



المدرسة الوطنية المتعددة الفنون
Ecole Nationale Polytechnique

République Algérienne Démocratique et Populaire

Ecole Nationale Polytechnique

Département de Génie Electrique

7/01

PROJET DE FIN D'ETUDE

OPTION AUTOMATIQUE

المدرسة الوطنية المتعددة الفنون
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

COMMANDE ROBUSTE DECENTRALISEE A BASE DES SYSTEMES FLOUS APPLICATION EN ROBOTIQUE

Proposé et dirigé par :

M. M. TADJINE
M. D. BOUKHETALA

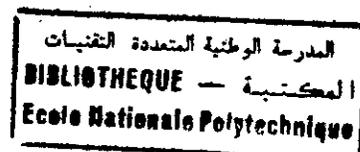
Etudié par :

M. Mehdi GATI
M. Mourad MAZOUZ

République Algérienne Démocratique et Populaire

Ecole Nationale Polytechnique

Département de Génie Electrique



PROJET DE FIN D'ETUDE

OPTION AUTOMATIQUE

COMMANDE ROBUSTE DECENTRALISEE A BASE DES SYSTEMES FLOUS APPLICATION EN ROBOTIQUE

Proposé et dirigé par :

**M. M. TADJINE
M. D. BOUKHETALA**

Etudié par :

**M. Mehdi GATI
M. Mourad MAZOUZ**

REMERCIEMENTS

Les travaux présentés dans ce mémoire ont été menés au Laboratoire de Commande des Processus à l'Ecole Nationale Polytechnique d'Alger.

Nous remercions nos promoteurs, messieurs : M. Tadjine et D. Boukhetela, pour leurs conseils et efforts durant toute l'année.

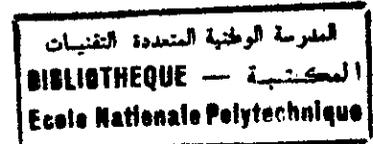
Nous remercions aussi Monsieur H. Chekireb pour tous ses précieux conseils et indications qui nous ont été d'une très grande utilité.

Nos remerciements vont aujourd'hui à tous ceux qui ont participé à notre formation du primaire à l'universitaire.

Enfin, nous remercions tout le personnel de la bibliothèque surtout ceux du service périodique.

Mourad MAZOUZ & Mehdi GATI

DEDICACES



Je dédie ce travail avant tout, à ma mère; qui m'a tout donné et à qui je dois tout, et j'espère que j'aurais un jour l'occasion de la remercier d'avantage.

Je dédie aussi ce travail, à mon père; pour sa disponibilité, pour les moyens qu'il m'a fournis, et que sans sa contribution, ce travail n'aurait pas cette finition.

Je n'oublie pas non plus, mon oncle et ma tante pour toute leur aide et leur disponibilité durant ces cinq années

A tous les membres de ma famille; frères, sœurs, cousins et cousines.

Enfin je dédie ce travail à ma nièce maïssa.

Mehdi GATI

Je dédie ce travail avant tout, à ma mère; qui m'a tout donné et à qui je dois tout, et j'espère que j'aurais un jour la possibilité de la remercier d'avantage.

Je dédie aussi ce travail, à mon père; pour sa disponibilité, pour les moyens qu'il m'a fournis, et que sans sa contribution, ce travail n'aurait pas cette finition.

A mes deux frères Abdennour et Nabil pour toutes leurs aides et leur disponibilité, ainsi que mes sœurs, mes beaux-frères et mes belles-sœurs

Je n'oublie pas non plus, mon neveu Samy et mes nièces Dounia et Manel

A la très chère Ines

Et enfin à la mémoire de Hayet

Mourad MAZOUZ

SOMMAIRE



	INTRODUCTION GENERALE	1
	NOTATIONS	3
CHAPITRE I	SYSTEMES FLOUS	4
I.1	Introduction	4
I.2	Ensemble flou et logique floue	4
I.2.1	Ensemble net.	4
I.2.2	Ensemble flou.	4
I.3	Les opérations en logique floue	5
I.4	Variables linguistiques	6
I.5	Fonction d'appartenance.	7
I.6	Implication floue.	8
I.7	Raisonnement flou.	9
I.8	Structure d'un système flou	9
I.8.1	Bases des règles floues.	10
I.8.2	Moteur d'inférences flous	10
I.8.3	Fuzzification	12
I.8.4	Defuzzification	12
I.8.4.1	Méthode du maximum	12
I.8.4.2	Méthode de la moyenne des maxima	12
I.8.4.3	Méthode du centre de gravité	12
I.8.4.4	Méthode des hauteurs pondérées	13
I.8.4.5	Méthode des hauteurs pondérées modifiée.	13
I.9	Analyse et description du système flou de Sugeno	13
I.10	Adaptation des systèmes flous.	15
I.11	Application de la logique floue à l'identification.	16
I.11.1	Introduction.	16
I.11.2	Résultats et commentaires	20
I.12	Conclusion.	26
CHAPITRE II	COMMANDE H_∞	27
II.1	Norme infinie	27
II.1.1	Valeurs singulières et norme H_∞ d'une matrice	27
II.1.2	Robustesse	28
II.1.3	Théorème du petit gain.	28
II.2	Incertitudes de modélisation	29
II.3	Robustesse en stabilité	30
II.4	Objectifs de la synthèse.	31
II.4.1	Fonction sensibilité.	31
II.4.2	Performances nominales	32
II.5	Problème standard	34
II.6	Approche par factorisation première.	37
II.6.1	Définition.	37

II.6.2	Position du problème	38
II.6.3	La solution	40
II.7	Application au robot PUMA 560	41
II.7.1	Synthèse des systèmes flous	42
II.7.2	Synthèse des correcteurs	43
II.7.3	Simulation non-linéaire	47
II.8	conclusion	48
CHAPITRE III	COMMANDE PAR RETOUR D'ETAT	50
III.1	Introduction	50
III.2	Position du problème	50
III.3	Etude de la stabilité	51
III.4	H ₂ synthèse dans le cas d'un retour d'état	53
III.5	H ₂ synthèse dans le cas d'un retour de sortie	54
III.6	Application au robot PUMA 560	56
III.7	Placement de pôles par retour d'état	59
III.8	Conclusion	62
CHAPITRE IV	TESTS DE ROBUSTESSE	63
IV.1	Introduction	63
IV.2	Test de Lheavy	63
IV.3	Effet de la variation paramétrique due à la masse	67
IV.4	Effet des variations paramétriques	71
IV.5	Test de décentralisation	75
IV.6	Influence de l'actionneur	79
IV.7	Test des perturbations externes	83
IV.8	Test des bruits de mesures	87
IV.9	Changement de trajectoire	91
	CONCLUSION GENERALE	97
	ANNEXE I	98
	ANNEXE II	106
	BIBLIOGRAPHIE	108

INTRODUCTION GENERALE



INTRODUCTION :

La synthèse de commande pour un système nécessite souvent la connaissance d'un modèle mathématique décrivant son comportement. Cependant, il est certain que ce modèle n'est pas une représentation parfaite de la dynamique du système réel car les phénomènes physiques ne sont pas, généralement, tous décrits par des équations. De plus, nous sommes souvent confrontés à des systèmes si complexes qu'on ne peut déduire toutes les équations physiques les régissant. Pour y remédier, plusieurs techniques d'identifications ont été proposées. Depuis quelques années on parle des systèmes flous de Sugeno [17], [18], ces systèmes basés sur la logique floue utilisent des algorithmes d'apprentissage pour adapter leurs paramètres afin d'approcher une sortie désirée. Cependant, le modèle obtenu n'est qu'une approximation de la réalité.

Dans une chaîne de commande, nous disposons toujours de capteurs pour mesurer les signaux. Comme dans la majorité des cas ces derniers sont à base d'électronique, ils produisent toujours des bruits de mesure (généralement haute fréquence) nuisible au système de commande.

Ainsi, le modèle sur lequel on se base pour la synthèse de la commande ne reflète pas toujours la réalité. Alors, la question qui nous vient à l'esprit est :

Est ce que la commande qui a été synthétisée sur le modèle mathématique, qui n'est qu'une approximation de la réalité, sera aussi efficace dans la réalité ?

Pour parer à ce problème, des techniques de synthèse de commandes, dite robustes, ont été développées [4], [5], [7]. Dans ces techniques la commande déduite à partir d'un système nominal doit maintenir un certains niveau de performance face à une classe d'incertitude de modèle.

Sachant que le modèle mathématique dont nous disposons n'est qu'une approximation de la réalité, ses paramètres sont donc incertains. Pour chaque valeur possible de ces paramètres nous obtenons un modèle. Nous avons donc, non pas un modèle mais une classe de modèles.

Parmi tous ces systèmes, il en existe un qui soit le plus défavorable dans le sens du degré de stabilité ou des performances. Alors si on arrive à stabiliser le système le plus défavorable on est certains que tous les autres seront stabilisés et par conséquent le système réel.

La commande H_∞ consiste à stabiliser un système dit nominal ainsi qu'une classe de systèmes incertains mais dont l'incertitude reste bornée [27].

Aussi le modèle dont nous disposons est souvent entaché d'erreurs. De plus le système est souvent sujet à des perturbations externes tels que les bruits de mesure ou les variations des conditions de fonctionnement, il faut donc minimiser l'effet les perturbations sur la sortie. Pour cela, on minimise l'énergie de transmission entre la perturbation et la sortie. Cette énergie n'est autre que la norme H_2 de la fonction de transfert entre la perturbation et la sortie [29].

ORGANISATION DU TRAVAIL :

Notre objectif est de synthétiser une commande robuste qui tient compte de l'erreur de modélisation et de l'effet des perturbations externes pouvant compromettre le bon fonctionnement d'un bras manipulateur du type *PUMA 560*.

Pour une meilleure synthèse de la commande nous approximations le système interconnecté de départ par trois sous systèmes linéaires et totalement indépendants. Il s'agit donc de faire apparaître l'effet des interconnexions dans les paramètres des systèmes estimés. Ces derniers sont déduits à l'aide des systèmes flous de Sugeno. C'est l'objectif du chapitre I.

Dans le Chapitre II nous synthétisons une commande H_∞ décentralisée basée sur les modèles flous du chapitre I. Comme le manipulateur est un système non linéaire, on utilise trois sous systèmes flous pour chaque articulation.

