot.

Les valeurs précises de ces paramètres ne sont pas connues. Ceci est dû à différents facteurs tels que les erreurs de mesures, l'environnement, les variations de la charge ...etc. Nous pouvons ainsi supposer que les valeurs réelles des matrices  $M(q), C(q, \dot{q})$  et  $G(q)$  peuvent être séparées en deux parties : une partie nominale qui sera notée  $M_0(q), C_0(q, \dot{q})$  et  $G_0(q)$  et une partie incertaine notée  $\Delta M(q), \Delta C(q, \dot{q})$  et  $\Delta G(q)$  respectivement [35] :

$$M(q) = M_0(q) - \Delta M(q), \quad (5.2)$$

$$C(q, \dot{q}) = C_0(q, \dot{q}) - \Delta C(q, \dot{q}), \quad (5.3)$$

$$G(q) = G_0(q) - \Delta G(q) \quad (5.4)$$

Supposons, en premier lieu, que le modèle dynamique du robot est parfaitement connu. Ceci implique que  $\Delta M(q), \Delta C(q, \dot{q})$  et  $\Delta G(q)$  dans (5.2)-(5.4) et  $f(q, \dot{q})$  dans (5.1) sont tous égaux à zéro.

Le modèle dynamique s'écrit alors comme le modèle nominal :

$$M_0(q)\ddot{q} + C_0(q, \dot{q})\dot{q} + G_0(q) = \tau \quad (5.5)$$

Considérons à présent la loi de commande :

$$\tau = M_0(q)(\ddot{q}_d - K_v\dot{e} - K_p e) + C_0(q, \dot{q})\dot{q} + G_0(q) \quad (5.6)$$

Les vecteurs des accélérations, vitesses et positions articulaires relatifs à la trajectoire désirée sont respectivement notés :  $\ddot{q}_d, \dot{q}_d, q_d \in R^n$ .  $e = q - q_d$  est le vecteur des erreurs de poursuite.  $K_p$  et  $K_v$  sont les matrices de gains proportionnels et dérivés. Ces matrices sont constantes.

En remplaçant la commande (5) dans l'équation du modèle dynamique (5.5), nous obtenons :

$$\ddot{e} + K_v \dot{e} + K_p e = 0 \quad (5.7)$$

$K_p$  et  $K_v$  sont choisis de façon à ce que l'erreur de poursuite tende asymptotiquement vers 0. La loi de commande (5.6) est appelée commande par couple calculé.

La commande par couple calculé est basée sur la linéarisation du modèle dynamique du manipulateur. Ce qui permet d'aboutir à un système linéaire invariant (5.7). La stabilité de ce système est garantie pourvu que les matrices  $K_p$  et  $K_v$  soient symétriques et définies positives.

A noter que cette procédure repose sur l'hypothèse que notre connaissance du modèle du robot est parfaite. Ce qui dans la pratique est impossible.

Si Nous appliquons, à présent, la commande (5.6) au modèle dynamique incluant les incertitudes, autrement dit au modèle représenté par l'équation (5.1) avec  $M(q), C(q, \dot{q})$  et  $G(q)$  vérifiant les équations (5.2)-(5.4), Nous obtenons l'équation suivante [35]:

$$\ddot{e} + K_v \dot{e} + K_p e = \rho \quad (5.8)$$

Avec :  $\rho = M_0^{-1} [ \Delta M(q) \ddot{q} + \Delta C(q, \dot{q}) \dot{q} + \Delta G(q) - f(q, \dot{q}) ]$ ,

$\rho$  est une fonction des variables articulaires, des paramètres physiques, de la variation de ces paramètres, des dynamiques non modélisées telles que les perturbations externes...etc.  $\rho$  regroupe les incertitudes structurées et les incertitudes non structurées. L'existence de ce terme rend invalide la commande par couple calculé.

Afin de remédier à ce problème, il faut compenser le terme  $\rho$ . Pour ce faire, nous allons rajouter au couple  $\tau$  représenté par l'équation (5.6), un couple de compensation  $\tau_c$ .

$$\tau = M_0(q) (\ddot{q}_d - K_v \dot{e} - K_p e) + C_0(q, \dot{q}) \dot{q} + G_0(q) + \tau_c \quad (5.9)$$

En remplaçant cette nouvelle équation (5.9) dans l'équation représentant le modèle dynamique réel du système, l'équation suivante est obtenue :

$$\ddot{e} + K_v \dot{e} + K_p e = \rho + M_0^{-1}(q)\tau_c \quad (5.10)$$

Le but est d'annuler cette dernière équation :

$$\ddot{e} + K_v \dot{e} + K_p e = \rho + M_0^{-1}(q)\tau_c = 0 \quad (5.11)$$

La loi de commande supplémentaire  $\tau_c$  sera calculée en estimant le terme  $\rho$  au moyen d'un compensateur flou.

### 5.3. Compensateur flou

Les systèmes flous sont des approximateurs universels. Les systèmes flous permettent d'obtenir une approximation de n'importe quelle fonction et ce en utilisant des règles d'inférence adaptatives ayant des paramètres ajustables. Cette propriété des systèmes flous a fait l'objet de plusieurs ouvrages et de plusieurs travaux de recherche.

Nous allons exploiter cette caractéristique afin d'obtenir une approximation de la fonction  $\rho$  regroupant les incertitudes structurées et les incertitudes non structurées auxquelles est sujet le manipulateur rigide et pouvoir ainsi les compenser.

#### 5.3.1. Conception du compensateur flou

Dans cette section, nous nous intéressons à l'élaboration d'un contrôleur flou qui servira de compensateur pour la commande par couple calculé du système avec incertitudes structurées et non-structurées, tel qu'il a été expliqué précédemment.

##### 5.3.1.1. Description du système d'inférence floue utilisé

La base de règles floues qui va être employée consiste en une collection de règles de la forme :

$R^j$  : si  $x_1$  est  $A_1^j$  et  $x_2$  est  $A_2^j$  et .... et  $x_n$  est  $A_n^j$ ,

Alors  $y_1$  est  $B_1^j$  et  $y_2$  est  $B_2^j$  et .... et  $y_m$  est  $B_m^j$ .

Où  $x = [x_1, x_2, \dots, x_n]^T$  et  $y = [y_1, y_2, \dots, y_m]^T$  sont respectivement les vecteurs d'entrée et de sortie du système flou.

$A_i^j$  ( $i = 1, 2, \dots, n$ ) et  $B_k^j$  ( $k = 1, 2, \dots, m$ ) sont les variables linguistiques relatives aux ensembles flous dans les sous-espaces  $U_i$  et  $V_k$ , décrits par leurs fonctions d'appartenance respectives  $\mu_{A_i^j}(x_i)$  et  $\mu_{B_k^j}(y_k)$ ,  $j=1, 2, \dots, M$ .  $M$  est le nombre total de règles floues.

En utilisant un fuzzificateur de type singleton, le produit d'inférence et un défuzzificateur de type centre de gravité, la sortie du système flou peut être exprimée comme suit :

$$y(x) = \hat{\Omega}(x|\hat{\Theta}) = \frac{\sum_{j=1}^M \bar{y}^j (\prod_{i=1}^n \mu_{A_i^j}(x_i))}{\sum_{j=1}^M (\prod_{i=1}^n \mu_{A_i^j}(x_i))} = \hat{\Theta} \xi(x) \quad (5.12)$$

Avec  $\mu_{A_i^j}(x_i)$  la fonction d'appartenance de la variable floue  $x_i$ ,  $\bar{y}^j$  est le point où  $\mu_{B_k^j}(y_k)$  atteint sa valeur maximale,  $\mu_{B_k^j}(\bar{y}_j) = 1$ .

$$\hat{\Theta} = \begin{pmatrix} \bar{y}_1^1 & \dots & \bar{y}_1^M \\ \vdots & \ddots & \vdots \\ \bar{y}_m^1 & \dots & \bar{y}_m^M \end{pmatrix} \quad (5.13)$$

$\hat{\Theta}$  est une matrice ajustable et  $\xi(x) = [\xi^1(x), \xi^2(x), \dots, \xi^M(x)]^T$  est la fonction vectorielle de base floue dans laquelle  $\xi^j(x)$  est défini par :

$$\xi^j(x) = \frac{\prod_{i=1}^n \mu_{A_i^j}(x_i)}{\sum_{j=1}^M (\prod_{i=1}^n \mu_{A_i^j}(x_i))}, \quad (j=1, 2, \dots, M) \quad (5.14)$$

Pour plus de détails voir annexe I.

### 5.3.1.2. Description du compensateur flou

Pour commencer, nous supposons que le terme  $\rho$  dans l'équation (5.10) peut être représenté par un système flou idéal [35]:

$$\rho(z) = W^* \xi(z) + \varepsilon(z) \quad (5.15)$$

Où  $\varepsilon(z)$  est l'erreur de reconstruction du système flou,  $\xi(z) = [\xi^1(z), \xi^2(z), \dots, \xi^N(z)]^T$  est une fonction vectorielle floue fixée par le concepteur.  $\xi^i(z), i = 1, 2, \dots, N$  correspond à l'équation (5.14) et  $z(t)$  est donné par  $z(t) = [\dot{q}^T, \dot{q}^T, q^T]^T$ .

$$W^* = [w_{ij}] \in \mathbb{R}^{n \times n}$$

$$W^* = \arg \min_w \left\{ \sup_{x \in Dz} |\hat{\rho}(z|\hat{W}) - \rho(z)| \right\} \quad (5.16)$$

$Dz$  indique les ensembles des limites adéquates de  $z$ . Nous supposons que  $z$  n'atteint jamais les bornes de  $Dz$ .

Dans l'équation (5.16),  $\hat{\rho}(z|\hat{W})$  est une estimation de  $\rho(z)$  qui peut être approchée par un système flou de la forme :

$$\hat{\rho}(z|\hat{W}) = \hat{W}\xi(z) \quad (5.17)$$

$\hat{W}$  est une matrice de poids ajustable correspondant à l'équation (5.13). La loi de contrôle compensatrice  $\tau_c$  s'exprime :

$$\tau_c = -M_0(q)\hat{\rho}(z|\hat{W}) \quad (5.18)$$

En remplaçant (5.18) dans (5.10), le système en boucle fermée devient

$$\ddot{e} + K_v \dot{e} + K_p e = \tilde{\rho}(z) \quad (5.19)$$

Où :

$$\tilde{\rho}(z) = \rho(z) - \hat{\rho}(z|\hat{W}) = \varepsilon(z) + \tilde{W}\zeta(z) \quad (5.20)$$

Avec :

$\tilde{W} = W^* - \hat{W}$  exprimant l'erreur sur la matrice de poids.

Si nous supposons que le vecteur d'état  $x = [e^T, \dot{e}^T]^T$ , l'équation d'état aura la forme [35]:

$$\dot{x} = Ax + B\tilde{\rho}(z) \quad (5.21)$$

$$\text{Où } A = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}, B = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

Nous allons nous intéresser à un problème de suivi de trajectoire. Pour développer le reste de la méthode, le postulat suivant doit être vérifié :

**Postulat n°1 :** La trajectoire désirée  $q_d$  est une fonction du temps connue, continue et bornée ayant des dérivées première et seconde connues et bornées.

Afin de poursuivre le développement et l'analyse de la stabilité du système en boucle fermée, les hypothèses suivantes sont posées :

**Postulat n° 2 :** l'erreur de reconstruction  $\varepsilon(z)$  est bornée, i.e.  $\|\varepsilon\| \leq c_\varepsilon \forall z \in Dz$  avec  $c_\varepsilon$  connu.

**Postulat n° 3 :** la norme de la matrice de poids optimale est bornée tel que  $\|W^*\| \leq c_w$ .

Théorème [35]:

Considérons un robot manipulateur comportant des incertitudes structurées et non structurées. Supposons que les postulats 1-3 sont vérifiés. La loi de commande est donnée par l'équation (5.9) comportant une commande par couple calculé (5.6), et la commande floue (5.18). Les paramètres de la commande floue sont calculés comme suit :

$$\hat{W} = \Gamma^{-1} B^T P x \xi^T \quad (5.22)$$

$\Gamma = \text{diag}(\Gamma_1, \Gamma_2, \dots, \Gamma_n)$ ,  $\Gamma_i > 0$  ( $i = 1, 2, \dots, n$ ) est la matrice de gain. P est une matrice définie positive solution de l'équation de Riccati :

$$A^T P + P A + P^T B B^T P + Q = 0 \quad (5.23)$$

Où :

$Q$  est une matrice constante donnée à l'avance.

Alors :

1)  $x(t)$  est uniformément borné.

2) Si l'erreur de reconstruction  $\varepsilon \in L_2$ , i.e.  $\int_0^\infty \varepsilon^2(t)dt < \infty$ , l'erreur de suivi du robot manipulateur tend vers 0 quand  $t$  tend vers l'infini.

La méthode repose sur l'idée que les incertitudes peuvent être estimées par un système flou. L'équation (5.22) permettant l'adaptation des poids est obtenue en utilisant le théorème de stabilité de Lyapunov.

Preuve [35]:

1) Considérons le candidat de Lyapunov :

$$V = x^T P x + Tr(\tilde{W}^T \Gamma \tilde{W}), \quad (5.24)$$

$$\text{Avec } \tilde{W} = W^* - \hat{W}$$

$\dot{V}$  est la dérivée de  $V$  par rapport au temps, le long de (20)

$$\dot{V} = x^T (A^T P + P A) x + \tilde{\rho}^T B^T P x + x^T P B \tilde{\rho} + 2Tr(\tilde{W}^T \Gamma \dot{\tilde{W}})$$

$$\dot{V} = x^T (A^T P + P A) x + 2x^T P B \tilde{\rho} - 2Tr(\hat{W}^T \Gamma \dot{\tilde{W}})$$

$$\dot{V} = -x^T (P^T B B^T P + Q) x + 2x^T P^T B (\varepsilon + \tilde{W} \xi) - 2Tr(\tilde{W}^T \Gamma \dot{\tilde{W}})$$

$$\dot{V} = -x^T (P^T B B^T P + Q) x + 2x^T P^T B \varepsilon + 2x^T P^T B \tilde{W} \xi - 2Tr(\tilde{W}^T \Gamma \dot{\tilde{W}})$$

$$\dot{V} = -x^T (P^T B B^T P + Q) x + 2x^T P^T B \varepsilon + 2Tr(\xi^T \tilde{W}^T B^T P x) - 2Tr(\tilde{W}^T \Gamma \dot{\tilde{W}})$$

$$\dot{V} = -x^T (P^T B B^T P + Q) x + 2x^T P^T B \varepsilon + 2Tr(\tilde{W}^T B^T P x \xi^T) - 2Tr(\tilde{W}^T \Gamma \dot{\tilde{W}})$$

$$\dot{V} = -x^T(P^T B B^T P + Q)x + 2x^T P^T B \varepsilon + 2Tr[\tilde{W}^T (B^T P x \xi^T - \Gamma \dot{\tilde{W}})]$$

$$\dot{V} = -x^T(P^T B B^T P + Q)x + 2x^T P^T B \varepsilon + 2Tr[\tilde{W}^T (B^T P x \xi^T - \Gamma(\Gamma^{-1} B^T P x \xi^T))]$$

$$\dot{V} = -x^T(P^T B B^T P + Q)x + 2x^T P^T B \varepsilon$$

$$\dot{V} = -x^T Q x - (B^T P x - \varepsilon)^T (B^T P x - \varepsilon) + \varepsilon^T \varepsilon$$

$$\dot{V} \leq -x^T Q x + \varepsilon^T \varepsilon$$

$$\dot{V} \leq -\lambda_{\min}(Q) \|x\|^2 + \|\varepsilon\|^2 \quad (5.25)$$

$V(x, \tilde{W})$  est négatif à l'extérieur du compact  $\Sigma_x$  :

$$\Sigma_x = \left\{ x(t) \left| 0 \leq \|x(t)\| \leq \sqrt{\frac{1}{\lambda_{\min}(Q)}} \|\varepsilon\| \right. \right\}$$

D'après le postulat 3,  $W$  est bornée ( $\tilde{W}$  est aussi bornée), ce qui implique que  $x(t)$  est uniformément bornée.