Dans le chapitre III nous déduisons un retour d'état optimal au sens H_2 que nous comparons à un retour d'état permettant d'effectuer un placement des pôles (placement de pôles par retour d'état).

Dans le chapitre IV nous soumettons les contrôleurs synthétisés dans les chapitres précédents à différents tests de robustesse. D'abord on applique à notre modèle une variation paramétrique, Puis un test de décentralisation est effectué en simulant des ruptures de commandes. Ensuite on insère un actionneur dans la chaîne de commande ce qui permet d'adoucir les commandes. Puis des bruits hautes fréquences sont superposés aux états mesurés afin de simuler l'effet des capteurs. Finalement, nous appliquons au manipulateur une consigne de forme exponentielle. Des conclusions sont tirées à la fin de chaque test.

NOTATIONS

I : Matrice identité.

\mathbb{R} : Corps des nombres réels.

\mathcal{RH}_∞ : Ensemble des fonctions de transfert rationnelles et stables.

s : Variable de Laplace.

ω : Pulsation(rad/s).

A^T : Matrice transposée de A .

A^* : Matrice complexe conjuguée de A .

$\text{Trace}(A)$: somme des élément diagonaux de la matrice A .

$[A, B, C, D]$ ou $\left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right)$: représentation d'état du système $G(s)$.

$$G(s) = C(sI - A)^{-1}B + D.$$

$\sigma_i(A)$: $i^{\text{ème}}$ valeur singulière de la matrice A .

$\bar{\sigma}$: la plus grande valeur singulière de A .

$\underline{\sigma}$: la plus petite valeur singulière de A .

CHAPITRE I

LES SYSTEMES FLOUS

Dans ce chapitre nous donnons, en premier lieu, une idée générale sur les notions de base requises pour la conception des systèmes flous que ce soit pour la commande ou pour l'identification et l'approximation des systèmes. Dans la deuxième partie de ce chapitre, nous appliquons cette méthode pour l'approximation du modèle centralisé et non linéaire du robot puma560 par trois sous systèmes linéaires et parfaitement dissociés.

I.1 INTRODUCTION

C'est à Lotfi Zadeh, spécialiste renommé en automatique et en théorie des systèmes, à l'université de Berkeley, que revient le mérite d'avoir établi les bases théoriques de la logique floue [34]. L'intérêt de la logique floue réside dans sa capacité à traiter l'imprécis, l'incertain et le vague. Elle est issue de la capacité de l'homme à décider et agir de façon pertinente malgré le flou des connaissances disponibles. En effet, la logique floue a été introduite pour approcher le raisonnement humain à l'aide d'une représentation adéquate des connaissances.

L'imitation des aspects qualitatifs et approximatifs du raisonnement humain a donné naissance à un type de système de commande qui trouve de puissantes applications, surtout dans les situations où le système à commander est mal défini mathématiquement et /ou en présence d'une expertise qui puisse fournir des règles de commande en termes d'implications vagues et floues.

Ces dernières années, la logique floue a été appliquée à divers domaines. Cependant, la commande floue est le domaine qui suscite le plus de curiosité et s'avère être le champ d'application le plus actif à travers le monde [8], [13]. Parmi les nombreuses applications de cette discipline, on cite : l'ajustement des images T.V., la reconnaissance de manuscrit, l'analyse des stocks, la commande des hélicoptères et l'automatisation des voitures [11],[13].

Dans ce chapitre, nous donnons un bref rappel sur la théorie des ensembles flous et un aperçu général sur la logique floue. Ensuite, nous présentons la structure des systèmes flous. Une analyse des systèmes flous de Sugeno est donnée, ainsi que leurs techniques d'apprentissage. Dans la dernière partie, nous abordons l'utilisation de la logique floue dans le domaine de l'identification.

I.2 ENSEMBLE FLOU ET LOGIQUE FLOUE

La logique floue, introduite en 1965 peut être vue comme une extension de la logique booléenne. De plus, elle permet de traiter des variables linguistiques dont les valeurs sont des mots ou expressions du langage naturel. Cette partie fournit une introduction générale et un résumé des concepts de base essentiels pour l'étude de la logique floue.

I.2.1 Ensemble net

Un ensemble A sur un référentiel U dont les limites sont définies précisément s'appelle ensemble net ou ordinaire figure(1.1-a). La description d'un ensemble net consiste en un énoncé des éléments qui appartiennent à cet ensemble, et qui le constituent. Une alternative pour définir est de donner la condition pour laquelle un élément $u \in A$. Alors, l'ensemble A peut être défini tel que :

$$A = \{u / u \text{ vérifie la condition } \}$$

De plus, on peut introduire une fonction cartésienne pour A , notée $\mu_A(u)$ telle que :

$$\mu_A(u) = \begin{cases} 1 & u \in A \\ 0 & u \notin A \end{cases} \quad (I.1)$$

I.2.2 Ensemble flou

La notion d'ensemble flou provient du constat que «très souvent, les classes d'objets rencontrées dans le monde physique ne possèdent pas de critères d'appartenance bien définis »[13].

Mathématiquement, un ensemble flou A est défini sur un référentiel U , par une fonction d'appartenance, notée μ_A , qui appliquée à un élément u de U , retourne un degré d'appartenance $\mu_A(u)$ de u à A . Un ensemble flou peut être représenté comme un ensemble de paires ordonnées :

$$A = \{ (u, \mu_A(u)) / u \in U \}$$

Si U est discret, A est représenté par :

$$A = \sum_{u_i \in U} \mu(u_i) / u_i \quad (I.2)$$

Si U est continu, A est représenté par :

$$A = \int \mu(u) / u \quad (I.3)$$

Les ensembles flous ont le grand avantage de constituer une représentation mathématique des termes linguistiques largement utilisés dans l'expression des connaissances expertes, qualitatives et qui sont manipulées par la logique floue.

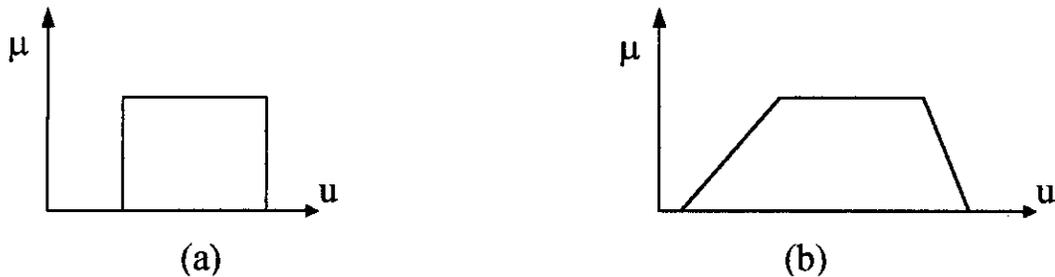


Figure (1.1) exemple de fonctions d'appartenance : (a) logique classique ; (b) logique floue.

I.3 LES OPERATIONS EN LOGIQUE FLOUE

Disposant d'une représentation formelle des classes floues sous forme de fonctions d'appartenance, il est naturel de chercher à faire avec les ensembles flous ce que l'on fait avec les ensembles classiques. Zadeh a ainsi généralisé les opérateurs des ensembles flous qui sont définis à partir de leurs fonctions d'appartenance.

Soit A et B deux ensembles flous définis dans le référentiel U avec les fonctions d'appartenance μ_A et μ_B respectivement.

Définition 1 : Union (Disjonction) :

La fonction d'appartenance $\mu_{A \cup B}$ ou A OR B est définie pour tout $u \in U$ par :

$$\mu_{A \cup B}(u) = \mu_A(u) + \mu_B(u) \quad (I.4)$$

Le symbole "+" représente la co-norme triangulaire [11]. Le Max est la co-norme triangulaire la plus utilisée dans le domaine de la commande.

$$\mu_A(u) + \mu_B(u) = \text{Max} (\mu_A(u), \mu_B(u)) \quad (I.5)$$

Définition 2 : Intersection (Conjonction) :

La fonction d'appartenance $\mu_{A \cap B}$ de l'intersection $A \cap B$ ou A AND B est définie pour tout $u \in U$ par :

$$\mu_{A \cap B}(u) = \mu_A(u) * \mu_B(u) \quad (I.6)$$

Le symbole "*" représente la norme triangulaire [11]. Deux normes triangulaires sont le plus souvent utilisées pour évaluer l'intersection floue.

▪ Min :
$$\mu_A(u) * \mu_B(u) = \text{Min}(\mu_A(u), \mu_B(u)) \quad (I.7)$$

▪ Produit algébrique :
$$\mu_A(u) * \mu_B(u) = \mu_A(u) \times \mu_B(u) \quad (I.8)$$

Définition 3 : Complémentation :

La fonction d'appartenance $\mu_{\bar{A}}$ du complément de l'ensemble A ; noté \bar{A} ou NOT(A), est définie pour tous $u \in U$ par :

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u) \quad (I.9)$$

Définition 4 : Produit cartésien :

Soit A_1, \dots, A_n des ensembles flous respectivement définis sur U_1, \dots, U_n , le produit cartésien de A_1, \dots, A_n est un ensemble flou défini sur U_1, \dots, U_n , de fonction d'appartenance :

$$\mu_{A_1 \times A_2 \times \dots \times A_n}(u_1, \dots, u_n) = \mu_{A_1} * \dots * \mu_{A_n} \quad (I.10)$$

Définition 5 : Relation floue :

Une relation floue représente le degré de présence ou, d'absence d'une association entre les éléments de deux ou de plusieurs ensembles flous. Une relation floue d'ordre n est un ensemble flou défini sur $U_1 \times \dots \times U_n$ par l'expression suivante :

$$R_{U_1 \times \dots \times U_n} = \{((U_1, \dots, U_n), \mu_R(u_1, \dots, u_2)) / (u_1, \dots, u_n) \in U_1 \times \dots \times U_n\} \quad (I.11)$$

Définition 6 : Composition des règles floues :

Soit R et S deux relations floues définies respectivement dans $U \times V$ et $V \times W$. La composition de R et S est un ensemble flou, symbolisé par $R \circ S$, de fonction d'appartenance :

$$\mu_{R \circ S}(u, w) = \left\{ (u, w), \sup_{v \in V} (\mu_R(u, v) * \mu_S(v, w)) \right\} \quad (I.12)$$

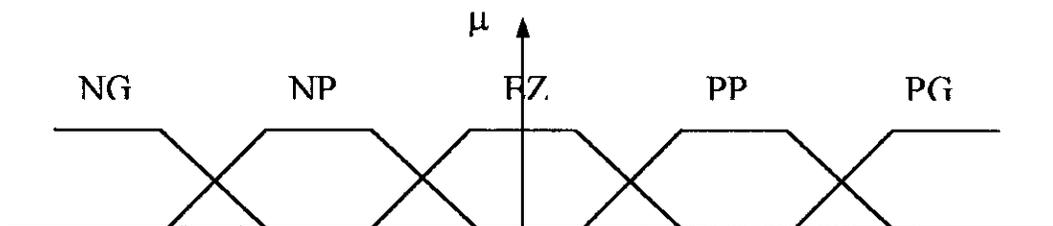
I.4 VARIABLE LINGUISTIQUE :

C'est une variable dont les valeurs ne sont pas des nombres, mais des mots ou phrases exprimés en langage naturel [11], [13]. La raison pour laquelle on utilise cette représentation, est que le caractère linguistique est moins précis que le caractère numérique.