2) En intégrant l'équation (5.25) de 0 à  $+\infty$ , nous obtenons :

$$\int_0^{+\infty} x^T Q x dt \leq \int_0^{+\infty} \varepsilon^T \varepsilon dt + V(0) - V(+\infty)$$

Ce qui implique :

$$\int_0^{+\infty} \|x\|^2 dt \leq \frac{k}{\lambda_{\min}(Q)} \quad (5.26)$$

$$\text{Où } k = \int_0^{+\infty} \varepsilon^T \varepsilon dt + V(0) - V(+\infty)$$

Puisque  $V(t)$  est une fonction non croissante et bornée par le bas, cela implique que  $V(0) - V(+\infty) < \infty$  et  $\int_0^{+\infty} \varepsilon^2(t) dt < \infty$ . Il en résulte  $k < \infty$ .

Il est facile de conclure à partir de l'équation (5.26) que  $x \in L_2$ .

$x(t)$  est borné par le haut, ce qui implique que  $x \in L_\infty$ .

A partir de l'équation du système en boucle fermée (5.21) et puisque  $x(t)$ ,  $\tilde{W}(t)$ , et  $\varepsilon(t)$  sont bornées, nous obtenons  $\dot{x} \in L_\infty$ .

Nous avons donc  $x \in L_\infty \cap L_2$  et  $\dot{x} \in L_\infty$ . Nous pouvons en conclure en utilisant le postulat 1 que  $\lim_{t \rightarrow \infty} x(t) = 0$ .

Nous venons de démontrer que le système en boucle fermée est stable, c.à.d. que l'erreur de suivi tend vers 0 quand  $t$  tend vers l'infini.

Les fonctions de base sont choisies à travers des tests de simulation. La fonction de base  $\xi$  est calculée en utilisant la relation (5.14).

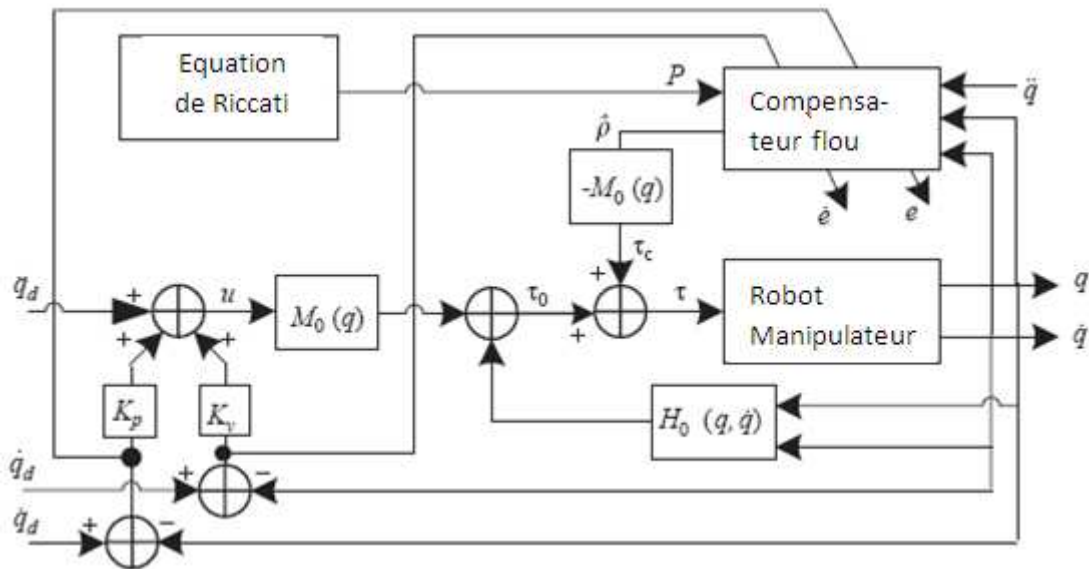


Figure 5.1. Diagramme du système linéarisé [35]

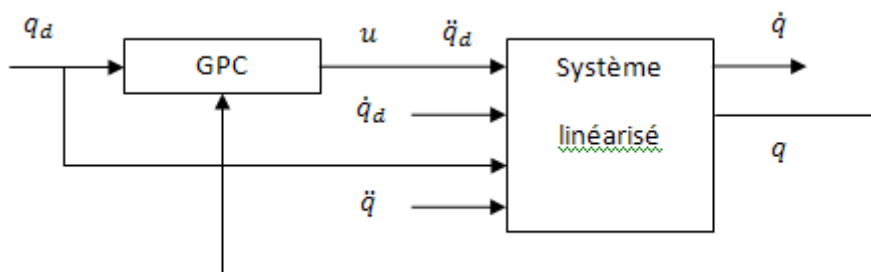


Figure 5.2. Commande GPC du système linéarisé.

## 5.4. Résultats et simulation

Afin de pouvoir juger l'efficacité de la méthode de compensation exposée dans la partie théorique de ce chapitre, nous allons effectuer des tests de simulation sur le robot manipulateur rigide présenté aux chapitres 1 et 3. La méthode présentée sera utilisée pour linéariser le modèle du manipulateur. Les paramètres de la commande par couple calculé sont  $K_p = \text{diag}(5,5)$  et  $K_d = \text{diag}(7.5,7.5)$ . Nous appliquons au modèle linéarisé la commande GPC proposée au chapitre 3.

Le compensateur flou a pour entrées les vecteurs  $q$ ,  $\dot{q}$  et  $\ddot{q}$ .

Le vecteur  $z$  dans l'équation (5.15) s'écrit :  $z = [z_1^T, z_2^T, z_3^T]^T$  où  $z_i^T = [z_{i1}, z_{i2}]^T$ .

$z_{1j} = \ddot{q}_j$ ,  $z_{2j} = \dot{q}_j$ ,  $z_{3j} = q_j$ . pour  $j = 1, 2$ .

Le vecteur  $\rho$ , sortie de notre compensateur flou s'écrit :  $\rho = [\rho_1, \rho_2]$ .

La valeur initiale des paramètres  $\widehat{W}(0) = 0$ .

Nous allons introduire des incertitudes lors de la simulation de la boucle de commande. Nous commençons par introduire des incertitudes structurées (variation des paramètres du robot). Nous allons varier les valeurs des masses  $m_1$  et  $m_2$  relatives aux axes 1 et 2 du manipulateur. Le but étant d'observer la réponse de la commande au fur et à mesure que l'écart entre les valeurs réelles et les valeurs connues (valeurs nominales utilisée pour concevoir la commande) augmente.

Nous nous intéresserons ensuite aux incertitudes non structurées (perturbations externes) notées  $f(q, \dot{q})$  dans l'équation du modèle dynamique  $M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + f(q, \dot{q}) = \tau$ .

Nous prendrons  $f(q, \dot{q}) = [\sin(t) \cos(t)]^T$  et augmenterons ensuite la valeur des perturbations pour observer la réponse de la boucle de commande.

Commençons par présenter les résultats obtenus en présence de perturbations sans compensation floue pour le robot rigide. Les figures suivantes illustrent le cas  $m_1=11$  kg,  $m_2=6.5$  kg et  $f(q, \dot{q}) = 2[\sin(t) \cos(t)]^T$ .

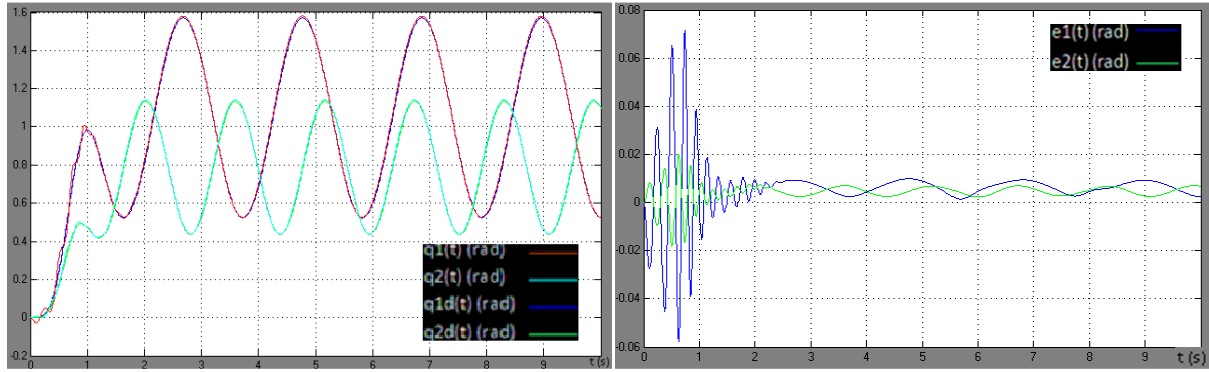


Figure 5.3. Suivi et erreurs de suivi sans compensation floue pour  $m_1=11$  kg,  $m_2= 6.5$  kg et  $f(q, \dot{q}) = 2[\sin(t) \cos(t)]^T$

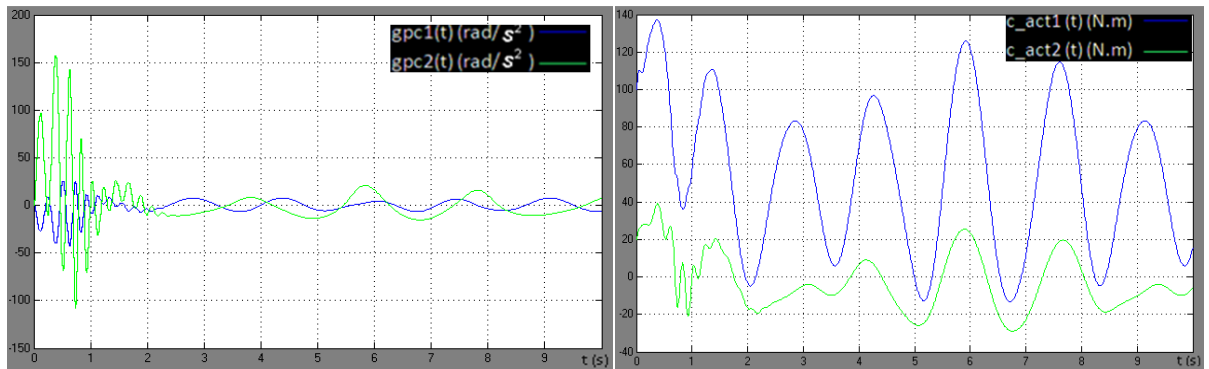


Figure 5.4. Commandes GPC1 et GPC2 et commandes actionneurs sans compensation floue pour  $m_1=11$  kg,  $m_2= 6.5$  kg et  $f(q, \dot{q}) = 2[\sin(t) \cos(t)]^T$

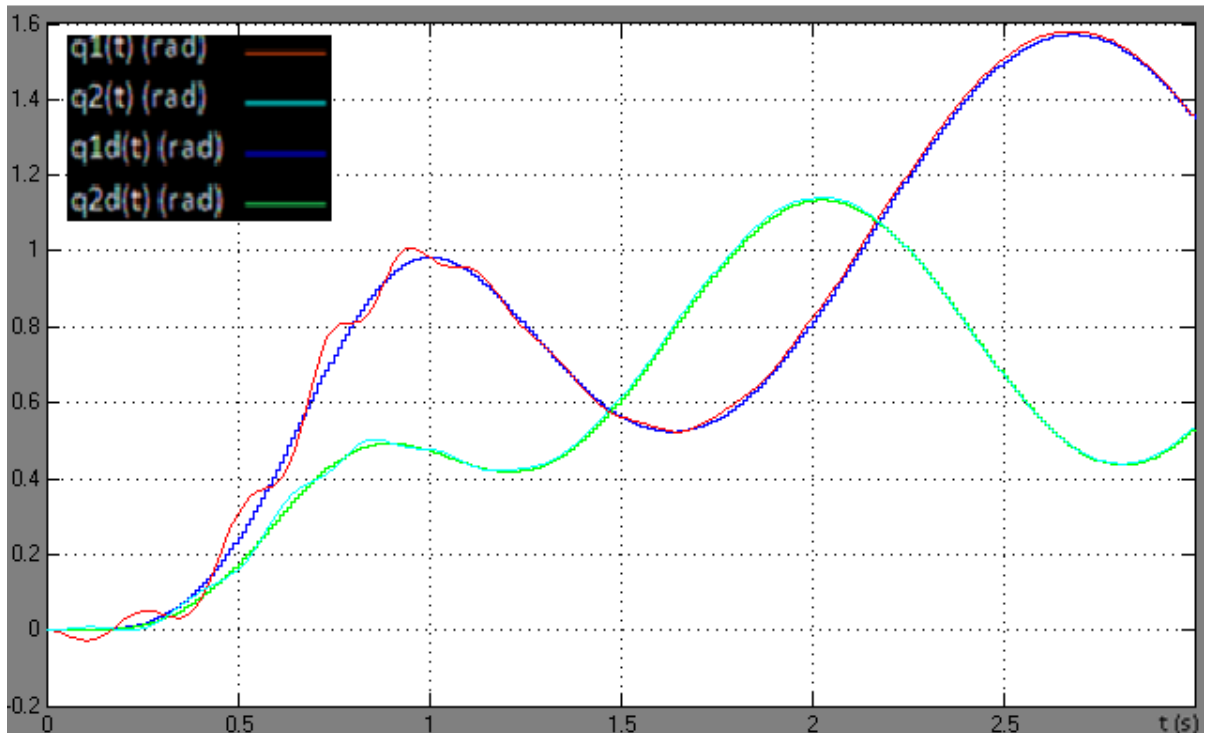


Figure 5.5. Suivi sans compensation floue pour  $m_1=11$  kg,  $m_2= 6.5$  kg et  $f(q, \dot{q}) = 2[\sin(t) \cos(t)]^T$

Nous pouvons aisément voir que la qualité de la commande se dégrade. Des oscillations importantes apparaissent au lancement de la commande. Au bout de deux secondes la commande devient plus lisse mais l'erreur de suivi ainsi que l'amplitude du signal de commande restent importantes.

Le tableau qui suit donne les erreurs de suivi maximales enregistrées pour différentes valeurs de  $m_1$  et  $m_2$ . Plus nous nous éloignons des valeurs nominales plus l'erreur de poursuite est importante :

$m_1$ (kg)	8	11	14	15	15.91	24
$m_2$ (kg)	6	6.5	9.5	10.5	11.36	18
$e_{1\_max}$ (rad)	0.007	0.0035	0.0006	0.00023	0.00036	0.006
$e_{2\_max}$ (rad)	0.015	0.0076	0.007	0.0069	0.0065	0.007

Tableau 5.1.  $e_{1\_max}$  et  $e_{2\_max}$  en fonction des masses  $m_1$  et  $m_2$

Nous pouvons conclure que les résultats relatifs au cas idéal ne sont jamais égalés mais le compensateur flou nous permet d'obtenir des résultats convenables en présence de perturbations.

Nous pouvons voir sur les figures suivantes la variation des coordonnées articulaires ainsi que l'erreur de poursuite pour  $m_1=8$  kg,  $m_2=6$  kg et pour  $m_1=24$ kg et  $m_2=18$ kg.

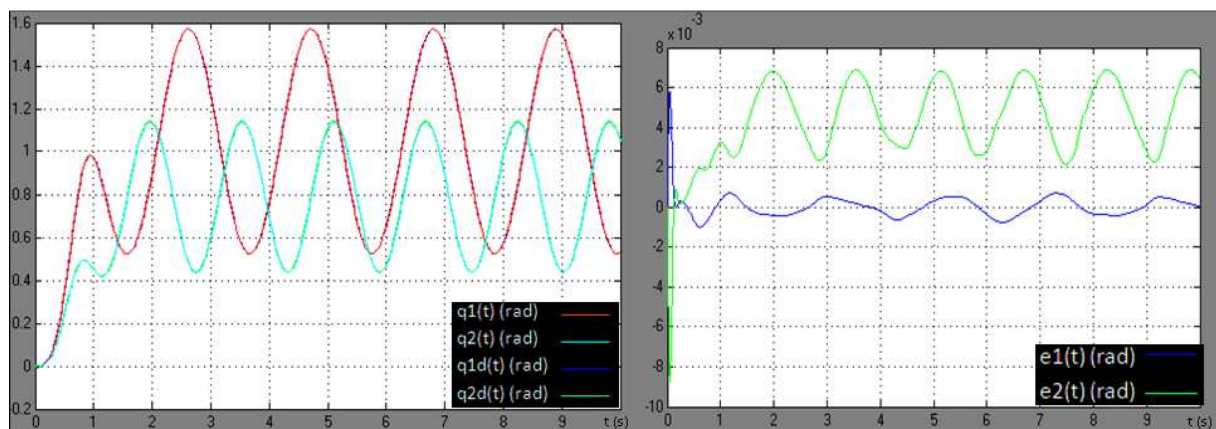


Figure 5.6. Suivi et erreurs de suivi pour  $m_1 = 8$  kg;  $m_2 = 6$  kg

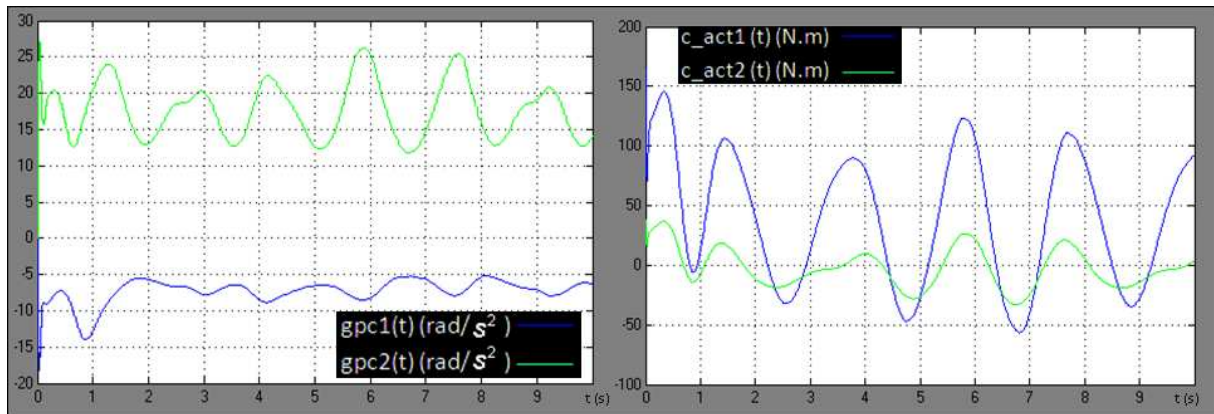


Figure 5.7. Commandes GPC1 et GPC2 et commandes actionneurs  $m1 = 8 \text{ kg}$ ;  $m2 = 6 \text{ kg}$

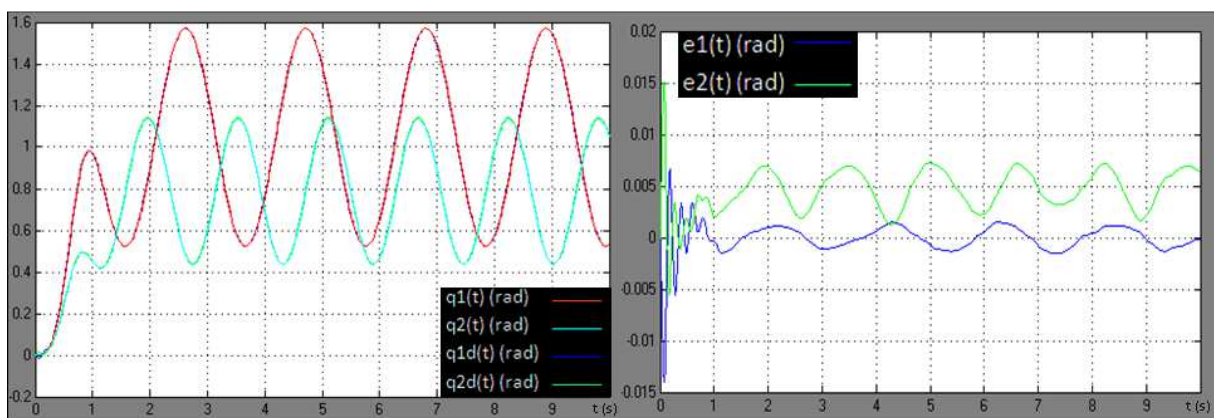


Figure 5.8. Suivi et erreurs de suivi pour  $m1 = 24 \text{ kg}$ ;  $m2 = 18 \text{ kg}$

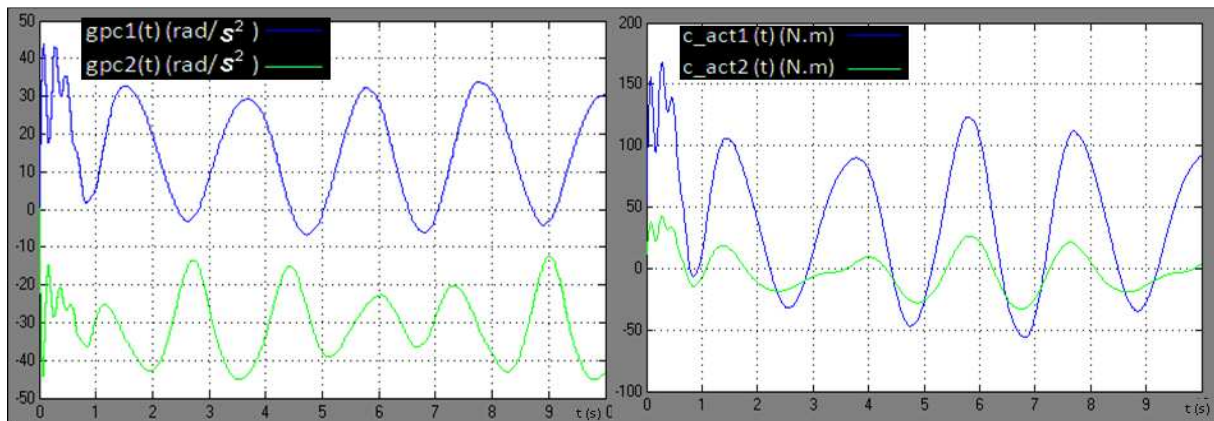


Figure 5.9. Commandes GPC1 et GPC2 et commandes actionneurs  $m1=24 \text{ kg}$ ;  $m2=18 \text{ kg}$

L'amplitude des signaux de commande augmente avec l'augmentation de la charge.

Pour un temps de simulation de 3 secondes, la durée réelle de la simulation est de 25,58 secondes avec une horloge de 800 Mhz. Cela peut paraître énorme, mais si nous regardons plus en détail, nous constatons que le bloc qui modélise le robot manipulateur

consomme 23,68 secondes, soit 92,6% du temps de simulation. Si nous appliquons la commande à un vrai robot manipulateur, nous économiserons donc ce temps.

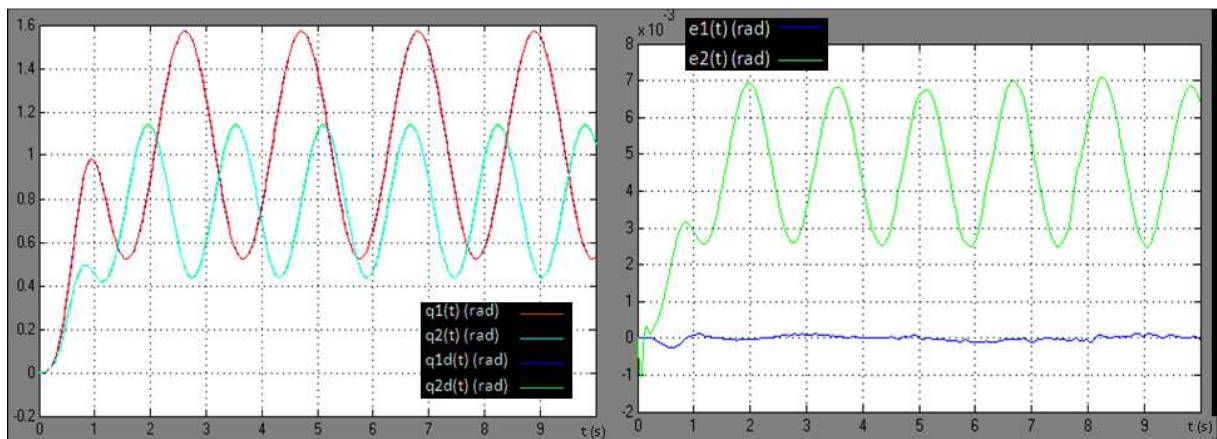
Intéressons nous à présent aux incertitudes non structurées (perturbations externes) notées  $f(q, \dot{q})$  dans l'équation du modèle dynamique  $M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + f(q, \dot{q}) = \tau$ .

Nous prendrons  $f(q, \dot{q}) = [\sin(t) \cos(t)]^T$  et augmenterons ensuite la valeur des perturbations pour observer la réponse de la boucle de commande.

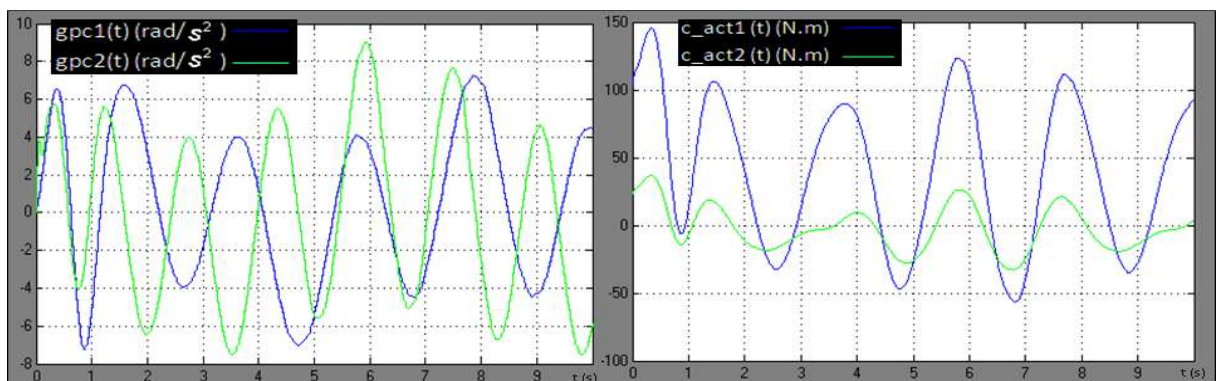
Le tableau qui suit donne les erreurs de suivi maximales enregistrées pour des valeurs différentes de  $f(q, \dot{q})$ . L'erreur de suivi augmente avec l'amplitude de  $f(q, \dot{q})$ .

$f(q, \dot{q})$	$[\sin(t) \cos(t)]^T$	$2[\sin(t) \cos(t)]^T$	$10[\sin(t) \cos(t)]^T$
$e_{1\_max}(rad)$	0.0001	0.00015	0.0003
$e_{2\_max}(rad)$	0.0065	0.007	0.0075

**Tableau 5.2. Erreurs de suivi maximales pour différentes valeurs de  $f(q, \dot{q})$**



**Figure 5.10. Suivi et erreurs de suivi pour  $f(q, \dot{q}) = 2[\sin(t) \cos(t)]^T$**



**Figure 5.11. Commandes GPC1 et GPC2 et commandes actionneurs  $f(q, \dot{q}) = 2[\sin(t) \cos(t)]^T$**

Les différents tests effectués en simulation sur le manipulateur rigide montrent un bon fonctionnement de la stratégie de commande face aux incertitudes.

Nous allons maintenant appliquer la méthode de commande précédente à un bras manipulateur plan à deux liaisons rotoïdes et deux axes flexibles.

Le tableau suivant résume les paramètres relatifs au manipulateur flexible utilisé en simulation :

$\rho_1 = \rho_2 = \rho$	Masse linéique	0.2	Kg/m
$l_1 = l_2$	Longueur axe i	0.5	m
$m_1 = m_2$	Masse de l'axe i	0.1	Kg
$m_p$	Masse de la charge	0.1	Kg
$J_{O1} = J_{O2}$	Inertie de l'axe i	0.0083	Kg.m <sup>2</sup>
$J_{h1} = J_{h2}$	Inertie de l'actionneur i	0.1	Kg.m <sup>2</sup>
$J_p$	Inertie de la charge	0.0005	KG ;M <sup>2</sup>
$E_I$	Rigidité en flexion	1	N.m <sup>2</sup>

**Tableau 5.3. Paramètre du manipulateur flexible**

Le but est de tester le comportement de la commande lors de l'augmentation de la flexibilité d'un bras manipulateur à axes flexibles.

Afin d'augmenter la flexibilité du robot, deux stratégies sont employées :

D'abord, l'augmentation de la charge ;

Ensuite, l'augmentation de la vitesse de travail du robot.

Nous observerons à chaque fois, la qualité du suivi, l'erreur de suivi, le signal de commande GPC ainsi que les couples délivrés par les actionneurs.