Une variable linguistique est généralement représentée par un triplet $(x, T(x), U)$, dans lequel x est le nom de la variable linguistique (vitesse, position, erreur,...), $T(x)$ est l'ensemble des valeurs linguistiques ; qui sont utilisées pour caractériser x . Par exemple, si l'erreur est considérée comme variable linguistique, définie sur le référentiel $U=[-10, + 10]$, ses valeurs linguistiques peuvent être définies comme suit :

$T(\text{erreur}) = \{\text{Négative Grande (NG), Négative Petite (NP), Environ Zéro (EZ), Positive Petite (PP), Positive Grande (PG)}\}$.

Ces symboles linguistiques peuvent être considérés comme des ensembles flous dont les fonctions d'appartenance sont représentées sur la figure (I.2).



Figure(I.2) Représentation de la variable linguistique *erreur*

I.5 FONCTION D'APPARTENANCE

Dans ce contexte, on associe à chaque valeur de la variable linguistique une fonction d'appartenance désignée par $\mu_A(x)$, où x est la variable linguistique, tandis que A indique l'ensemble concerné. Une valeur précise de $\mu_A(x)$, est désignée par le degré ou le facteur d'appartenance. Il est à noter que l'ensemble des éléments x pour lesquels $\mu_A(x) > 0$, est appelé support de A . De plus afin de permettre un traitement numérique des variables linguistiques dans la prise de décision floue sur ordinateur, une définition des variables linguistiques à l'aide de fonctions d'appartenance s'impose. Souvent, on utilise pour les fonctions d'appartenance les fonctions suivantes (figure 1.3) [31], [8], [11] :

- *Fonction triangulaire* : Elle est définie par trois paramètres $\{a, b, c\}$, qui déterminent les coordonnées des trois sommets.

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (I.13)$$

- *Fonction trapézoïdale* : Elle est définie par quatre paramètres $\{a, b, c, d\}$

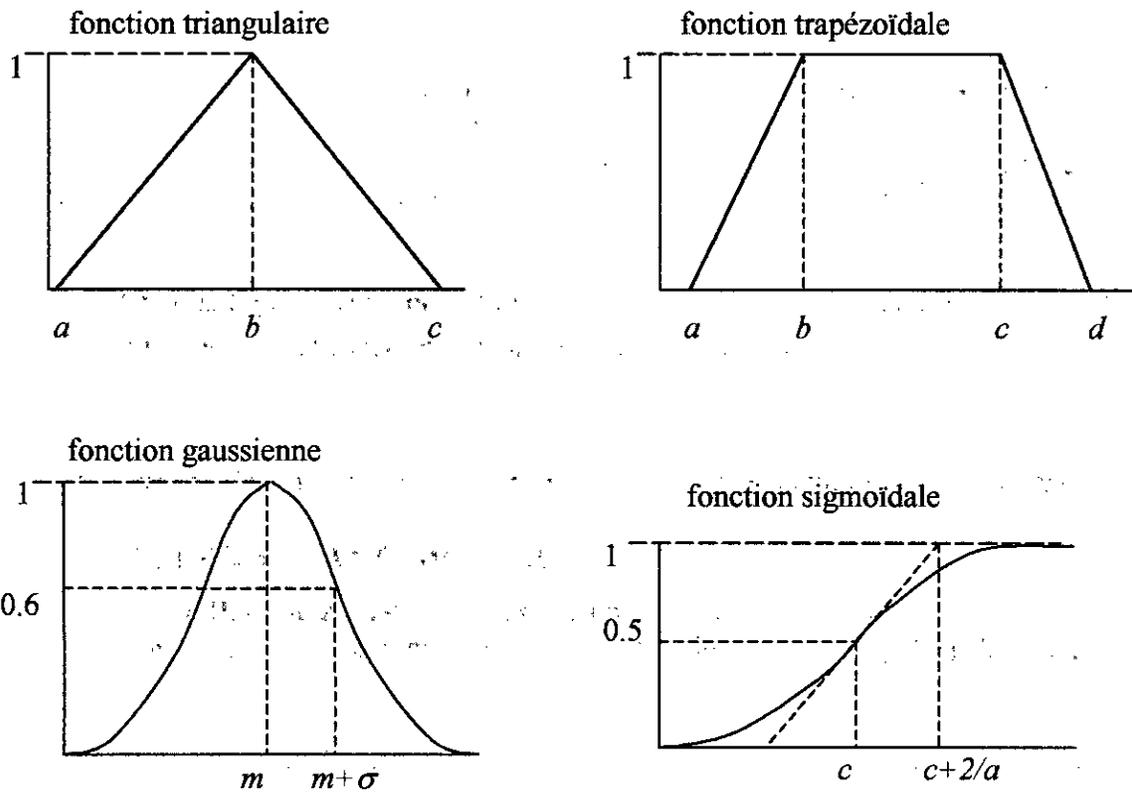
$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{d-x}{d-c}, 1\right), 0\right) \quad (1.14)$$

- **Fonction gaussienne** : Elle est définie par deux paramètres $\{\sigma, m\}$

$$\mu(x) = \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \tag{I.15}$$

- **Fonction sigmoïdale** : Une fonction sigmoïdale est définie par deux paramètres $\{a, c\}$

$$\mu(x) = \frac{1}{1 + \exp(-a(x-c))} \tag{I.16}$$



Figure(1.3) formes usuelles des fonctions d'appartenance

I.6 IMPLICATION FLOUE

L'implication floue est un opérateur qui permet d'évaluer le degré de vérité d'une règle R de la forme "Si x est A alors y est B " à partir des valeurs de la prémisse d'une part, et de celle de la conclusion d'autre part.

$$\mu_R(x, y) = \text{imp}(\mu_A(x), \mu_B(y)) \tag{I.17}$$

Les opérateurs les plus utilisés sont les opérateurs de Mamdani et de Larsen.

- **Implication de Mamdani**

$$\mu_R(x, y) = \min(\mu_A(x), \mu_B(y)) \tag{I.18}$$

- La fuzzification transforme la grandeur physique d'entrée en une grandeur floue.
- La base des règles floues ou base des connaissances, contient les règles floues décrivant le comportement du système ; elle est le cœur du système entier dans le sens où tous les autres composants sont utilisés pour interpréter et combiner ces règles pour former le système final.
- Le moteur d'inférence floue transforme, à l'aide des techniques de raisonnement flou la partie floue issue de la fuzzification en une nouvelle partie floue.
- La défuzzification transforme la grandeur floue issue de l'inférence en une grandeur physique.

1.8.1 Base des règles floues

Une base des règles floues R est une collection de règles floues de la forme SI-ALORS $R=[R_1, \dots, R_M]$ telle que :

$$R_l : \text{SI } x_1 \text{ est } A_{l1} \text{ et } \dots \text{ et } x_n \text{ est } A_{ln} \quad \text{ALORS } y \text{ est } B_l$$

Ou sous la forme de Sugeno :

$$R_l : \text{SI } x_1 \text{ est } A_{l1} \text{ et } \dots \text{ et } x_n \text{ est } A_{ln} \quad \text{ALORS } y = f_l(x)$$

Avec $f_l(x)$ un polynôme

1.8.2 Moteur d'inférence floue

Le moteur d'inférence flou utilise la base des règles floues pour effectuer une transformation à partir des ensembles flous dans l'espace d'entrée vers les ensembles flous dans l'espace de sortie en se basant sur les opérations de la logique floue. L'antécédent de la règle R_l définit un produit cartésien de A_{l1}, \dots, A_{ln} , et la règle elle-même R_l est vue comme une implication. Soit A_x un ensemble flou dans U , alors chaque règle R_l , détermine un ensemble flou $B_l = A_x \circ R_l$ dans V . la fonction d'appartenance de B_l est donnée par la règle compositionnelle (I.20), [13].

Les ensembles des M règles constituant la base sont liées par l'opérateur de disjonction "ou". Ainsi, l'ensemble flou final $B' = A_x \circ R$, est donné par la relation