#### 5.4.1. Augmentation de la charge

Le tableau suivant montre les différentes valeurs de la charge  $m_p$ , ainsi que les fréquences de vibration correspondantes :

$m_p$	$M_{L1}/m_1$	$M_{L2}/m_2$	$f_{11}$	$f_{12}$	$f_{21}$	$f_{22}$
0	11	0	0.74	21.98	4.98	31.31
0.1	12	1	0.72	21.98	2.19	23.12
0.2	13	2	0.70	21.98	1.65	22.55
0.4	15	5	0.63	21.98	1.08	22.10
0.9	20	9	0.55	21.98	0.81	22.09
3.9	50	39	0.34	21.87	0.40	21.98

**Tableau 5.4. Fréquences de vibration en fonction de la charge  $m_p$**

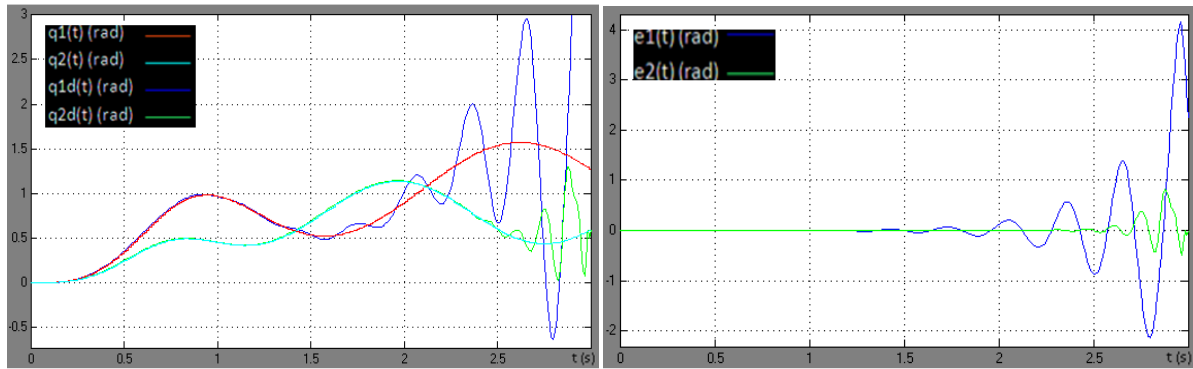
$M_{L1}$  est la charge totale supportée par l'axe 1.

$M_{L2}$  est la charge totale supportée par l'axe2.

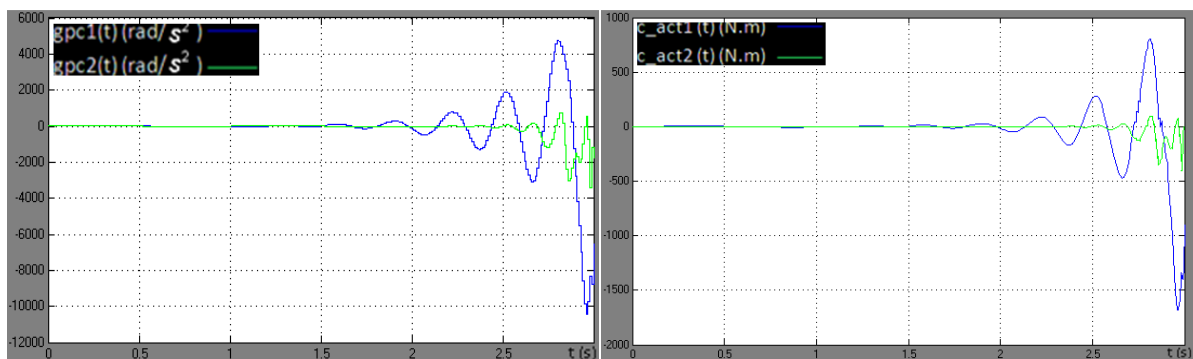
$f_{ij}$  est la  $j^{\text{ème}}$  fréquence naturelle correspondant à l'axe  $i$ .

Nous remarquons que l'augmentation de la charge  $m_p$  induit une diminution des fréquences de vibration. Ce qui correspond bien à une augmentation de la flexibilité.

Nous allons à présent présenter les résultats de simulation sans compensation pour le manipulateur flexible. Les figures présentées ci-dessous illustrent le cas avec une augmentation de la charge :  $m_p=0.2$  kg.



**Figure 5.12. Suivi et erreurs de suivi sans compensation floue pour  $m_p=0.2$  kg**



**Figure 5.13. Commandes GPC1 et GPC2 et commandes actionneurs sans compensation floue pour  $m_p=0.2$  kg**

Nous pouvons voir que la commande est de très mauvaise qualité. Nous observons l'apparition d'oscillations croissantes très peu de temps après le lancement de la commande.

Nous allons maintenant, présenter les courbes relatives à la simulation de la commande du bras manipulateur flexible pour différentes valeurs de la charge. Nous pouvons voir sur les figures suivantes : le suivi de trajectoire ( $q_1$ ,  $q_2$ ), l'erreur de suivi ( $e_1$ ,  $e_2$ ), les signaux de commande GPC ( $gpc_1$ ,  $gpc_2$ ) ainsi que les couples délivrés par les actionneurs ( $c_{act1}$ ,  $c_{act2}$ ).

Le temps de simulation est de 5s pour un temps d'échantillonnage  $T_e=10$ ms.

$m_p=0.1\text{kg}$

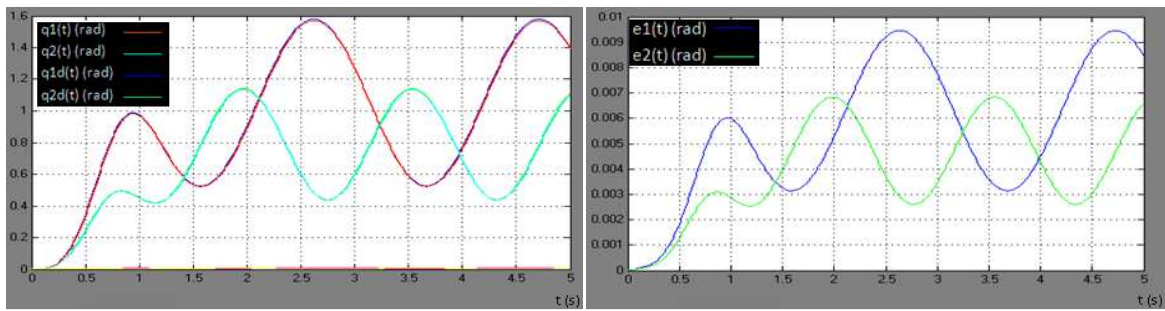


Figure 5.14. Suivi et erreurs de suivi pour  $m_p=0.1\text{ kg}$

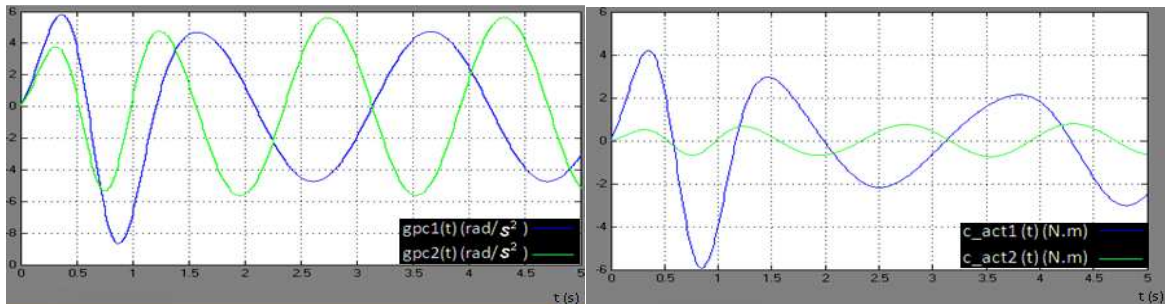


Figure 5.15. Commandes GPC1 et GPC2 et commandes actionneurs pour  $m_p=0.1\text{ kg}$

$m_p=0.4\text{kg}$

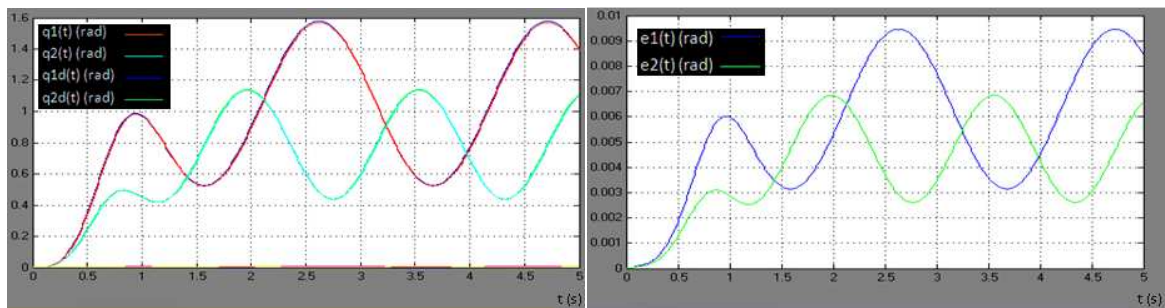


Figure 5.16. Suivi et erreurs de suivi pour  $m_p=0.4\text{ kg}$

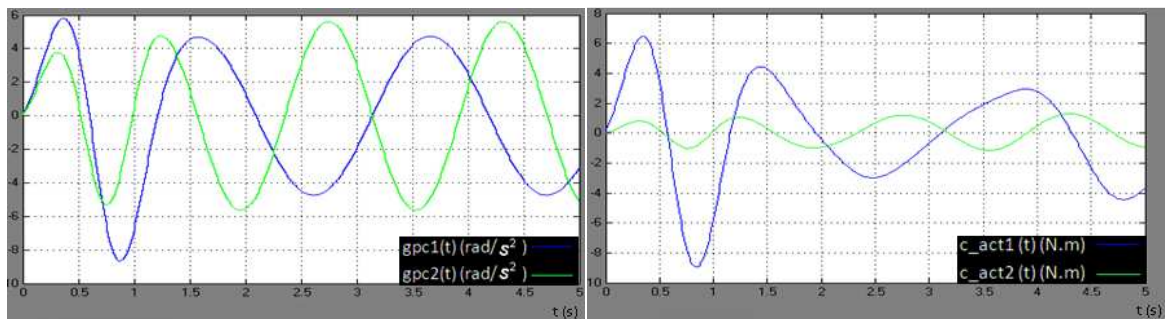
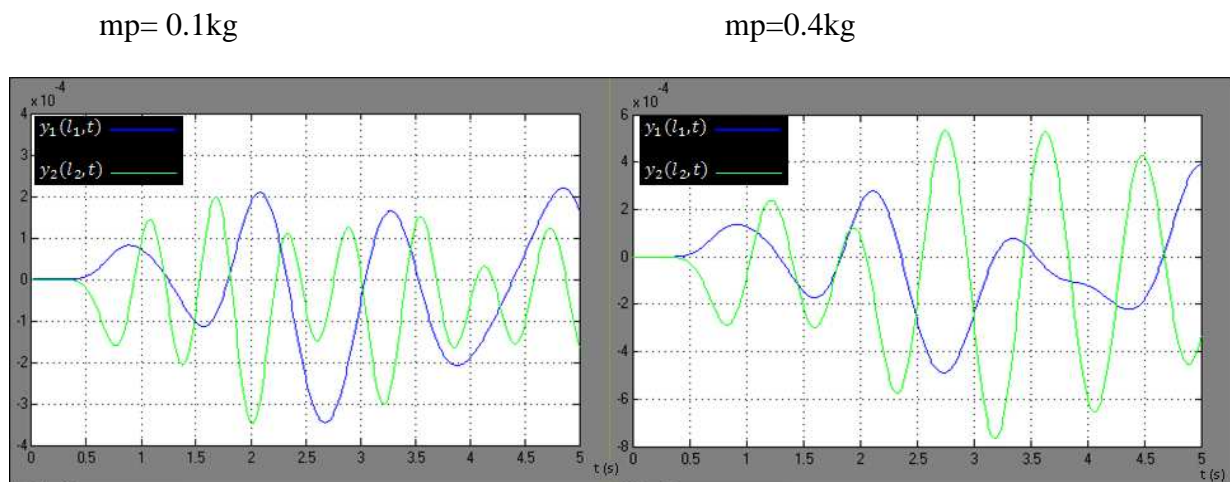


Figure 5.17. Commandes GPC1 et GPC2 et commandes actionneurs pour  $m_p=0.4\text{kg}$

Nous avons pu constater à travers les différents résultats obtenus, que la qualité du suivi reste la même pour les différentes valeurs de la charge. L'erreur de suivi ne se dégrade pas.

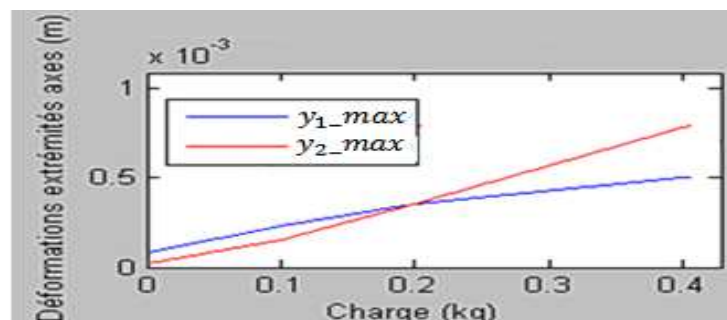
Par contre, nous observons une augmentation des couples de commande délivrés par les actionneurs avec l'augmentation de la charge.

Nous allons à présent, nous intéresser à la déformation des axes flexibles du robot durant l'accomplissement de la tâche. Nous présentons les courbes  $y_1(l_1, t)$  et  $y_2(l_2, t)$  représentant, respectivement, les déviations des extrémités des axes 1 et 2 en fonction du temps



**Figure 5.18. Déviation des extrémités des axes 1 et 2 pour  $m_p=0.2$  kg et  $m_p=0.4$ kg**

Nous pouvons constater une augmentation de l'amplitude des déviations des extrémités des axes en conséquence de l'augmentation de la charge et donc de l'augmentation de la flexibilité. Cette augmentation est illustrée sur la figure suivante :



**Figure 5.19. Les déviations maximales  $y_{1\_max}$  et  $y_{2\_max}$  des extrémités des axes 1 et 2 en fonction de  $m_p$**

Nous remarquons que pour une charge inférieure à 0.4 kg soit un rapport charge/masse  $axe2 = 4$ , les performances de la commande sont satisfaisantes :

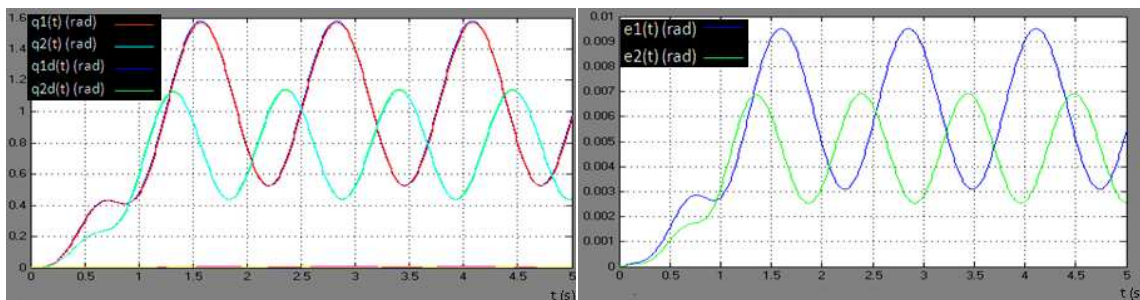
Erreur de suivi max axe 1	0.0095 rad
Erreur de suivi max axe 2	0.0068 rad
Couple max actionneur 1	9 N.m
Couple max actionneur 2	1.2 N.m
Déviation max extrémité axe 1	0.00005 m
Déviation max extrémité axe 2	0.00078 m

**Tableau 5.5. Performances de la commande pour  $mp=0.4$**

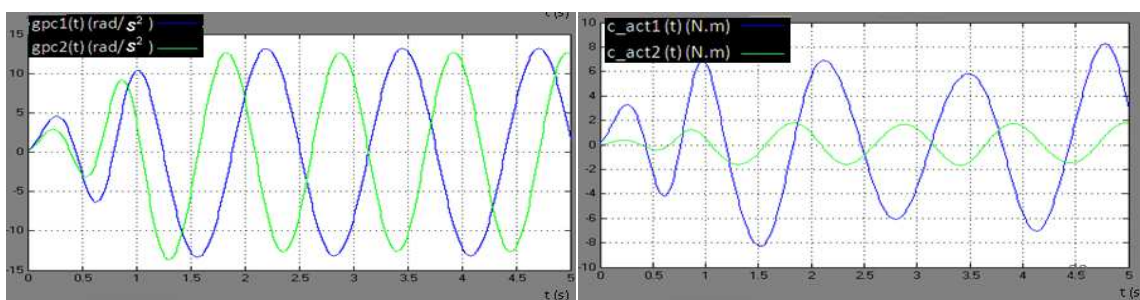
### 5.4.2. Augmentation de la vitesse de travail

Nous allons, dans ce qui suit, présenter les courbes relatives à la simulation de la commande du bras manipulateur pour différentes valeurs de  $\omega_1$  et  $\omega_2$  dans les signaux de référence (voir chapitre 3).