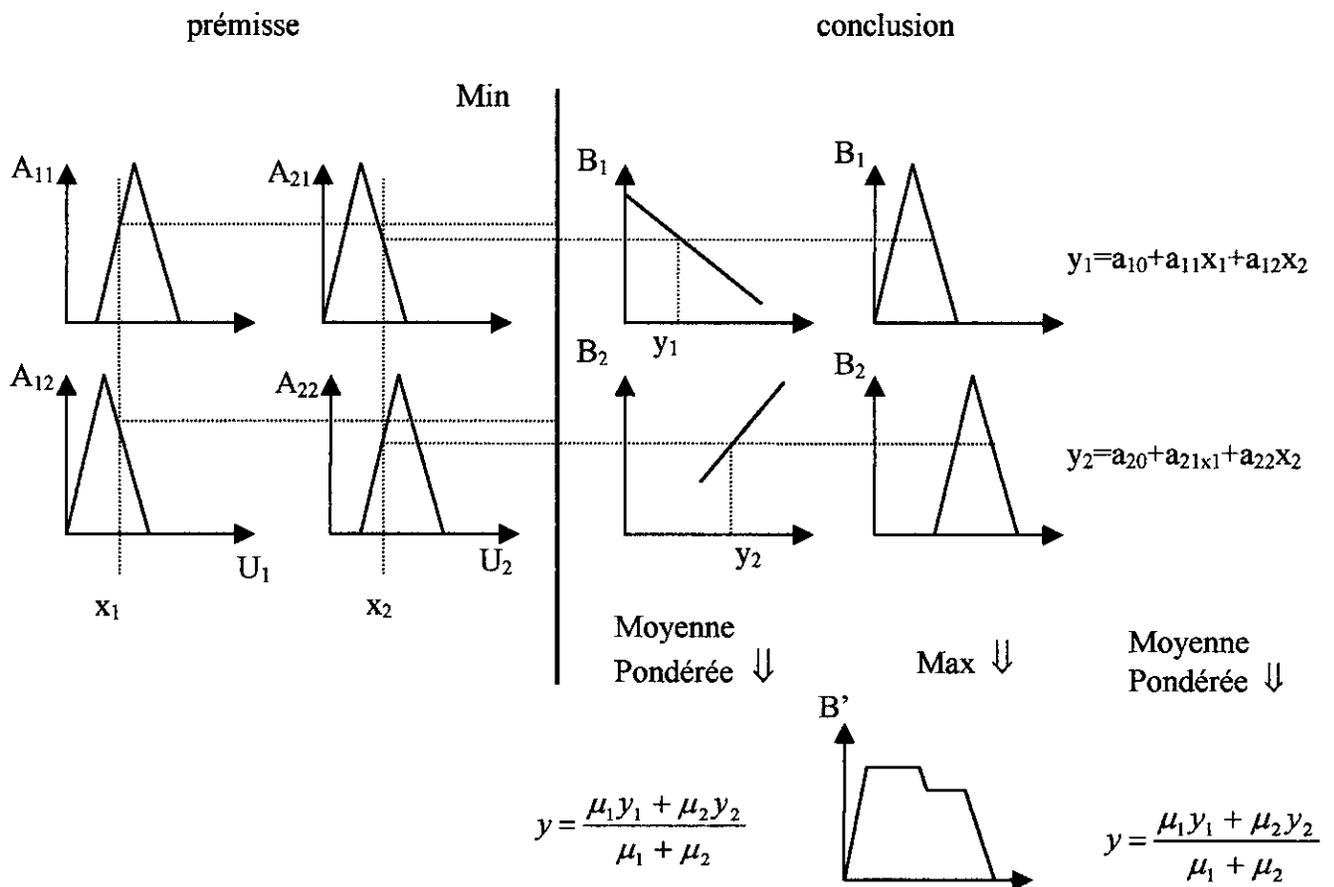
$$\begin{cases} B' = B'_1 + \dots + B'_M \\ \mu_{B'}(y) = \mu_{A_x \circ R_1}(y) + \dots + \mu_{A_x \circ R_M}(y) \end{cases} \quad (I.21)$$

Dans le jeu de règles du système flou interviennent les opérateurs flous ET (AND) et OU (OR). L'opérateur AND s'applique aux variables à l'intérieur d'une règle, tandis que l'opérateur OR lie les différentes règles. Plusieurs types de raisonnement flou ont été proposés dans la littérature suivant la réalisation des opérateurs flous ET et OU et le type de règles floues utilisées. Les moteurs d'inférence flous les plus utilisés peuvent être classés en trois catégories [11], [12], [18], [21] :

- Type 1 (*méthode de Tsukamoto*) : dans ce cas, des fonctions monotones sont associées aux variables de sortie. La sortie totale est une moyenne pondérée des degrés d'activations de chaque règle et des valeurs des fonctions des variables de sortie.

- Type 2 (*méthode de Mamdani*) : dans ce type les conclusions sont symboliques ou linguistiques et le calcul de la sortie nécessite une méthode de défuzzification.
- Type 3 (*méthode de Sugeno*) : les règles floues de type Sugeno sont déterministes. En effet la conclusion de chaque règle a une forme polynomiale ou plus généralement une fonction des variables d'entrées. La sortie finale est égale à la moyenne pondérée de chaque règle floue.

La figure (I.5) illustre les trois types de raisonnement flou pour un système flou à deux entrées et une base de connaissance de deux règles floues. On constate que les différences viennent de la spécification de la partie conclusion d'une part, et de la méthode de défuzzification d'autre part.



Figure(I.5) types de règles floues et de moteurs d'inférence flous

1.8.3 Fuzzification

La fuzzification consiste à relier le point numérique $x_0 = (x_{10}, \dots, x_{n0})$ de U à l'ensemble flou $A_x = [A_{x1}, \dots, A_{xn}]$ dans $U = U_1 \times \dots \times U_n$, où A_{xi} est un ensemble flou dans U_i . Il existe deux méthodes de fuzzification suivant la définition de A_x [11], [13].

- A_x est un singleton flou défini par :

$$\mu_{A_x}(x) = \begin{cases} 1 & x = x_0 \\ 0 & x \neq x_0 \end{cases} \quad (I.22)$$

Dans ce cas on considère que la valeur de x est précise et certaine.

- A_x est un ensemble flou de fonction d'appartenance $\mu_{A_x}(x_0) = 1$ et $\mu_{A_x}(x)$ décroît lorsque x s'éloigne de x_0 .

Dans ce cas on prend en compte le comportement de la variable autour de la valeur x_0 . Par exemple l'erreur de mesure est modélisée par une fonction d'appartenance triangulaire.

1.8.4 Défuzzification

Le rôle de la défuzzification est de transformer la partie floue issue de l'inférence en une grandeur numérique. Malheureusement, il n'y a pas de procédure systématique pour choisir la stratégie de défuzzification. Nous présentons les méthodes les plus répandues [31], [8], [11], [13]

1.8.4.1 Méthode du maximum

La méthode du maximum examine l'ensemble flou B issu de l'inférence et choisit comme sortie la valeur y pour laquelle la fonction $\mu_{B'}(y)$ est maximale. Cependant ce défuzzificateur présente un certain inconvénient lorsqu'il existe plusieurs valeurs pour lesquelles $\mu_{B'}(y)$ est un maximum.

1.8.4.2 Méthode de la moyenne des maximas

Ce défuzzificateur examine l'ensemble flou B issue de l'inférence et détermine en premier temps les valeurs y_i pour lesquelles la fonction $\mu_{B'}(y)$ est un maximum. Ensuite, il calcule la moyenne de ces valeurs comme sortie de défuzzification.

1.8.4.3 Méthode du centre de gravité

Dans ce cas, on calcule le centre de gravité \bar{y} de l'ensemble résultant B' et considère cette valeur comme résultat de la défuzzification.

$$\bar{y} = \frac{\int_S y \mu_{B'}(y) dy}{\int_S \mu_{B'}(y) dy} \quad (I.23)$$

Où s représente le support de B' . Souvent, on utilise la version discrète de l'intégrale ce qui donne :

$$\bar{y} = \frac{\sum_i y_i \mu_{B'}(y_i)}{\sum_i \mu_{B'}(y_i)} \quad (I.24)$$

Il est à noter que le centre de gravité est généralement difficile à calculer. De ce fait, cette méthode est la plus coûteuse en temps de calcul.

1.8.4.4 Méthode des hauteurs pondérées

La valeur réelle de la sortie est donnée par la relation suivante :

$$\bar{y} = \frac{\sum_i \bar{y}_i \mu_i}{\sum_i \mu_i} \quad (I.25)$$

Où

μ_i : représente le degré d'activation de la règle R_i

\bar{y}_i : le centre de gravité de l'ensemble flou de la variable de sortie associée à la règle R_i .

Cette méthode de défuzzification est très simple à implémenter. En effet, les centres de gravités des fonctions d'appartenance sont connus a priori. Cependant, son inconvénient est qu'elle n'utilise pas la forme entière des fonctions d'appartenance de la conclusion. Elle utilise seulement le centre \bar{y}_i du support de la fonction d'appartenance de la conclusion, sans tenir compte du fait que la fonction d'appartenance est étroite ou large.

1.8.4.5 Méthode des hauteurs pondérées modifiée

Contrairement à la méthode précédente, la sortie du système flou est calculée en tenant compte du support de la fonction d'appartenance de la conclusion, et est égale à :

$$\bar{y} = \frac{\sum_i \bar{y}_i \mu_i / \delta_i^2}{\sum_i \mu_i / \delta_i^2} \quad (I.26)$$

Où δ est une mesure de l'étendue de la fonction d'appartenance de la variable de sortie associée à la règle R_i . Pour des fonctions d'appartenance de type triangulaire ou trapézoïdale, δ peut être la base du triangle ou du trapèze, pour une fonction gaussienne, il peut être sa variance.

1.9 ANALYSE ET DESCRIPTION DU SYSTEME FLOU DE SUGENO

Il existe un grand nombre de possibilités de réalisation des systèmes flous. En effet, pour chaque module constitutif d'un système flou (figure I.4), il existe une multitude de choix différents, et chaque combinaison de ces choix engendre une classe de systèmes flous. Dans ce paragraphe, nous nous intéressons au modèle flou de type Sugeno, développé initialement par Sugeno et Takagi.

Pour la modélisation des systèmes à partir de données numériques [17]. Une règle typique dans le système de Sugeno est de la forme suivante :

$$R_i : \text{SI } x_1 \text{ est } A_1^i \text{ et } \dots \text{ et } x_n \text{ est } A_n^i \quad \text{ALORS} \quad y_i = f(x) \quad (\text{I.27})$$

Où A_j^i , sont des ensembles flous dans l'espace d'entrée, tandis que $f(x)$ est une fonction numérique dans l'espace de sortie. En générale, $f(x)$ est une fonction arbitraire tant elle puisse décrire convenablement le comportement du système étudié. Si $f(x)$ est un polynôme d'ordre un, on dit que le système de Sugeno est d'ordre un, et ainsi de suite. Etant donné que chaque règle possède une conclusion numérique, la sortie finale du système flou est obtenue par le calcul de la moyenne pondérée de chaque règle floue, et de cette manière, le temps consommé par la procédure de défuzzification est évité.

Comme notre but est de développer une loi d'adaptation des paramètres du système flou. Il est donc indispensable de donner l'expression analytique de la sortie du système flou de Sugeno, et de présenter ensuite l'algorithme d'estimation des paramètres du système flou pour approximer une fonction non linéaire quelconque à partir des données numériques.