$\omega_1 = 5 \text{ rad/s}$  et  $\omega_2 = 6 \text{ rad/s}$

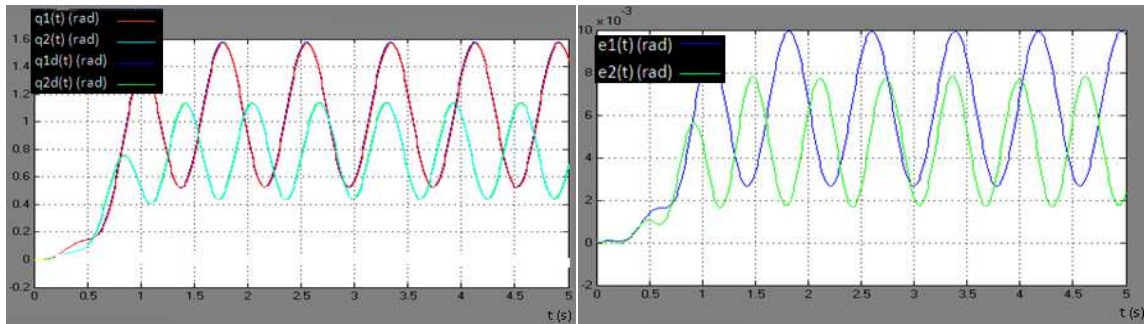


**Figure 5.20. Suivi et erreurs de suivi pour  $\omega_1 = 5 \text{ rad/s}$  et  $\omega_2 = 6 \text{ rad/s}$**

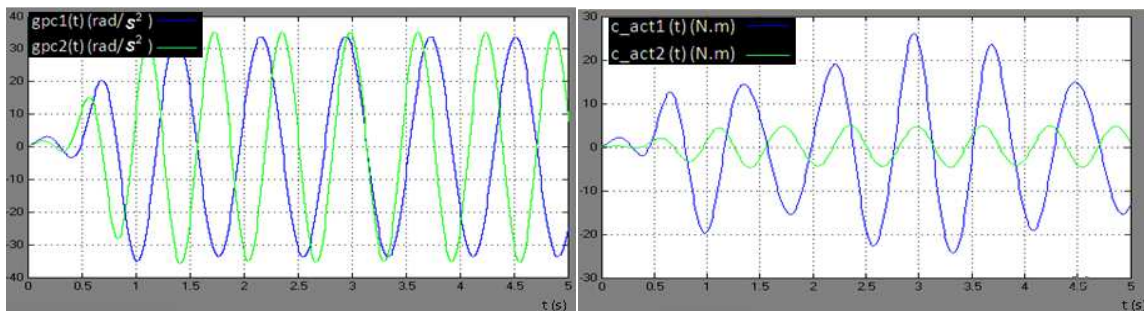


**Figure 5.21. Commandes GPC1 et GPC2 et commandes actionneurs pour  $\omega_1 = 5 \text{ rad/s}$  et  $\omega_2 = 6 \text{ rad/s}$**

$$\omega_1 = 8 \text{ rad/s et } \omega_2 = 10 \text{ rad/s}$$



**Figure 5.22. Suivi et erreurs de suivi pour  $\omega_1 = 8 \text{ rad/s}$  et  $\omega_2 = 10 \text{ rad/s}$**



**Figure 5.23. Commandes GPC1 et GPC2 et commandes actionneurs pour  $\omega_1 = 8 \text{ rad/s}$  et  $\omega_2 = 10 \text{ rad/s}$**

La dégradation des performances de la commande est visible quand nous augmentons la vitesse de travail du robot.

Le signal délivré par la commande GPC augmente sensiblement.

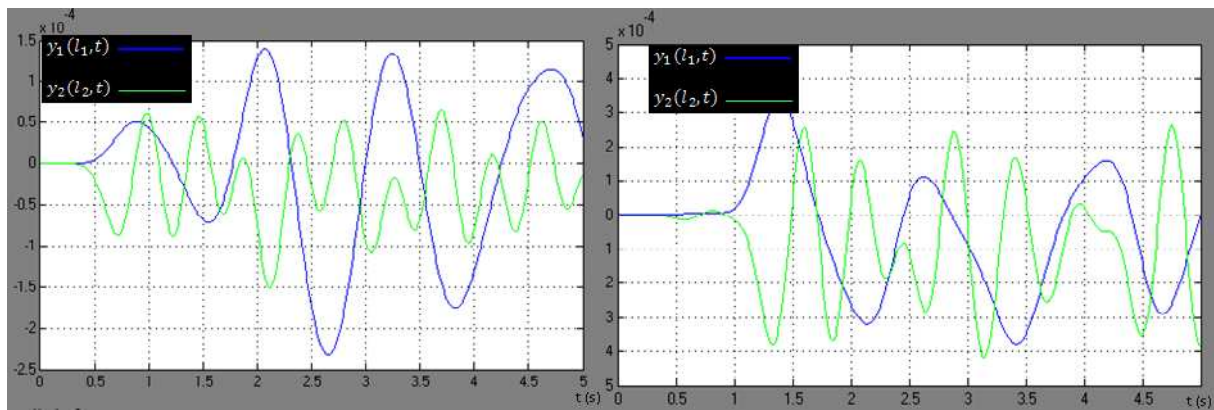
Les couples appliqués au niveau des articulations augmentent eux aussi sensiblement.

L'erreur de suivi est plus importante.

Nous allons à présent, nous intéresser à la déformation des axes flexibles du robot durant l'accomplissement de la tâche. Nous présentons les courbes  $y_1(l_1, t)$  et  $y_2(l_2, t)$  représentant, respectivement, les déviations des extrémités des axes 1 et 2 en fonction du temps.

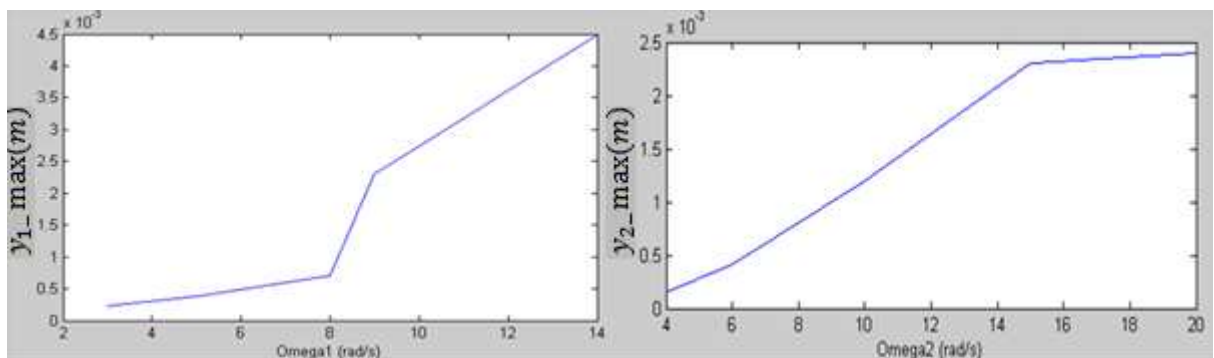
$\omega_1 = 5 \text{ rad/s}$  et  $\omega_2 = 6 \text{ rad/s}$

$\omega_1 = 8 \text{ rad/s}$  et  $\omega_2 = 10 \text{ rad/s}$



**Figure 5.24. Déviation des extrémités des axes 1 et 2 pour différentes valeurs de  $\omega_1$  et  $\omega_2$**

Nous constatons l'augmentation de l'amplitude des déviations des extrémités des axes en conséquence de l'augmentation de la vitesse de travail. Les simulations effectuées permettent d'illustrer cette augmentation sur les figures suivantes :



**Figure 5.25. Les déviations maximales  $y_{1\_max}$  et  $y_{2\_max}$  des extrémités des axes 1 et 2 en fonction de  $\omega_1$  et  $\omega_2$**

Nous pouvons remarquer que pour  $\omega_1 < 8 \text{ rad/s}$ ,  $\omega_2 < 9 \text{ rad/s}$ , les performances de la commande sont satisfaisantes :

Erreur de suivi max axe 1	0.01 rad
Erreur de suivi max axe 2	0.008 rad
Couple max actionneur 1	25 N.m
Couple max actionneur 2	5 N.m
Déviation max extrémité axe 1	0.0007 m
Déviation max extrémité axe 2	0.001 m

**Tableau 5.6. Performances pour  $\omega_1 = 8$  rad/s,  $\omega_2 = 9$  rad/s**

En conclusion, la commande réagit assez bien à une augmentation raisonnable de la flexibilité du robot.

## 5.5. Conclusion

Dans ce chapitre, nous nous sommes intéressés au problème de la linéarisation des robots manipulateurs avec incertitudes. Un compensateur flou, capable de corriger les erreurs dues aux incertitudes, a été proposé.

La commande par couple calculé est présentée et ses limites en présence d'incertitudes démontrées. Enfin, le compensateur flou proposé pour compenser les incertitudes est décrit.

Nous avons effectué des tests de simulation sur le robot manipulateur rigide à 2 ddl rotoïdes. D'abord en présence d'incertitudes structurées que nous avons simulées en variant les masses des axes du robot. Ensuite, avec des incertitudes non structurées. Les tests ont montré un bon fonctionnement de la commande en présence de ces différentes perturbations.

La commande conçue pour un robot rigide a également été testée sur robot flexible. Nous avons augmenté la flexibilité du robot et surveillé le suivi de trajectoire et les déviations des axes flexibles. La commande a montré des résultats acceptables.

Le principal apport du présent chapitre par rapport aux travaux présent dans la littérature a été l'application de la commande linéarisante à un robot flexible.

# Chapitre 6

---

## Méthode des Perturbations Singulières

## 6.1 Introduction

Les bras manipulateurs flexibles sont des systèmes hautement non linéaires et difficiles à commander.

Plusieurs stratégies de commande ont été employées dans la commande des manipulateurs flexibles. Citons à titre d'exemple quelques méthodes qui reviennent fréquemment dans la littérature : la commande optimale [36, 37], la commande par surfaces glissantes[38, 39], la commande par inversion du modèle[8, 15, 40], la commande basée sur l'énergie[39, 41], la méthode des perturbations singulières[40, 42]...etc. Toutes ces méthodes de commande ont été appliquées sur un modèle à dimension finie et donc tronqué du robot flexible comme par exemple celui présenté dans le chapitre 2 où seuls les deux premiers modes de vibration de chaque axe flexible sont pris en considération.

Dans le présent chapitre, nous allons utiliser la méthode des perturbations singulières. Celle-ci permet une séparation du modèle du robot flexible en un sous-modèle rigide et un sous-modèle flexible. Ce qui permet d'utiliser les stratégies de commande développées tout au long de ce travail pour commander le sous-système rigide.

## 6.2 Manipulateur flexible singulièrement perturbé

Le manipulateur flexible est un système à deux échelles de temps lent/rapide[43]. Il existe une séparation naturelle entre les échelles de temps de la dynamique flexible et de la dynamique rigide. Les coordonnées modales  $\delta$  varient beaucoup plus rapidement que les variables articulaires  $q$ . En utilisant la méthode des perturbations singulières, le robot flexible peut être modélisé à l'aide de deux sous-modèles à ordre réduit[44]. En découplant les variables rapides  $\delta$  des variables lentes  $q$ , nous obtenons un sous-modèle lent représentant la dynamique rigide et un sous-modèle rapide représentant la dynamique flexible. L'intérêt de la méthode étant de pouvoir déterminer une commande sous forme composite pour le système global[40, 42].

D'après le chapitre 2, l'équation de mouvement d'un manipulateur à  $n$  axes flexibles s'écrit comme suit :

$$\begin{bmatrix} B_{qq}(q, \delta) & B_{q\delta}(q, \delta) \\ B_{q\delta}^T(q, \delta) & B_{\delta\delta}(q, \delta) \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} h_q(q, \delta, \dot{q}, \dot{\delta}) \\ h_\delta(q, \delta, \dot{q}, \dot{\delta}) \end{bmatrix} + \begin{bmatrix} 0 \\ K\delta \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix} \quad (6.1)$$

Où  $q$  est le vecteur  $n \times 1$  des variables articulaires et  $\delta$  le vecteur  $m \times 1$  des coordonnées modales.

$B$  est la matrice d'inertie,  $h_q$  et  $h_\delta$  sont deux termes dus aux forces centrifuge et de Coriolis,  $K$  est la matrice de raideur.  $u$  est un vecteur de longueur  $n$  composé des couples actionneurs appliqués au niveau des articulations.

La longueur du vecteur de coordonnées modales  $\delta$  est égale à  $m$  tel que  $m = \sum_{i=1}^n m_i$ .

$m_i$  est le nombre de coordonnées modales prises en compte pour modéliser la déformation de l'axe  $i$ .

Puisque la matrice d'inertie  $B$  est définie positive, son inverse  $B^{-1}$  existe :

$$D = B^{-1} = \begin{bmatrix} D_{qq} & D_{q\delta} \\ D_{\delta q} & D_{\delta\delta} \end{bmatrix} \quad (6.2)$$

L'équation (6.1) devient :

$$\ddot{q} = -D_{qq}h_q - D_{q\delta}h_\delta - D_{q\delta}K\delta + D_{qq}u \quad (6.3)$$

$$\ddot{\delta} = -D_{\delta q}h_q - D_{\delta\delta}h_\delta - D_{\delta\delta}K\delta + D_{\delta q}u \quad (6.4)$$

Introduisons le facteur  $k$  tel que [45]:

$$K = k\tilde{K} \quad (6.5)$$

$K$  étant la matrice de raideur.

Soit la nouvelle variable  $\zeta$ . Nous allons effectuer le changement de variable suivant :

$$\zeta = k\tilde{K}\delta \quad (6.6)$$

En posant  $\epsilon^2 = 1/k$ , nous pouvons réécrire les équations (6.3) et (6.4) comme suit[44] :

$$\begin{aligned} \ddot{q} = & -D_{qq}(q, \epsilon^2 \zeta) h_q(q, \dot{q}, \epsilon^2 \zeta, \epsilon^2 \dot{\zeta}) \\ & - D_{q\delta}(q, \epsilon^2 \zeta) h_\delta(q, \dot{q}, \epsilon^2 \zeta, \epsilon^2 \dot{\zeta}) - D_{q\delta}(q, \epsilon^2 \zeta) \zeta \\ & + D_{qq}(q, \epsilon^2 \zeta) u \end{aligned} \quad (6.7)$$

$$\begin{aligned} \epsilon^2 \ddot{\zeta} = & -D_{\delta q}(q, \epsilon^2 \zeta) h_\theta(q, \dot{q}, \epsilon^2 \zeta, \epsilon^2 \dot{\zeta}) \\ & - D_{\delta\delta}(q, \epsilon^2 \zeta) h_\delta(q, \dot{q}, \epsilon^2 \zeta, \epsilon^2 \dot{\zeta}) - D_{\delta\delta}(q, \epsilon^2 \zeta) \zeta \\ & + D_{\delta q}(q, \epsilon^2 \zeta) u \end{aligned} \quad (6.8)$$

C'est le modèle singulièrement perturbé du manipulateur flexible.

En choisissant  $x_1 = q$  et  $x_2 = \dot{q}$  et  $z_1 = \zeta$ ,  $z_2 = \epsilon \dot{\zeta}$ , nous aboutissons à la représentation d'état du système (6.7) et (6.8) :

$$\dot{x}_1 = x_2$$

$$\begin{aligned} \dot{x}_2 = & -D_{qq}(x_1, \epsilon^2 z_1) h_q(x_1, x_2, \epsilon^2 z_1, \epsilon z_2) \\ & - D_{q\delta}(x_1, \epsilon^2 z_1) h_\delta(x_1, x_2, \epsilon^2 z_1, \epsilon z_2) \\ & - D_{q\delta}(x_1, \epsilon^2 z_1) z_1 + D_{qq}(x_1, \epsilon^2 z_1) u \end{aligned} \quad (6.9)$$

$$\epsilon \dot{z}_1 = z_2$$

$$\begin{aligned} \epsilon \dot{z}_2 = & -D_{\delta q}(x_1, \epsilon^2 z_1) h_\theta(x_1, x_2, \epsilon^2 z_1, \epsilon z_2) \\ & - D_{\delta\delta}(x_1, \epsilon^2 z_1) h_\delta(x_1, x_2, \epsilon^2 z_1, \epsilon z_2) \\ & - D_{\delta\delta}(x_1, \epsilon^2 z_1) z_1 + D_{\delta q}(x_1, \epsilon^2 z_1) u \end{aligned} \quad (6.10)$$

Procédons à présent à l'identification des deux sous-systèmes lent et rapide.

Nous obtenons le modèle du sous-système lent en mettant  $\epsilon = 0$  et en résolvant l'équation (6.8) pour  $\zeta$ :

$$\begin{aligned} \bar{\zeta} = & D_{\delta\delta}^{-1}(\bar{q}, 0) [-D_{\delta q}(\bar{q}, 0)h_{\theta}(\bar{q}, \dot{\bar{q}}, 0, 0) + D_{\delta q}(\bar{q}, 0)\bar{u}] \\ & - h_{\delta}(\bar{q}, \dot{\bar{q}}, 0, 0) \end{aligned} \quad (6.11)$$

Où la barre supérieure indique que  $\epsilon = 0$ .

En substituant l'équation (6.11) dans l'équation (6.7), toujours avec  $\epsilon = 0$ , nous obtenons :

$$\begin{aligned} \ddot{q} = & [D_{qq}(\bar{q}, 0) \\ & - D_{q\delta}(\bar{q}, 0)D_{\delta\delta}^{-1}(\bar{q}, 0)D_{\delta q}(\bar{q}, 0)] [-h_q(\bar{q}, \dot{\bar{q}}, 0, 0) \\ & + \bar{u}] \end{aligned} \quad (6.12)$$

A noter que :

$$D_{qq}(\bar{q}, 0) - D_{q\delta}(\bar{q}, 0)D_{\delta\delta}^{-1}(\bar{q}, 0)D_{\delta q}(\bar{q}, 0) = B_{qq}^{-1}(\bar{q}) \quad (6.13)$$

Alors, l'équation (6.12) devient :

$$\ddot{q} = B_{qq}^{-1}(\bar{q}) [-h_q(\bar{q}, \dot{\bar{q}}, 0, 0) + \bar{u}] \quad (6.14)$$

Avec le changement de variable, nous obtenons la représentation d'état suivante pour le sous-système lent :

$$\begin{aligned} \dot{\bar{x}}_1 &= \bar{x}_2 \\ \dot{\bar{x}}_2 &= B_{qq}^{-1}(\bar{x}_1) [-h_q(\bar{x}_1, \bar{x}_2) + \bar{u}] \end{aligned} \quad (6.15)$$

Pour trouver l'équation du sous-système rapide, nous introduisons une nouvelle échelle de temps  $\tau = t/\epsilon$ . Le système devient [40, 46]:

$$\dot{x}_1 = x_2$$

$$\begin{aligned} \dot{x}_2 = & -D_{qq}(\bar{x}_1, \epsilon^2(\eta_1 + \bar{\zeta}))h_q(x_1, x_2, \epsilon^2(\eta_1 + \bar{\zeta}), \epsilon\eta_2) \\ & - D_{q\delta}(x_1, \epsilon^2(\eta_1 + \bar{\zeta}))h_\delta(x_1, x_2, \epsilon^2(\eta_1 + \bar{\zeta}), \epsilon\eta_2) \\ & - D_{q\delta}(x_1, \epsilon^2(\eta_1 + \bar{\zeta}))(\eta_1 + \bar{\zeta}) \\ & + D_{qq}(x_1, \epsilon^2(\eta_1 + \bar{\zeta}))u \end{aligned} \quad (6.16)$$

$$\dot{\eta}_1 = \eta_2$$

$$\begin{aligned} \epsilon\dot{\eta}_2 = & -D_{\delta q}(x_1, \epsilon^2(\eta_1 + \bar{\zeta}))h_\theta(x_1, x_2, \epsilon^2(\eta_1 + \bar{\zeta}), \epsilon\eta_2) \\ & - D_{\delta\delta}(x_1, \epsilon^2(\eta_1 + \bar{\zeta}))h_\delta(x_1, x_2, \epsilon^2(\eta_1 + \bar{\zeta}), \epsilon\eta_2) \\ & - D_{\delta\delta}(x_1, \epsilon^2(\eta_1 + \bar{\zeta}))(\eta_1 + \bar{\zeta}) \\ & + D_{\delta q}(x_1, \epsilon^2(\eta_1 + \bar{\zeta}))u \end{aligned} \quad (6.17)$$

Où les variables rapides  $\eta_1$  et  $\eta_2$  sont définies par :

$$\eta_1 = z_1 - \bar{\zeta} = z_1 - \bar{z}_1$$

Et

$$\eta_2 = z_2$$

Par rapport à la nouvelle échelle de temps, les variables lentes  $x_1$  et  $x_2$  sont considérées comme des constantes.

$h_q(x_1, x_2, 0, 0)$  et  $h_\delta(x_1, x_2, 0, 0)$  sont supposés égaux à zéro puisque ces termes sont composés de produits de  $x_1$  et  $x_2$  avec  $\epsilon^2 z_1$  et  $\epsilon z_2$  [47].

Le sous-système rapide s'écrit :

$$\frac{d\eta_1}{d\tau} = \eta_2$$

$$\frac{d\eta_2}{d\tau} = -D_{\delta\delta}(\bar{x}_1, 0)\eta_1 + D_{\delta q}(\bar{x}_1, 0)u_f \quad (6.18)$$

C'est un système linéaire avec  $\bar{x}_1$  comme paramètre.

Le signal de commande  $u_f = u - \bar{u}$ .

### 6.3 Commande du manipulateur flexible avec perturbation singulière

Dans le paragraphe précédent, nous avons vu qu'en utilisant la méthode des perturbations singulières, le robot flexible peut être représenté à l'aide de deux sous-modèles, l'un représentant la dynamique rigide lente et l'autre la dynamique flexible rapide. Le but de cette séparation est de pouvoir commander indépendamment les deux sous-systèmes. La stratégie de commande vise à imposer un suivi de trajectoire  $q_d$  à la sortie  $q$  du sous-système rigide et d'amortir les vibrations dues aux mouvements flexibles.

Le signal de commande  $u$  appliqué au manipulateur flexible est composé de deux commandes indépendantes  $\bar{u}$  et  $u_f$  telles que [40, 44] :

$$u = \bar{u}(\bar{x}_1, \bar{x}_2) + u_f(\bar{x}_1, \eta_1, \eta_2) \quad (6.19)$$

$\bar{u}$  étant la commande destinée à imposer un suivi de trajectoire au sous-système rigide et  $u_f$  la commande visant à amortir les mouvements du sous-système flexible.

Avec la contrainte :  $u_f(\bar{x}_1, 0, 0) = 0$  de façon à ce que  $u_f$  soit inactive le long de la solution de  $\bar{\eta}$ .  $\bar{\eta}$  est une trajectoire d'équilibre du système (6.8) [46]

#### 6.3.1. Sous-commande lente

Le but de la sous-commande lente est d'imposer un suivi de trajectoire aux variables articulaires. Afin de commander le sous système lent, les différentes stratégies utilisées pour la commande des robots manipulateurs rigides peuvent être appliquées [13, 40].

Le sous-système lent s'écrit sous la forme :

$$\dot{\bar{x}}_1 = \bar{x}_2$$

$$\dot{\bar{x}}_2 = B_{qq}^{-1}(\bar{x}_1)[-h_q(\bar{x}_1, \bar{x}_2) + \bar{u}] \quad (6.20)$$

Nous allons appliquer ici les différentes stratégies de commande élaborée lors des chapitres précédents pour la commande des manipulateurs rigides.

### 6.3.2. Sous-commande rapide

La sous-commande rapide a pour rôle de stabiliser les déflexions des modes flexibles autour de la position d'équilibre  $\bar{\zeta}$ .

Le sous-système rapide s'écrit sous la forme :

$$\frac{d\eta}{d\tau} = A\eta + Bu_f \quad (6.21)$$

$$u_f = u - \bar{u}$$

$$A = \begin{bmatrix} 0 & I \\ -D_{\delta\delta}(\bar{x}_1, 0) & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ D_{\delta q}(\bar{x}_1, 0) \end{bmatrix}$$

La méthode de la perturbation singulière exige que la couche limite soit uniformément stable le long de la trajectoire d'équilibre  $\bar{\zeta}$ . Ceci est vrai si la paire

$$A = \begin{bmatrix} 0 & I \\ -D_{\delta\delta} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ D_{\delta q} \end{bmatrix}$$

est uniformément stabilisable pour toute trajectoire lente  $\bar{x}_1(t)$ [47]. Ceci est habituellement vrai pour les manipulateurs flexibles [44].

Plusieurs stratégies de commande telles que la commande LQR peuvent être appliquées afin de concevoir la commande  $u_f$ .

Un retour d'état de la forme [44]:

$$u_f(\bar{x}_1, \eta_1, \eta_2) = K_{f1}(\bar{x}_1)\eta_1 + K_{f2}(\bar{x}_1)\eta_2 \quad (6.22)$$

Où  $K_{f1}$  et  $K_{f2}$  sont conçus en utilisant la commande LQR, stabilise (6.21) à  $\eta_1 = 0$  et  $\eta_2 = 0$ .

Sous ces conditions, le théorème de Tikhonov, un théorème fondamental dans la méthode de la perturbation singulière, garantit que les variables d'état du système global peuvent être approchées par :

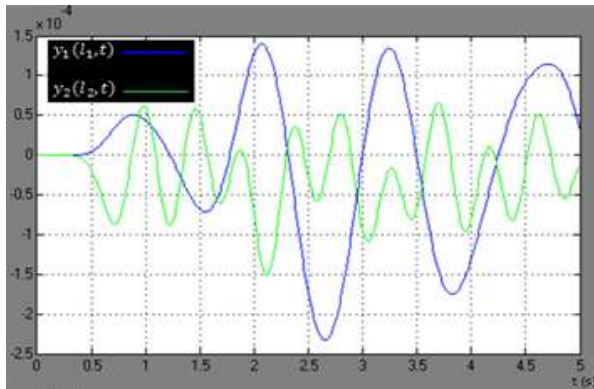
$$\begin{aligned}
 x_1 &= \bar{x}_1 + O(\epsilon) \\
 x_2 &= \bar{x}_2 + O(\epsilon) \\
 z_1 &= \bar{\zeta}_1 + \eta_1 + O(\epsilon) \\
 z_2 &= \eta_2 + O(\epsilon)
 \end{aligned}
 \tag{6.23}$$

### 6.4 Résultats et simulations

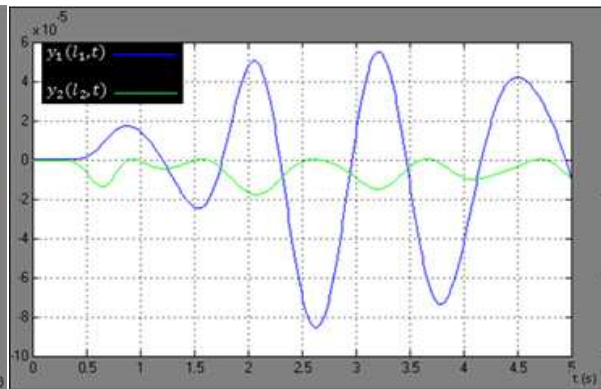
Nous allons tester la commande composite sur le modèle du robot manipulateur flexible. Le sous-système rigide sera commandé en utilisant la commande GPC du chapitre 3. Le sous-système flexible sera commandé en utilisant la commande LQR.

Nous pouvons voir sur les figures suivantes le résultat de la simulation de la commande composite appliquée au robot manipulateur flexible. Celle-ci est comparée au cas où seul le sous-système rigide est commandé.

**Commande rigide**

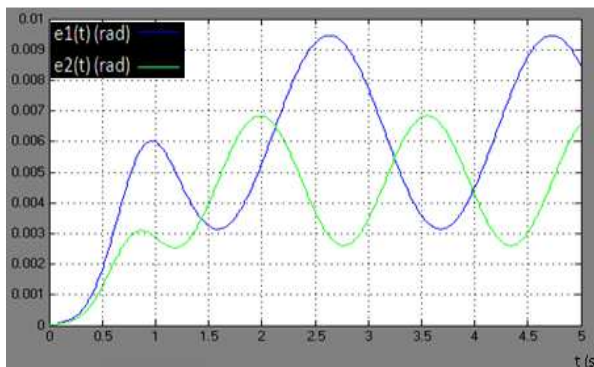


**Commande composite**

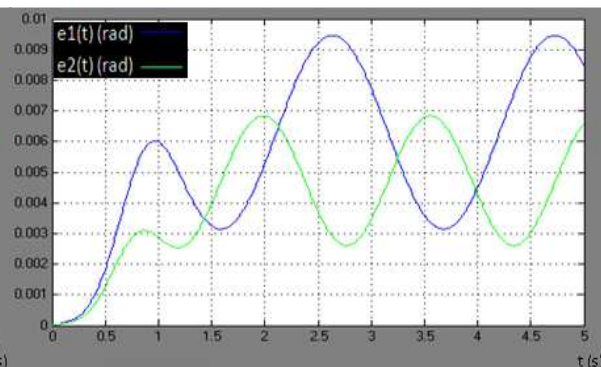


**Figure 6.1. Déviation des extrémités des axes 1 et 2 avec commande rigide (à gauche) et commande composite (à droite).**

**Commande rigide**



**Commande composite**



**Figure 6.2. Erreurs de suivi e1 et e2 avec commande rigide (à gauche) et commande composite (à droite).**

Les figures montrent une atténuation de l'amplitude de la déformation des extrémités des axes du robot. L'erreur de poursuite reste la même.