1.9.1 Architecture du système flou de Sugeno

Notons les entrées du système flou par x_1, \dots, x_n , et par y sa sortie. Pour chaque x_i est associé m_i ensembles flous A_j^i dans U_i , tel que pour x_i quelconque de U_i , il existe au moins une fonction d'appartenance. La base des règles du système flou comporte $M = \prod_{i=1}^n m_i$ règles telles que :

$$R_i : \text{SI } (x_1 \in A_1^{i_1}) \text{ et } (x_2 \in A_2^{i_2}) \text{ et } \dots \text{ et } (x_n \in A_n^{i_n}) \quad \text{ALORS } y_i(x) = a_{0i} + a_{1i}x_1 + \dots + a_{ni}x_n \quad (\text{I.28})$$

Chaque règle floue correspond à une combinaison de $(A_1^{i_1}, A_2^{i_2}, \dots, A_n^{i_n})$ ensembles flous où $1 \leq i_j \leq m_j$. En effet la base de connaissance comporte toutes les combinaisons possibles des ensembles flous des variables d'entrées.

La sortie du système flou est donnée par la relation suivante [25] :

$$\hat{y} = \frac{\sum_{i=1}^M y_i \mu_i}{\sum_{i=1}^M \mu_i} \quad (\text{I.29})$$

avec

$$\mu_i = \prod_{j=1}^n \mu_{A_j^{i_j}}(x_j) \quad (\text{I.30})$$

Et qui représente le degré d'activation de la règle R_i .

Dans le cas où les paramètres des prémisses sont fixés a priori par l'utilisateur, les seuls paramètres ajustables sont ceux de la conclusion. Ainsi, la sortie finale peut être écrite sous la forme suivante :

$$y = W(x) A \quad (\text{I.31})$$

où A est une matrice regroupant les paramètres a_i
 W est une matrice de fonction floues de la forme :

$$W_{ij}(x) = \frac{\mu_i x_j}{\sum_{l=1}^M \mu_l} \quad (\text{I.32})$$

1.10 ADAPTATION DES SYSTEMES FLOUS

Compte tenu du nombre important de paramètres à déterminer (paramètres des prémisses et des conclusions), la détermination des paramètres d'un système flou est la phase la plus délicate dans sa conception. De ce fait, un expert ne pourra garantir un réglage optimal du système flou qu'après un certain nombre d'essais, et ce malgré une très grande rigueur et une méthodologie conséquente [8], [13].

Des techniques d'optimisation et d'apprentissage ont été développées dans le but de remédier à ce problème. Ainsi l'apport des réseaux de neurones artificiels, a permis de résoudre en partie le problème d'apprentissage, et d'élaborer par la suite des systèmes flous répondants à des critères de performances et d'optimisation.

Un exemple de ces algorithmes d'apprentissage est celui dérivé de la technique dite de Kalman étendu[25]. Considérons un système flou caractérisé par un vecteur de paramètres θ . Et soit un ensemble de données d'entrées/sorties. Notre objectif est de trouver la valeur du vecteur θ pour que la sortie du système flou approxime le mieux possible la sortie désirée. Cette approche donne, finalement, l'algorithme d'apprentissage suivant :

$$\theta(t+1) = \theta(t) + \eta(t) \frac{\partial f(x, \theta)}{\partial \theta} \Big|_{\theta=\theta(t)} e(t) \quad (\text{I.33})$$

et

$$f(x, \theta) = \hat{y} = \frac{\sum_{i=1}^l \mu_i \hat{y}_i}{\sum_{i=1}^l \mu_i} \quad (\text{I.34})$$

Avec : $\eta(t)$ le gain d'adaptation paramétrique.

$\theta(t)$ vecteur des paramètres à ajuster.

$\hat{y}(t)$ la sortie approximée.

$e(t)$ différence entre la sortie réelle et la sortie approximée.

Les différentes méthodes d'identification récursives citées dans la littérature se distinguent essentiellement par le choix du gain, qui dans tous les cas, doit être pris égale à une matrice positive. Les méthodes les plus utilisées sont la méthode du gradient, la méthode du gradient modifié et la méthode des moindres carrés récursives.

- **Méthode du gradient**

C'est la méthode la plus simple à mettre en œuvre. Le gain $\eta(t)$ est choisit égale au produit d'un gain scalaire par la matrice identité

$$\eta(t) = \alpha I, \quad \alpha > 0 \quad (I.35)$$

- **Méthode du gradient modifié**

Dans cette méthode, le gain $\eta(t)$ varie à chaque itération, il est donné par la relation suivante :

$$\eta(t) = \frac{\alpha_1 I}{\alpha_2 + \left(\frac{\partial f}{\partial \theta}\right)^T \left(\frac{\partial f}{\partial \theta}\right)}, \quad \alpha_1 > 0, \alpha_2 > 0 \quad (I.36)$$

- **Méthode des moindres carrés récursives**

Dans ce cas le gain d'estimation est donné par la relation suivante :

$$\eta(t) = \eta(t-1) + \frac{\left(\frac{\partial f}{\partial \theta}\right) \eta(t-1) \left(\frac{\partial f}{\partial \theta}\right)^T}{1 + \left(\frac{\partial f}{\partial \theta}\right) \eta(t-1) \left(\frac{\partial f}{\partial \theta}\right)^T} \quad (I.37)$$

avec $\eta(0) = \alpha I$ où $\alpha > 0$.

Nous remarquons que ces méthodes nécessitent toutes le calcul du gradient $\left(\frac{\partial f}{\partial \theta}\right)$, ce dernier est calculé par la méthode de rétropropagation utilisée dans les réseaux de neurones [8], [13].

1.11 APPLICATION DE LA LOGIQUE FLOUE A L'IDENTIFICATION :

1.11.1 Introduction

Nous allons dans le présent paragraphe présenter les principaux résultats obtenus pour l'approximation du modèle dynamique du robot PUMA 560 (annexe I) par trois sous systèmes, chacun d'eux correspond à une articulation du bras manipulateur.

Partant d'un système multivariable non linéaire ayant une équation d'état de la forme

$$\dot{X} = F(X) + G(X) U \quad (I.38)$$

Où :

X vecteur d'état du système élément de R^n .

U commande ou entrée du système élément de R^p .

On désire approximer ce dernier par k sous systèmes de la forme

$$\begin{cases} x_1^k = x_2^k \\ x_2^k = x_3^k \\ \vdots \\ x_m^k = a_0^k + a_1^k x_1^k + \dots + a_m^k x_m^k + b^k u_1 \end{cases} \quad (I.39)$$

Où k est l'indice du $k^{\text{ème}}$ sous système.

Pour la simplicité de la notation nous écrivons x_i au lieu d'écrire x_i^k , il faut seulement se rappeler que l'algorithme qui sera présenté par la suite est à utiliser pour chaque sous système.

Afin de déterminer les valeurs des paramètres a_i de chaque sous système nous utilisons un modèle flou de Takagi-Sugeno ou la $i^{\text{ème}}$ règle floue s'écrit

$$\text{Si } (x_1 \in A_1^i) \text{ et } (x_2 \in A_2^i) \text{ et } \dots \text{ et } (x_n \in A_n^i) \text{ et } (b \in B) \text{ alors } \hat{y}_i = a_{0i} + a_1 x_1 + \dots + a_n x_n + b_i u \quad (I.40)$$

Si on note u par x_{n+1} et b_i par a_{n+1i} , alors le degré d'activation de chaque règle est donné par

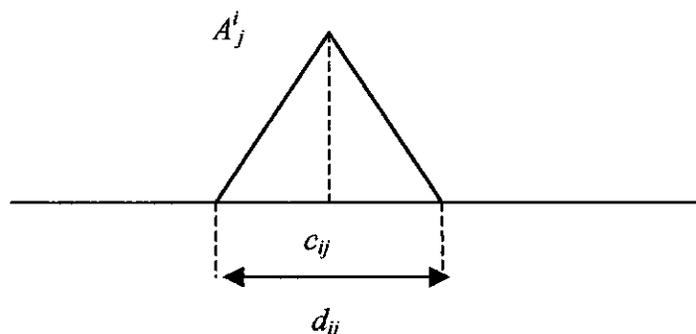
$$\mu_i = \prod_{j=1}^{n+1} \mu_{A_j^i}(x_j) \quad (I.41)$$

La sortie approximée est donc

$$\hat{y} = \frac{\sum_{i=1}^l \mu_i \hat{y}_i}{\sum_{i=1}^l \mu_i} \quad (I.42)$$

Où l représente le nombre total de règles floues.

Chaque variable ayant m ensembles flous (fonction d'appartenance) $\mu_{A_j^i}$, d'une forme triangulaire, par exemple.



Figure(1.6) fonction d'appartenance en triangle

Pour les triangles isocèles, l'équation est complètement déterminée en connaissant le centre et la largeur de la base. D'où

$$\mu_{A_j}(x_j) = \begin{cases} 1 - 2|x_j - c_{ij}|/d_{ij} & ; |x_j - c_{ij}| \leq \frac{d_{ij}}{2} \\ 0 & \text{autrement} \end{cases} \quad (\text{I.43})$$

les arguments des prémisses (c_{ij} et d_{ij}) ainsi que ceux des conséquences (a_{ij}) sont ajustés par un algorithme d'adaptation paramétrique de forme générale

$$\theta(t+1) = \theta(t) + \eta(t) \frac{\partial f(x, \theta)}{\partial \theta} \Big|_{\theta=\theta(t)} e(t) \quad (\text{I.44})$$

Où :

$$f(x, \theta) = \hat{y} = \frac{\sum_{i=1}^I \mu_i \hat{y}_i}{\sum_{i=1}^I \mu_i} \quad (\text{I.45})$$

$\theta = [\theta^1 \ \theta^2 \ \dots \ \theta^I]^T$ vecteur des paramètres à ajuster. et chaque θ^i étant défini par :

$$\theta^i = [a_{i0} \ \dots \ a_{im} \ c_{i1} \ \dots \ c_{im} \ d_{i1} \ \dots \ d_{im}]$$

$$e(t) = y_{\text{réel}} - f(x, \theta)$$

$\eta(t)$ le gain d'adaptation paramétrique.

Pour notre application, nous avons opté pour l'algorithme du type gradient modifié dont l'expression du gain est donné par l'équation (I.36).