## 6.5 Conclusion

Afin de pouvoir tester, les stratégies de commande développée pour un robot manipulateur rigide sur un manipulateur flexible. Nous avons appliqué à ce dernier la théorie des perturbations singulières. Celle-ci permet de décomposer le modèle du robot flexible en un sous-modèle rigide et un sous-modèle flexible donnant ainsi la possibilité de commander séparément les deux sous-systèmes à l'aide d'une commande composite.

Nous avons utilisé la commande GPC afin d'assurer le suivi de trajectoire du sous-système rigide et la commande LQR pour amortir les mouvements du sous-système flexible.

Le principal apport de ce chapitre a été de permettre l'application des différentes commandes présentée dans ce mémoire à un robot manipulateur flexible sans rien changer dans les commandes elle-même.

Les résultats obtenus en simulation ont montré l'atténuation de la déformation des axes flexibles et un bon suivi de trajectoire.

**Conclusion et**

**Perspectives**

---

## Conclusion et perspectives

L'objet du présent travail, était de montrer les avantages que peut apporter l'association de la commande prédictive et de la logique floue à l'exécution d'une tâche par un robot manipulateur.

L'application d'une boucle de commande prédictive pour la commande de notre robot rigide, nous a permis de profiter de son caractère anticipatif lors du suivi de trajectoire. De plus, Pour le cas sans contrainte, la commande GPC permet d'obtenir une loi programmable par une simple équation aux différences, ce qui nous a évité une optimisation en ligne (toujours coûteuse). Cette représentation sous forme polynomiale nous a permis d'analyser la stabilité de la commande en utilisant les outils de l'automatique de base.

Les performances de la commande GPC dépendent du choix des paramètres de synthèse : horizon de commande  $N_u$ , horizon de prédiction  $N_2$  ainsi que de la pondération [18] sur les incréments de commande  $\lambda$ . Il n'est pas souvent évident de définir des valeurs adéquates pour ces paramètres. Afin de régler ce problème, nous avons introduit un superviseur flou capable de surveiller les performances de la boucle de commande GPC et d'ajuster les paramètres de synthèse.

Une autre difficulté liée à l'application de la commande GPC est que cette dernière est une commande linéaire. Son application à la commande du robot manipulateur rigide requiert la linéarisation du modèle de celui-ci. Or la linéarisation exige une parfaite connaissance du modèle du robot. Ce qui est pratiquement impossible. Pour pallier à ce problème, nous avons exploité la propriété d'approximateurs universels des systèmes flous afin d'estimer les incertitudes structurées et non-structurées entachant notre connaissance du modèle du robot manipulateur et pouvoir ainsi les compenser.

Enfin, nous avons souhaité appliquer ces stratégies de commande à un manipulateur à axes flexibles. La théorie des perturbations singulières nous a permis de décomposer le modèle du manipulateur flexible en deux sous-modèles : un sous-modèle rigide et un sous-modèle flexible. Afin d'assurer le suivi de trajectoire, nous avons appliqué au sous-modèle rigide une boucle de commande GPC avec compensateur flou. Une boucle de commande optimale a été utilisée afin d'amortir les mouvements dus à la flexibilité.

Les tests réalisés en simulation nous ont permis d'évaluer les performances des différentes stratégies de commande appliquées aux manipulateurs rigide et flexible. Nous avons constaté l'amélioration des performances de la commande GPC suite à l'ajustement des paramètres de synthèse. Nous avons augmenté la valeur des perturbations subies par les robots rigide ainsi que la flexibilité des axes du manipulateur flexible afin de tester les limites et la robustesse de la commande GPC avec compensateur flou. Les résultats obtenus montrent d'assez bons résultats pour ces stratégies de commande.

La prochaine étape de ce travail, devrait être la prise en compte des contraintes. La commande prédictive sous contraintes constitue un domaine de recherche très actif. La décennie précédente a vu l'apparition de la commande prédictive explicite [18, 48, 49]. Afin de diminuer la complexité inhérente aux procédures d'optimisation en ligne, La commande prédictive explicite donne accès à une multitude de problèmes (plus simples) d'optimisation par morceaux. Le but consiste à élaborer pour chaque partition une fonction simple à évaluer, de sorte que le recours aux solveurs en ligne pour les problèmes d'optimisations ne soit plus nécessaire.

La loi de commande GPC avec contraintes peut alors être structurée sous forme polynomiale à l'aide de régulateurs 'RST affines' définis pour des régions polyédrales. Il devient possible d'appliquer les méthodes développées dans le cas d'une loi de commande prédictive polynomiale sans contrainte[50].

La logique floue est aussi présente dans le domaine de la commande prédictive explicite [51, 52].

# **ANNEXE I**

---

## **Notions de base sur la logique floue**

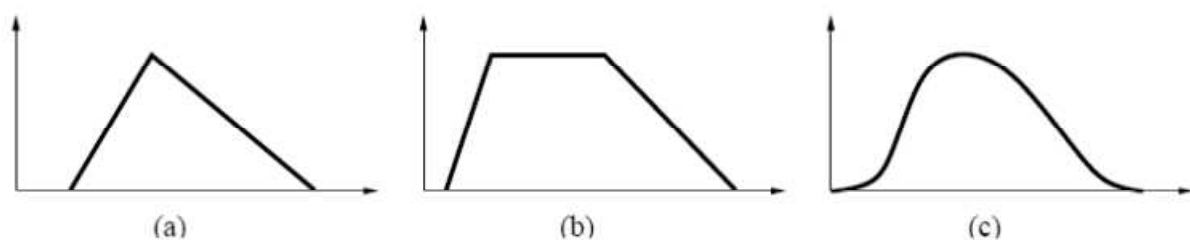
Cette annexe introduit quelques notions de logique floue utilisées dans la suite du chapitre. Nous trouvons dans la littérature quantité d'ouvrages dédiés à la logique floue [53,54].

## 1. Ensembles flous

La théorie des ensembles flous repose sur la notion d'appartenance partielle : chaque élément appartient partiellement à un ensemble flou défini [55].

Un ensemble flou est défini par sa fonction d'appartenance. Une fonction d'appartenance  $\mu_A(x)$  fait correspondre à toute valeur de la variable d'entrée  $x$ , appartenant à l'univers de discours  $U$ , son degré d'appartenance à l'ensemble  $A$ . Cette valeur graduelle est comprise entre 0 et 1.

Les fonctions d'appartenance peuvent prendre n'importe quelle forme. Dans la pratique, les formes les plus utilisées sont la forme triangulaire, la forme trapézoïdale, et les fonctions sous forme de cloche, ces formes sont représentées sur la figure A.1.



**Figure A.1. Formes les plus courantes des fonctions d'appartenance**

Plusieurs opérations peuvent être appliquées aux ensembles flous. Le complément d'un ensemble flou correspondant à l'opérateur « négation », l'intersection correspondant à l'opérateur « ET », l'union de deux ensembles flous correspondant à l'opérateur « OU »...etc. Ces opérateurs permettent d'écrire des combinaisons logiques entre notions floues.

## 2. Base de règles floues

La base de règles floues se compose d'une collection de règles floues de type SI-Alors (SI prédicat ALORS implication). Ces règles peuvent être exprimées de différentes manières[56].

Sous le modèle Mamdani, les règles floues sont de la forme :

$$\begin{aligned} R^i: & \text{ SI } x_1 \text{ est } A_1^i \text{ ET } x_2 \text{ est } A_2^i \text{ ET } \dots \text{ ET } x_n \text{ est } A_n^i \\ & \text{ ALORS } y_1 \text{ est } B_1^i \text{ ET } y_2 \text{ est } B_2^i \text{ ET } \dots \text{ ET } y_m \text{ est } B_m^i \end{aligned} \quad (A.1)$$

Les règles peuvent aussi être exprimées sous le modèle Takagi-Sugeno (TS) de la façon suivante :

$$\begin{aligned} R^i: & \text{ SI } x_1 \text{ est } A_1^i \text{ ET } x_2 \text{ est } A_2^i \text{ ET } \dots \text{ ET } x_n \text{ est } A_n^i \\ & \text{ ALORS } y_1 = f_{1i}(x_1, x_2, \dots, x_n) \text{ ET } y_2 = f_{2i}(x_1, x_2, \dots, x_n) \text{ ET } \dots \text{ ET} \\ & \quad y_m = f_{mi}(x_1, x_2, \dots, x_n) \end{aligned} \quad (A.2)$$

Où

$x_1, x_2, \dots, x_n$  sont les variables linguistiques d'entrée.

$y_1, y_2, \dots, y_m$  sont les variables linguistiques de sortie.

$A_1^i, A_2^i, \dots, A_n^i$  sont les valeurs des variables linguistiques d'entrée dans les universs de discours  $X_1, X_2, \dots, X_n$ .

Dans l'équation (1),  $B_j^i$  est la valeur de la sortie  $j$  dans l'univers de discours  $Y_j$ ,  $j = 1, \dots, m$ .

Dans l'équation (2),  $y_j = f_{ji}(x_1, x_2, \dots, x_n)$  est une fonction prédéfinie des variables d'entrée. Dans la pratique  $f_{ji}(x_1, x_2, \dots, x_n)$  est très souvent choisie comme étant une fonction affine avec des paramètres linéaires.

Dans la base  $R$  de règles floues,  $R^i (i = 1, \dots, l)$  représente la règle  $i$ .

$$R = \bigcup_{i=1}^l R^i \quad (A.3)$$

### 3. Système d'inférence floue (SIF)

Le traitement effectué par un SIF se divise, comme indiqué sur la figure IV.3. en trois étapes :

- La fuzzification ;
- L'inférence ;
- La défuzzification.

#### 3.1. La fuzzification

La fuzzification permet de passer du domaine réel au domaine du flou. Elle consiste à déterminer le degré d'appartenance d'une valeur (mesurée par exemple) à un ensemble flou [55].

Une base de règles floues comprend des règles linguistiques faisant appel à des fonctions d'appartenance pour décrire les concepts utilisés. La fuzzification consiste à évaluer les fonctions d'appartenance utilisées dans les prédicats des règles.

Si la valeur  $x_0$  fournie par le capteur est supposée précise. La fuzzification singleton peut être utilisée. La fonction d'appartenance de la variable  $x$  à l'ensemble flou  $A_{x_0}$  est alors définie par:

$$\mu_{A_{x_0}}(x) = \begin{cases} 1 & \text{si } x = x_0 \\ 0 & \text{si } x \neq x_0 \end{cases} \quad (\text{A.4})$$

La figure A.2. donne une représentation de  $\mu_{A_{x_0}}(x)$ .

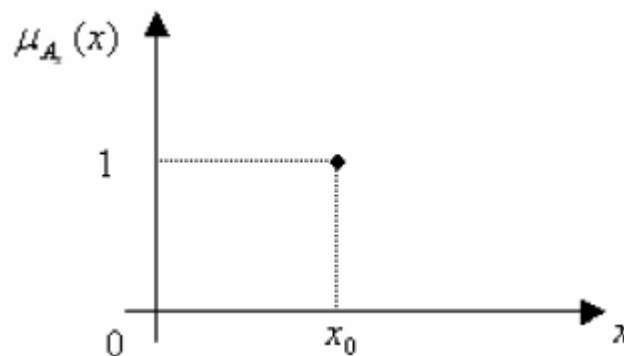


Figure A.2. Méthode de fuzzification singleton

### 3.2. L'inférence

L'inférence floue est le cœur du SIF, elle produit l'image de la partie floue issue de la fuzzification par une relation floue construite à partir de règles floues. C'est un mécanisme de décision.

Le moteur d'inférence évalue le prédicat et l'implication de chaque règle. Ensuite, un mécanisme d'agrégation permet de trouver l'ensemble flou global de sortie. Les opérateurs « ET » et « OU » interviennent dans le mécanisme d'inférence. Il existe différents méthodes d'inférence qui se distinguent entre elles par la manière dont elles évaluent ces opérateurs.

Prenons l'exemple de la méthode d'inférence « product inference » dans laquelle l'opérateur « ET » liant les propositions du prédicat correspond au produit des degrés

d'appartenance de ces propositions, l'implication correspond au produit de ce dernier résultat par la fonction d'appartenance de la proposition d'implication, et l'opérateur d'agrégation « OU » correspond au maximum entre les fonctions d'appartenance résultantes pour chaque règle. La figure A.3 donne une représentation de la méthode d'inférence « product inference » pour deux variables d'entrée, une variable de sortie et deux règles.

### 3.3. Défuzzification

Une fois l'inférence floue effectuée, l'ensemble flou obtenu en sortie doit être transformé en valeur numérique. Cette fonction est effectuée par le défuzzificateur.

Il existe plusieurs méthodes de défuzzification. Les plus connues et les plus utilisées sont « la méthode du centre de gravité » [56].

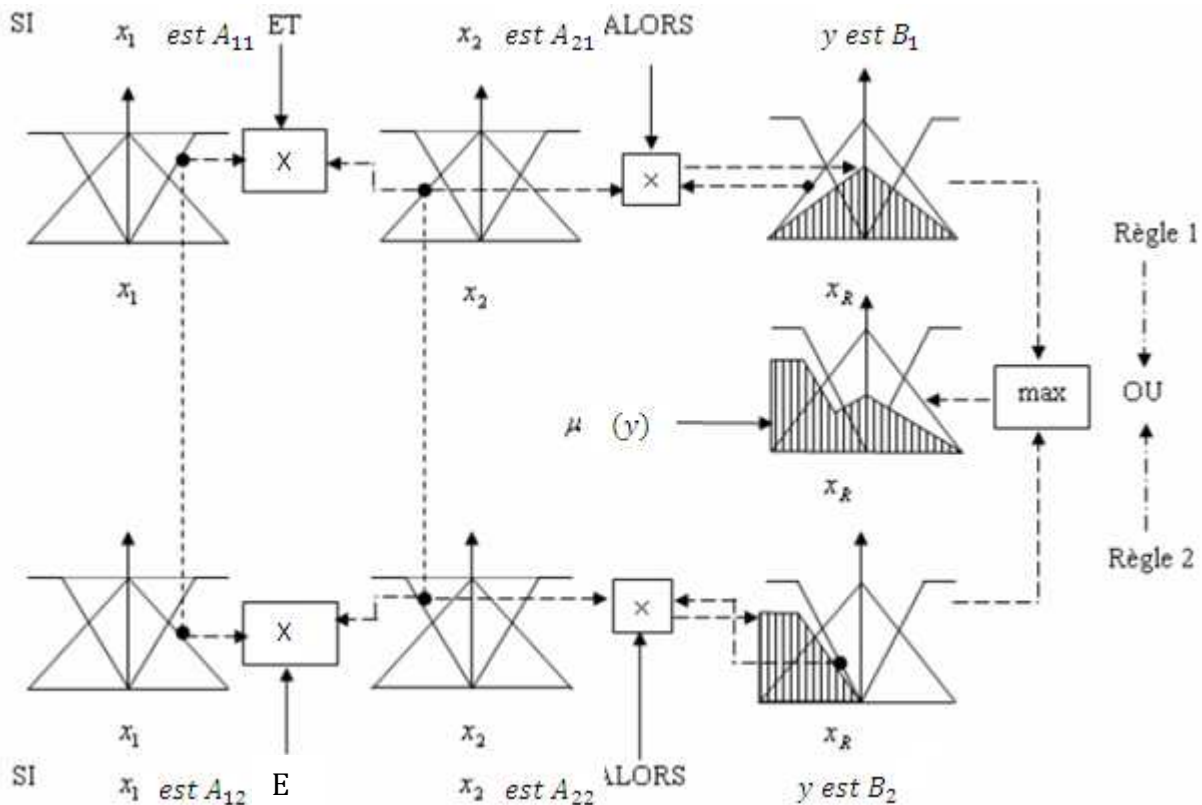


Figure A.3. Méthode d'inférence « product inference »

La méthode du centre de gravité produit la valeur  $x^*$  du centre de gravité de l'ensemble flou de sortie  $A_s$  appartenant à l'univers de discours  $X$ .

$$x^* = \frac{\sum_{i=1}^n x_i \mu_{A_s}(x_i)}{\sum_{i=1}^n \mu_{A_s}(x_i)} \tag{A.5}$$

Où

$n$  est le nombre d'éléments discrets dans l'univers de discours  $X$ .