Aussi l'expression de la dérivée de la fonction f est donnée par :

$$\left(\frac{\partial f}{\partial \theta} \right)^T = \left[\frac{\partial f}{\partial \theta^1} \quad \dots \quad \frac{\partial f}{\partial \theta^I} \right] \quad (\text{I.46})$$

et on définit :

$$\frac{\partial f}{\partial \theta^i} = \left[\left(\frac{\partial f}{\partial a_{0i}} \right) \quad \left(\frac{\partial f}{\partial a_{ij}} \right) \quad \left(\frac{\partial f}{\partial c_{ij}} \right) \quad \left(\frac{\partial f}{\partial d_{ij}} \right) \right] \quad (\text{I.47})$$

Ce qui donne pour les fonctions d'appartenances choisies, les formules suivantes :

$$\left\{ \begin{array}{l} \frac{\partial f}{\partial a_{i0}} = \frac{\mu_i}{\sum_{i=1}^I \mu_i} \\ \frac{\partial f}{\partial a_{ij}} = \frac{\mu_i}{\sum_{i=1}^I \mu_i} x_j \\ \frac{\partial f}{\partial c_{ij}} = \frac{\hat{y}_i - \hat{y}}{\mu_{A_j^i(x_j)} \sum_{i=1}^I \mu_i} \left(\frac{2 \mu_i}{d_{ij}} \text{sign} (x_j - c_{ij}) \right) \\ \frac{\partial f}{\partial d_{ij}} = \frac{\hat{y}_i - \hat{y}}{\mu_{A_j^i(x_j)} \sum_{i=1}^I \mu_i} \left(\frac{2 \mu_i}{(d_{ij})^2} |x_j - c_{ij}| \right) \end{array} \right. \quad (I.48)$$

avec $\text{sign}(x)$ est la fonction signe définie par :

$$\text{sign}(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -1 & \text{si } x < 0 \end{cases} \quad (I.49)$$

L'équation d'état de chaque sous système est donc

$$\begin{cases} \dot{x} = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} x + \begin{bmatrix} 0 \\ b \end{bmatrix} u + \begin{bmatrix} 0 \\ a_0 \end{bmatrix} \\ y = [1 \quad 0] x \end{cases} \quad (I.50)$$

avec

$$a_j = \frac{\sum_{i=1}^I a_{ij} \mu_i}{\sum_{i=1}^I \mu_i} \quad \text{et} \quad b = \frac{\sum_{i=1}^I b_i \mu_i}{\sum_{i=1}^I \mu_i} \quad (I.51)$$

1.11.2 Résultats et commentaires

Après avoir développé toutes les notions théoriques nécessaires pour l'approximation de notre système interconnecté par trois sous systèmes découplés, nous présentons dans cette section les résultats obtenus avec un algorithme d'apprentissage du type gradient modifié. Nous avons inclus dans le vecteur des paramètres à ajuster les coefficients des prémisses et des conséquences du système flou de Sugeno. Le programme ainsi fait se doit d'ajuster les paramètres pour permettre l'approximation de chaque articulation du système de départ par un système linéaire. Ce dernier est estimé à l'instant $t/2$ de chaque intervalle $[t, t+1]$. Ce qui nous permet d'obtenir plusieurs sous systèmes pour divers points de fonctionnements.

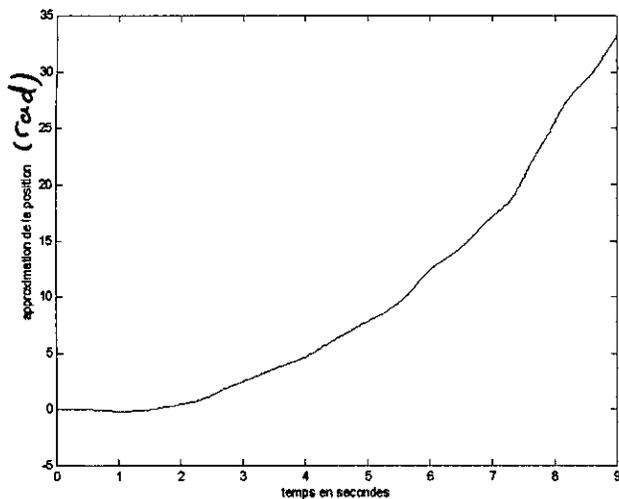
L'algorithme choisi pour l'apprentissage ne permettant pas d'éviter les minimums locaux, nous étions obligés de changer plusieurs fois les valeurs initiales ainsi que les paramètres α_1 et α_2 du gain équation (I.38). Après plusieurs essais nous avons abouti aux résultats du tableau(I.1), l'erreur étant représentée sur la figure (I.7).

- *Première articulation*

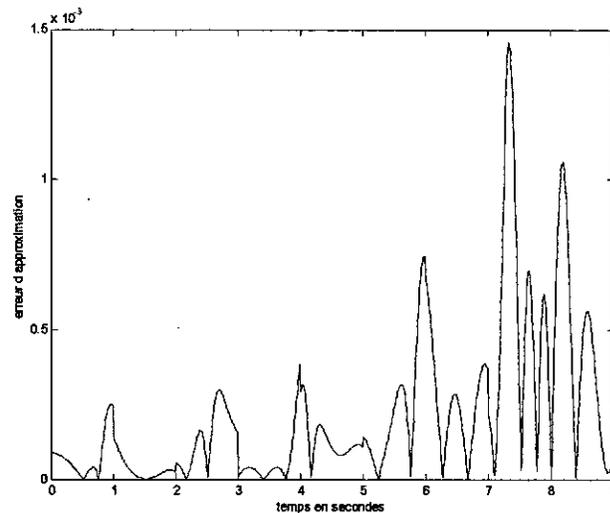
Intervalle	a_0	a_1	A_2	b	Erreur max.
[0, 1]	-0.5537	0.1921	0.3349	-0.8537	0.0003
[1, 2]	0.5334	0.1993	0.2237	0.2334	0.0003
[2, 3]	0.6485	0.3844	0.6014	0.3485	0.0003
[3, 4]	0.3644	-0.2758	-0.0688	0.3356	0.0004
[4,5]	0.4743	0.0680	0.1514	0.1743	0.0004
[5, 6]	0.4922	0.1435	0.1847	0.2078	0.0007
[6, 7]	0.4836	0.0349	0.1646	0.2164	0.0007
[7, 8]	0.4742	-0.1161	0.1530	0.1742	0.0015
[8, 9]	0.4819	-0.0576	0.1452	0.1819	0.0015

Tableau(I.1) valeurs des paramètres de sortie du système flou

La figure (I.7) représente la réponse indicielle du système initial ainsi que celle des sous systèmes combinés. On remarque que les deux réponses se confondent et que l'erreur est très faible relativement à l'ordre de grandeur de la sortie. Il en est de même pour la vitesse représentée par la figure (I.8), mais on remarque toute fois que l'erreur est légèrement plus importante que celle de la position.

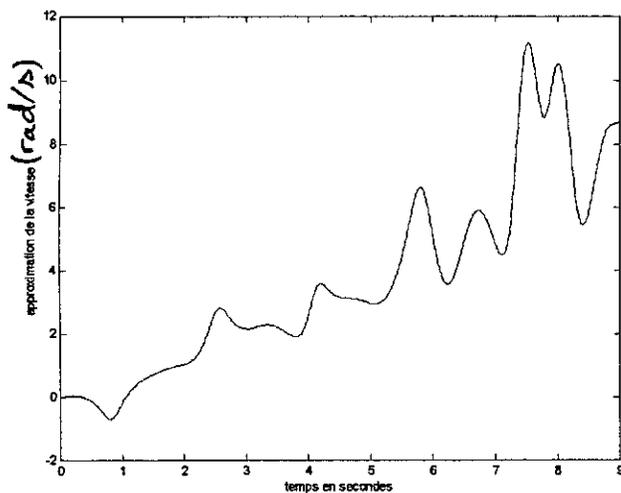


(a) position de la première articulation

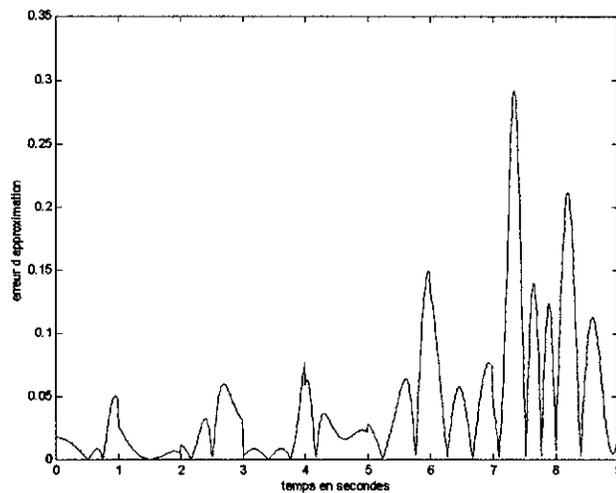


(b) erreur en position

Figure (1.7) réponse indicielle de la première articulation



(a) vitesse de la première articulation



(b) erreur en vitesse

Figure(1.8) vitesse de la première articulation pour une réponse indicielle

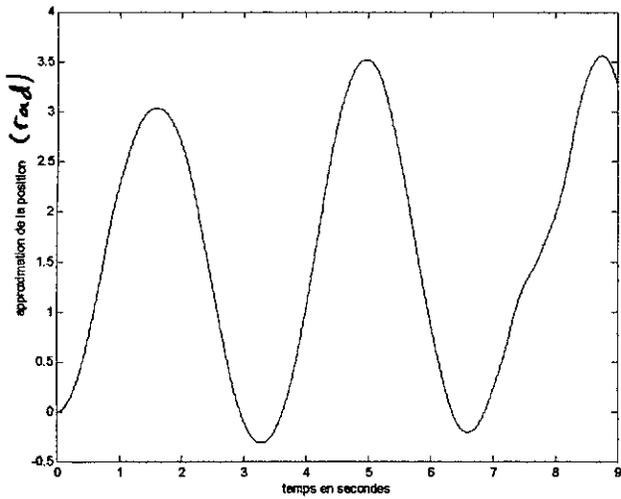
- *Seconde articulation*

Nous présentons ici un tableau récapitulatif des résultats obtenus pour l'approximation de la seconde articulation.

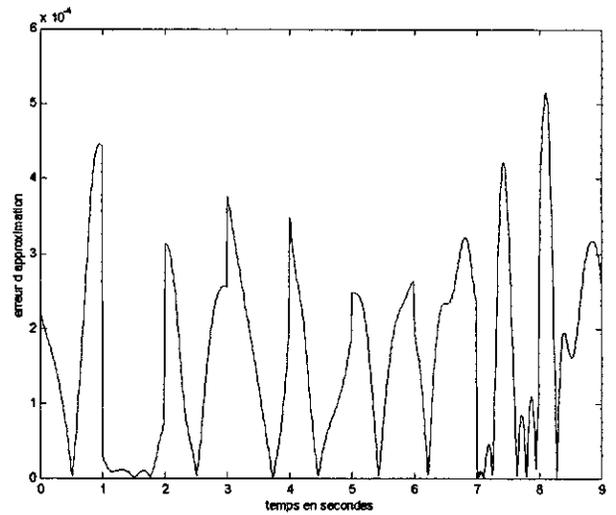
Intervalle	a_0	a_1	A_2	b	Erreur max.
[0, 1]	0.7881	0.4113	1.0062	0.4881	0.0004
[1, 2]	-0.0171	-1.3642	-0.0038	-0.3171	0.0004
[2, 3]	0.4895	0.1873	0.2350	0.1895	0.0004
[3, 4]	1.9836	-0.1060	1.8465	-1.2836	0.0004
[5, 6]	0.1234	-0.8195	-0.8830	-0.1766	0.0004
[6, 7]	0.1953	-0.6562	1.0187	0.5047	0.0004
[7, 8]	1.7981	-0.2238	-1.7515	-1.0981	0.0004
[8, 9]	0.5528	0.2469	0.3995	0.2528	0.0004

Tableau(1.2) valeurs des paramètres de sortie du système flou

Comme pour la première articulation les figures (I.10) et (I.11) représentent respectivement la réponse indicielle en position et la vitesse de l'articulation. Les mêmes remarques sont à faire : l'erreur est relativement plus importante en vitesse qu'en position. Ceci est dû au fait que dans l'équation d'état la dérivée de la position est égale à la vitesse, et ne dépend pas des paramètres de la sortie du système flou. Par contre la vitesse dépend de tous les paramètres estimés. Cette remarque est valable pour les trois articulations.

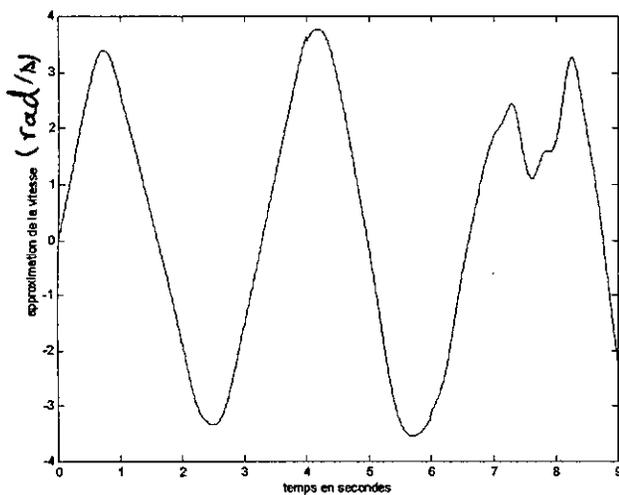


(a) position de la seconde articulation

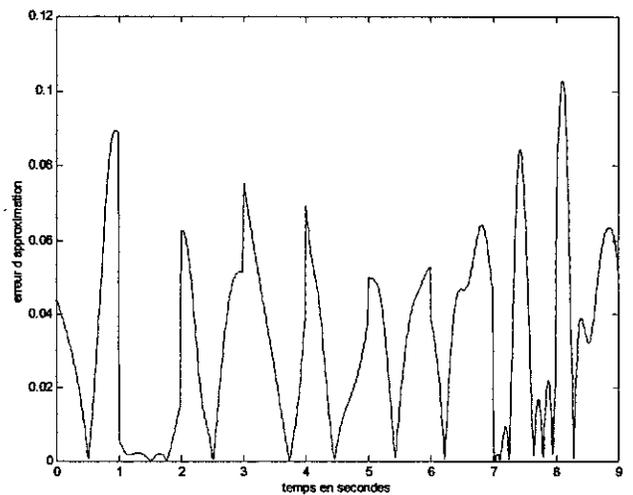


(b) erreur en position

figure(1.9) réponse indicielle de la seconde articulation



(a) vitesse de la seconde articulation



(b) erreur en vitesse

figure(1.10) vitesse de la seconde articulation pour une réponse indicielle

- *Troisième articulation*

Comme nous avons fait pour les deux premières articulations, nous présentons ci dessous un tableau récapitulatif des résultats obtenus pour l'approximation de l'équation d'état de la troisième articulation, ainsi que les tracés de la réponse indicelle en position et la vitesse angulaire de l'articulation.

Intervalle	a_0	a_1	a_2	b	Erreur max.
[0, 1]	2.7146	0.3060	2.0838	2.4164	0.0008
[1, 2]	0.5389	0.3369	0.3403	0.2389	0.0008
[2, 3]	0.3876	-0.9085	-0.4743	0.0876	0.0008
[3, 4]	0.4791	-0.0712	0.1704	0.2209	0.0008
[5, 6]	0.4577	-0.4456	0.1703	0.1577	0.0008
[6, 7]	0.4556	-0.4636	0.2738	0.2444	0.0012
[7, 8]	0.4825	-0.0300	1.1973	0.2175	0.0012
[8, 9]	0.5452	0.7903	0.2840	0.2425	0.0028

Tableau (1.3) valeurs des paramètres de sortie du système flou

La figure(I.11) représente la réponse indicelle de la troisième articulation. Les deux réponses (celle du système réel et de l'approximateur) se confondent et l'erreur en position est toujours plus faible que celle de la vitesse. Ceci est dû, comme nous l'avons dit auparavant, au fait que les paramètres du système flou n'interviennent pas de manière directe dans l'équation différentielle de la position.

CHAPITRE II

COMMANDE H_∞ DECENTRALISEE

Dans ce chapitre nous commençons par introduire toutes les notions théoriques requises pour la définition de la norme infinie d'une matrice, ensuite tous les objectifs de commande sont traduits en norme infinie. Enfin nous exposons l'algorithme de Mc Farlane et Glover qui permet une synthèse rapide et peu compliquée du contrôleur H_∞ . Ce dernier est appliqué au robot PUMA 560

II.1 NORME H_∞ :

On peut probablement dire que l'approche H_∞ est, dans le domaine de la commande robuste, le sujet qui a donné lieu au plus grand nombre de publications depuis le milieu des années 80. Cela n'est pas par hasard, car c'est une façon très intuitive de formuler la problématique de la robustesse.

Tout a commencé par le travail de Zames en 1981, et suivi par d'autres articles écrits par, ou en collaboration avec le même auteur [5] [6]. Ces travaux ont été mal compris par beaucoup quand ils ont paru, et ils ne concernaient pas le problème de la robustesse mais plutôt celui du rejet des perturbations. C'est Kurima qui, en 1984, a formulé le premier le problème de la robustesse [10]. La nouveauté apportée, réside dans le fait de l'utilisation explicite de l'interpolation dite de nevanlinna-pick. Et ce n'est qu'en 1987 que Francis [4] a fait une très bonne synthèse des travaux parus jusqu'à cette date et les a unifiés grâce à la notion de problème standard, qui est due à Doyle.

La synthèse d'un contrôleur H_∞ en résolvant le problème standard a progressé d'une manière très importante en 1988 avec l'algorithme de Glover Doyle [7], qui utilise la représentation d'état. Une autre façon de résoudre le problème de la commande robuste dans le cadre H_∞ a été développée par Mc Farlane et Glover [14] en 1990, en utilisant la factorisation coprime normalisée, cette factorisation existait déjà, mais la nouveauté réside dans le fait qu'elle soit particulière, car ils ont bien traduit le problème de la commande robuste en facteurs coprimes normalisés. La particularité de cette méthode est qu'elle utilise les concepts de l'automatique classique dans la synthèse du contrôleur et ceci via la technique de loop-shaping.

II.1.1 Valeurs singulières et norme H_∞ d'une matrice :

Soit $G(s)$ la matrice de transfert d'un système à n entrées et m sorties, et soit q le plus petit des entiers m et n . La matrice $G(j\omega)G(-j\omega)^T$ est Hermitienne semi-définie positive, et ses valeurs propres sont donc toutes réelles positives ou nulles. Les racines carrées de ces q valeurs propres sont appelées valeurs singulières de $G(j\omega)$.

$$\sigma_i(G(j\omega)) = \sqrt{\lambda_i(G(j\omega)G(-j\omega)^T)} \quad (\text{II.1})$$

On note en général $\bar{\sigma}(G)$ la plus grande valeurs singulière et $\underline{\sigma}(G)$ la plus petite valeurs singulières. Quelques propriétés des valeurs singulières doivent être données, à savoir :

$$\bar{\sigma}(G) = \max_{x \in \mathbb{C}^n} \frac{\|Gx\|_2}{\|x\|_2} \quad (\text{II.2})$$

$$\underline{\sigma}(G) = \min_{x \in \mathbb{C}^n} \frac{\|Gx\|_2}{\|x\|_2} \quad (\text{II.3})$$

La relation (II.2) montre que la valeur singulière supérieure est la norme matricielle induite par la norme euclidienne usuelle.

Considérons à présent une excitation harmonique $e(t) = A \exp(j\omega t)$. La sortie du système s'écrit :

$$s(t) = G(j\omega) A \exp(j\omega t)$$

Les relations (II.2) et (II.3) permettent d'écrire :

$$\underline{\sigma}(G(j\omega)) \leq \frac{\|G(j\omega)A \exp(j\omega t)\|_2}{\|A \exp(j\omega t)\|_2} \leq \overline{\sigma}(G(j\omega)) \quad (\text{II.4})$$

Les valeurs singulières $\sigma_i(G(j\omega))$ constituent donc une généralisation aux systèmes multivariables de la notion de gain. Elles peuvent être représentées dans le plan de Bode classique. Pour un système multivariable, le gain à une fréquence donnée dépend donc en fait du vecteur complexe A , et sera compris entre les valeurs singulières supérieure et inférieure.

De nombreuses propriétés s'expriment en terme de gain pour les systèmes multivariables en considérant les fonctions $\overline{\sigma}(G(j\omega))$ et $\underline{\sigma}(G(j\omega))$. On dira que le gain est faible si la valeur $\overline{\sigma}(G(j\omega))$ est petite et que le gain est fort si la valeur $\underline{\sigma}(G(j\omega))$ est grande. Notons finalement que les valeurs singulières ne permettent pas la généralisation de la notion de phase.