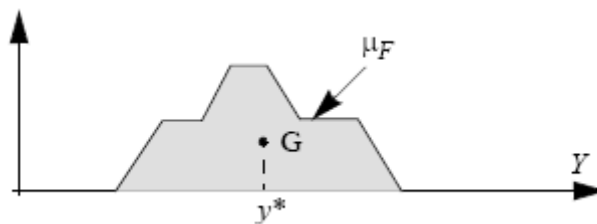
$x_i$  est la valeur de l'élément discret.

$\mu_{A_s}(x_i)$  représente la valeur de la fonction d'appartenance  $\mu_{A_s}$  au point  $x_i$ .

Si l'univers de discours est continu, alors la formule du centre de gravité s'écrit :

$$x^* = \frac{\int_{x \in X} x \mu_{A_s}(x) dx}{\int_{x \in X} \mu_{A_s}(x) dx} \quad (A.6)$$

La valeur  $x^*$  produite par la défuzzification est définie comme étant la coordonnée du centre de gravité de la surface située sous  $\mu_{A_s}$  (voir figure A.4).



**Figure A.4. défuzzification par le centre de gravité**

---



---

Références bibliographiques

---



---

- [1] S. Niku, *Introduction to robotics*: Wiley, 2010.
- [2] R. Kelly, V. Santibáñez, and A. Loría, *Control of robot manipulators in joint space*: Springer Verlag, 2005.
- [3] E. Dombre and W. Khalil, *Modeling, performance analysis and control of robot manipulators*: Wiley Online Library, 2007.
- [4] B. Siciliano, L. Sciavicco, and L. Villani, *Robotics: modelling, planning and control*: Springer Verlag, 2009.
- [5] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot manipulator control: theory and practice* vol. 15: CRC Press, 2004.
- [6] B. Siciliano and O. Khatib, *Springer handbook of robotics*: Springer-Verlag New York Inc, 2008.
- [7] F. Y. Wang and Y. Gao, *Advanced studies of flexible robotic manipulators: modeling, design, control and applications* vol. 4: World Scientific Pub Co Inc, 2003.
- [8] A. De Luca and P. Lucibello, "Inversion techniques for trajectory control of flexible robot arms," *Journal of robotic systems*, vol. 6, pp. 325-344, 1989.
- [9] A. De Luca and B. Siciliano, "Explicit dynamic modeling of a planar two-link flexible manipulator," 1990, pp. 528-530 vol. 2.
- [10] M. O. Tokhi and A. K. M. Azad, *Flexible robot manipulators: modelling, simulation and control* vol. 68: Iet, 2008.
- [11] M. Benosman and G. Le Vey, "Control of flexible manipulators: A survey," *Robotica*, vol. 22, pp. 533-545, 2004.
- [12] S. K. Dwivedy and P. Eberhard, "Dynamic analysis of flexible manipulators, a literature review," *Mechanism and Machine Theory*, vol. 41, pp. 749-777, 2006.
- [13] R. J. Theodore, "Dynamic modeling and control analysis of multilink flexible manipulators," 2012.
- [14] T. Fan and C. De Silva, "Dynamic modelling and model predictive control of flexible-link manipulators," *International Journal of Robotics and Automation* 2008, vol. 23, 2008.
- [15] A. De Luca and B. Siciliano, "Inversion-based nonlinear control of robot arms with flexible links," *Journal of guidance, control, and dynamics*, vol. 16, pp. 1169-1176, 1993.
- [16] E. F. Camacho and C. Bordons, *Model predictive control*: Springer Verlag, 2004.
- [17] J. Richalet, G. Lavielle, and J. Mallet, "La commande prédictive: mise en œuvre et applications industrielles," *Recherche*, vol. 67, p. 02, 2004.
- [18] P. Boucher and D. Dumur, "La commande prédictive: avancées et perspectives," *Hermes science*, 2006.
- [19] J. Richalet, "La commande prédictive," *REVUE DE L ELECTRICITE ET DE L ELECTRONIQUE*, pp. 37-38, 1998.
- [20] P. Boucher and D. Dumur, *La commande prédictive* vol. 8: Editions Technip, 1996.

- [21] T. Andreescu, D. Andrica, and I. Cucurezeanu, *An Introduction to Diophantine Equations: A Problem-based Approach*: Springer, 2010.
- [22] N. P. Smart, *The algorithmic resolution of Diophantine equations* vol. 41: Cambridge Univ Pr, 1998.
- [23] D. W. Clarke, C. Mohtadi, and P. Tuffs, "Generalized predictive control--Part I. The basic algorithm," *Automatica*, vol. 23, pp. 137-148, 1987.
- [24] D. W. Clarke and C. Mohtadi, "Properties of generalized predictive control," *Automatica*, vol. 25, pp. 859-875, 1989.
- [25] D. W. Clarke, "Application of generalized predictive control to industrial processes," *Control Systems Magazine, IEEE*, vol. 8, pp. 49-55, 1988.
- [26] R. B. Abdennour, *Identification et commande numérique des procédés industriels* vol. 11: Editions Technip, 2001.
- [27] M. E. Magana, "Feasibility study of robotic neural controllers," 1990.
- [28] K. Michels, F. Klawonn, R. Kruse, and A. Nürnberger, "Fuzzy Control: Fundamentals, Stability and Design of Fuzzy Controllers. 2006," ed: Springer, Berlin.
- [29] S. Sivanandam, S. Sumathi, and S. Deepa, *Introduction to fuzzy logic using MATLAB*: Springer Verlag, 2007.
- [30] J. Mamboundou and N. Langlois, "Robustness analysis of indirect adaptive model predictive control supervised by fuzzy logic," 2012, pp. 284-291.
- [31] S. Breban, F. Mollet, C. Saudemont, B. Robyns, and M. M. Radulescu, "Embedded electric power system with fuzzy-logic supervision for vehicular applications," 2012, pp. 1575-1579.
- [32] A. F. Amer, E. A. Sallam, and W. M. Elawady, "Adaptive fuzzy sliding mode control using supervisory fuzzy control for 3 DOF planar robot manipulators," *Applied Soft Computing*, 2011.
- [33] P. Sarhadi, K. Salahshoor, and A. Khaki-Sedigh, "Robustness analysis and tuning of generalized predictive control using frequency domain approaches," *Applied Mathematical Modelling*, 2012.
- [34] P. Borne, M. Ksouri, F. M'sahli, and R. B. Abdennour, *Identification et commande numérique des procédés industriels* Editions Technip, 2001.
- [35] Z. Song, J. Yi, D. Zhao, and X. Li, "A computed torque controller for uncertain robotic manipulator systems: Fuzzy approach," *Fuzzy sets and systems*, vol. 154, pp. 208-226, 2005.
- [36] S. C. Saini, Y. Sharma, M. Bhandari, and U. Satija, "Comparison of Pole Placement and LQR Applied to Single Link Flexible Manipulator," 2012, pp. 843-847.
- [37] B. Subudhi and A. Morris, "Soft computing methods applied to the control of a flexible robot manipulator," *Applied Soft Computing*, vol. 9, pp. 149-158, 2009.
- [38] G. Mamani, J. Becedas, and V. Feliu, "Sliding mode tracking control of a very lightweight single-link flexible robot robust to payload changes and motor friction," *Journal of Vibration and Control*, 2011.
- [39] V. Feliu, F. J. Castillo, F. Ramos, and J. A. Somolinos, "Robust tip trajectory tracking of a very lightweight single-link flexible arm in presence of large payload changes," *Mechatronics*, 2012.
- [40] T. Gorius, R. Seifried, and P. Eberhard, "Comparing exact inversion and singular perturbation approaches for a serial flexible manipulator," *PAMM*, vol. 11, pp. 45-46, 2011.

- [41] G. G. Rigatos, "Model-based and model-free control of flexible-link robots: A comparison between representative methods," *Applied Mathematical Modelling*, vol. 33, pp. 3906-3925, 2009.
- [42] P. V. Kokotović, H. K. Khalil, and J. O'reilly, *Singular perturbation methods in control: analysis and design* vol. 25: Society for Industrial Mathematics, 1999.
- [43] A. R. Fraser and R. W. Daniel, *Perturbation techniques for flexible manipulators* vol. 138: Springer, 1991.
- [44] B. Siciliano and W. J. Book, "A singular perturbation approach to control of lightweight flexible manipulators," *The International Journal of Robotics Research*, vol. 7, pp. 79-90, 1988.
- [45] T. Zhiling, "Modeling and control of flexible link robots," Master of Engineering, Department of Electrical and Computer Engineering, National University of Singapore, 2004.
- [46] Y. Aoustin and C. Chevallereau, "The singular perturbation control of a two-flexible-link robot," 1993, pp. 737-742 vol. 3.
- [47] !!! INVALID CITATION !!!
- [48] S. Olaru, "La commande des systèmes dynamiques sous contraintes Interaction optimisation-géométrie-commande," 2011.
- [49] S. Olaru and D. Dumur, "Lois de commande prédictives robustes: une approche par polyèdres paramétrés," 2006.
- [50] S. Olaru, "Contribution à l'étude de la commande prédictive sous contraintes par approche géométrique " Docteur en sciences, Université Paris XI Orsay 2005.
- [51] Z. Preitl, R. E. Precup, J. K. Tar, and M. Takács, "Use of Multi-parametric Quadratic Programming in Fuzzy Control Systems," *Acta Polytechnica Hungarica*, vol. 3, pp. 29-43, 2006.
- [52] K. M. a. F. M. Herceg Martin, "Stabilization of an inverted pendulum via fuzzy explicit predictive control  
" presented at the 8th International Scientific – Technical Conference – PROCESS CONTROL 2008, Koutny and Desnou, Czech Republic, 2008.
- [53] Guanrong Chen, Trung Tat Pham, Introduction to fuzzy sets, fuzzy logic and fuzzy control systems, CRC Press Taylor and Francis Group, 2000.
- [54] Timothy J Ross, Fuzzy logic with engineering applications, Second Edition, John Wiley and Sons Ltd, 2004.
- [55] F. Chevré, F. Guély, Cahier technique numéro 191: La logique floue, Groupe Schneider, 1998.
- [56] Yung C. Shin, Chengying Xu, Intelligent systems: Modeling, Optimization and Control , CRC Press Taylor and Francis Group, 2009.]

## Résumé

Dans le présent travail, la commande prédictive et la logique floue sont associées pour la commande d'un bras manipulateur. Le travail commence par la modélisation des robots manipulateurs. Nous nous intéresserons aux manipulateurs rigides et aux manipulateurs à axes flexibles.

La commande prédictive généralisée GPC est ensuite présentée sous forme polynomiale et appliquée au modèle linéarisé d'un manipulateur rigide. Une couche de supervision floue capable d'ajuster les paramètres de synthèse de la GPC est proposée et testée en simulation.

La linéarisation du modèle dynamique d'un robot manipulateur rigide repose sur l'hypothèse que les incertitudes structurées et non structurées sont nulles. Nous exploitons la propriété d'approximateurs universels des systèmes flous afin d'estimer les incertitudes auxquelles est sujet le robot et pouvoir ainsi les compenser. Le compensateur réalisé est testé en simulation en association avec la commande GPC sur un manipulateur rigide avec d'incertitudes structurées et non structurées et sur un manipulateur à axes flexibles. La méthode des perturbations singulières est ensuite appliquée au modèle du manipulateur flexible afin de le décomposer en sous-modèle rigide et sous modèle flexible. La commande GPC avec estimation floue des incertitudes est utilisée pour la commande du sous-modèle rigide. Le sous-modèle flexible sera commandé en utilisant la commande GPC. Des tests de simulations sont réalisés et les résultats comparés.

**Mots clé :** robot manipulateur rigide, robot manipulateur flexible, commande prédictive généralisée, supervision floue, approximateur flou, perturbations singulières.

## Abstract

In this work, predictive control and fuzzy logic are combined to control a robotic arm. The work begins with the modeling of robot manipulators. We focus on rigid manipulators and flexible link manipulators.

The generalized predictive control GPC is then presented in polynomial form and applied to the linearized model of a rigid manipulator. A fuzzy supervision able to adjust the parameters of the GPC is proposed and tested in simulation.

Linearization of the dynamic model of a rigid robot manipulator based on the assumption that the structured and unstructured uncertainties are zero. We exploit the property of universal approximators of fuzzy systems to estimate the uncertainties and compensate them. The compensator is tested in simulation in association with the GPC control on a rigid manipulator with structured and unstructured uncertainties and on a flexible link manipulator. Singular perturbation method is then applied to the model of flexible manipulator to decompose it into a rigid sub-model and a flexible sub-model. GPC control with fuzzy uncertainty estimation is used to control the rigid sub-model. The flexible sub-model will be controlled using GPC. Simulation tests are performed and results compared.

**Key words :** rigid manipulator, flexible manipulator, generalized predictive control, fuzzy supervision, fuzzy approximator, singular perturbation.

## ملخص

في هذا العمل، يتم الجمع بين التحكم التنبؤي والمنطق الغامض للتحكم في ذراع روبوتي ويبدأ العمل بتقديم نماذج أذرع الروبوت ونحن نركز على الأذرع الجامدة والمرنة. و بعد ذلك تم تقديم التحكم التنبؤي المعمم GPC على طريقة متعدد الحدود وتطبيقها على نموذج خطي لذراع روبوتي جامدة. ونقترح طبقة إشراف من المنطق الغامض قادرة على تكييف معاملات GPC وتم اختبارها في المحاكاة. حساب النموذج الديناميكي الخطي لذراع روبوتي جامد يتم على أساس افتراض أن الشكوك المنظمة وغير المنظمة هي صفر. نقوم باستغلال خاصية التقريب لدى المنطق الغامض لتقدير أوجه عدم اليقين من الروبوت وتعويضها. يتم اختبار طريقة التحكم هذه بالمحاكاة. على الأذرع الجامدة والمرنة. ثم يتم تطبيق طريقة الاضطراب المتفرد على نموذج ذراع روبوت مرنة لتتحلل الى نموذج جامد و نموذج مرن. يستخدم التحكم GPC مع تقدير عدم اليقين للسيطرة على الموديل الفرعي الجامد. وسيتم التحكم في نموذج المرن باستعمال GPC ويتم إجراء اختبارات المحاكاة و مقارنة النتائج.

**الكلمات الدالة :** ذراع روبوتي صلب، ذراع روبوتي مرن، التحكم التنبؤي المعمم، المنطق الغامض، إشراف باستعمال المنطق الغامض، تقريب باستعمال المنطق الغامض، اضطراب متفرد.