Enfin la norme H_∞ d'une matrice de transfert est définie de la manière suivante :

$$\|G\|_\infty = \sup_\omega \overline{\sigma}(G(j\omega)) \quad (\text{II.5})$$

Pour un système monovariante, la norme H_∞ est simplement la valeur maximale du gain de $G(j\omega)$.

II.1.2 Robustesse :

Généralement la mise en équations d'un système physique nécessite des approximations, il en résulte par conséquent des incertitudes de modèle. Celles-ci peuvent être regroupées en deux grandes classes : les incertitudes de modèle et les dynamiques négligées. L'exemple d'un avion permet d'illustrer ce point de vue : son comportement est décrit par les lois physiques connues (la mécanique du vol), dont les paramètres sont mesurés en soufflerie avec une plage d'incertitudes non négligeable, et varie fortement en fonction des conditions de vol. De plus, la synthèse du correcteur fait appel à un modèle simplifié, dans lequel sont négligées, par exemple, les dynamiques hautes fréquences du corps solide, celles des capteurs ou des actionneurs, et d'éventuels retards pur.

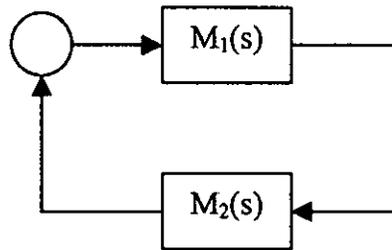
On dira que la commande est robuste si le comportement du système asservi est relativement insensible à ces différentes incertitudes. En particulier la *robustesse en stabilité* sera obtenue si la stabilité du système est conservée. Une qualité plus exigeante est la recherche d'une robustesse en *performances*, au sens d'un critère à spécifier.

Une commande est dite robuste si elle garantit la stabilité et/ou les performances, non pour un seul modèle mais pour toute une classe (famille) de systèmes définis à partir d'une marge d'erreur.

II.1.3 Théorème du petit gain :

Avant de donner les différentes conditions du maintien de stabilité issues de cette approche, nous rappelons la forme classique du théorème du petit gain.

Considérons le système représenté sur la figure (II.1)



Figure(II.1)

Où $M_1(s)$ et $M_2(s)$ sont deux matrices de transfert propres et stables. Le théorème du petit gain classique s'énonce de la façon suivante :

• **Théorème [29] :**

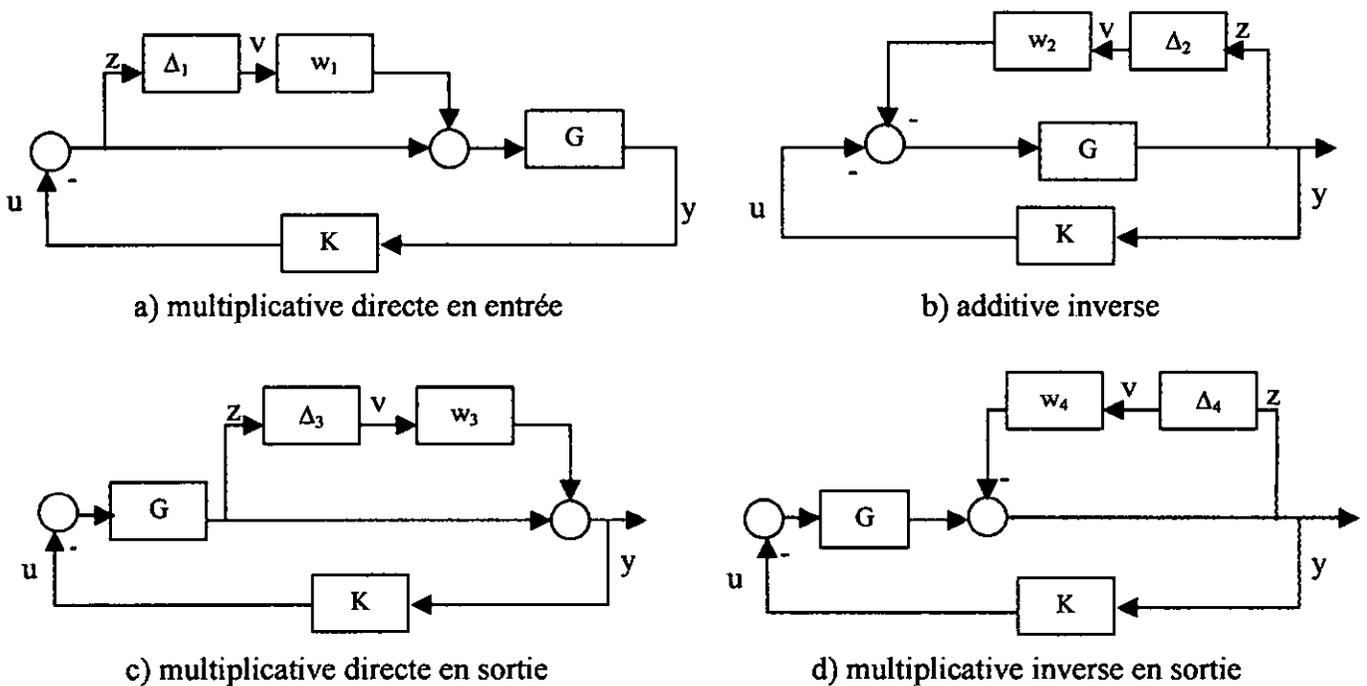
Le système reste stable au sens de Lyapunov si et seulement si $\|M_2(s)M_1(s)\|_\infty < 1$

Une autre condition impliquant la précédente et plus adaptée à l'étude de la robustesse en stabilité est donnée par :

$$\|M_1(s)\|_\infty \|M_2(s)\|_\infty < 1$$

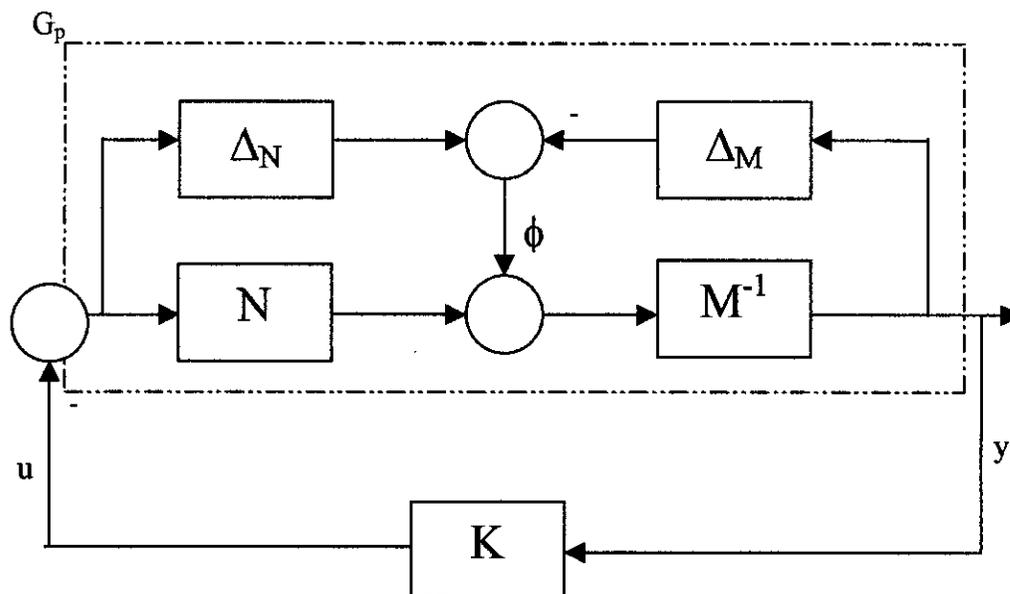
II.2 INCERTITUDES DE MODELISATION [32] :

Une représentation générale des incertitudes de modélisation est donnée par la figure(II.2a) qui rassemble quelques formes courantes de ces dernières.



Figure(II.2-a) : quelques formes de représentations des incertitudes de modèle

Une autre forme intéressante d'incertitudes de modélisation est l'incertitude sous forme de facteurs premiers représentée par la figure (II.2-b)



Figure(II.2-b) décomposition en facteurs premiers

De façon générale la matrice $\Delta(s)$ est dite non structurée si elle vérifie la condition :

$$\|\Delta\|_\infty = \sup_{\omega} \bar{\sigma}(\Delta(j\omega)) \leq 1 \tag{II.6}$$

II.3 ROBUSTESSE EN STABILITE :

On dira que le système est robuste en stabilité s'il reste stable pour toute incertitude $\Delta(s)$ vérifiant (II.6). Si nous supposant que le modèle nominal $G(s)$ est stable, la seule source d'instabilité est le bouclage par la matrice $\Delta(s)$. Il est donc équivalent d'étudier la stabilité du système représenté par la figure (II.1). On peut alors établir la proposition suivante :

Si $M(s)$ et $\Delta(s)$ n'ont que des pôles à partie réelle négative, le système de la figure(II.2) est stable pour tout $\Delta(s)$ tel que $\|\Delta\|_\infty \leq 1$ si et seulement si $\|M\|_\infty \leq 1$.

Si en effet, cette condition est vérifiée, alors :

$$\bar{\sigma}(\Delta M) \leq \bar{\sigma}(\Delta) \bar{\sigma}(M) < 1, \forall \omega \tag{II.7}$$

Réciproquement si $\|M\|_\infty \geq 1$ on peut construire un Δ particulier tel que $I - \Delta M$ soit singulière pour une fréquence particulière, ce qui rend le système instable.

Ce résultat peut être appliqué à chacune des formes de la figure(II.2), en l'identifiant à la figure(II.1) où $\Delta = M_1$ et $M = M_2$, et nous obtenons :

- $M = (I + KG)^{-1} KG w_1$ pour des incertitudes sous formes multiplicatives directe en entrée.

- $M = (I+GK)^{-1}Gw_2$ pour des incertitudes sous formes additives inverses en entrée.
- $M = (I+GK)^{-1}GKw_3$ pour des incertitudes sous formes multiplicatives directe en sortie.
- $M = (I+GK)^{-1}w_4$ pour des incertitudes sous formes multiplicatives inverses en sortie.

Pour la décomposition en facteurs premiers le schéma équivalent est très bien détaillé dans le paragraphe (II.6.2).

On remarque que la condition de robustesse en stabilité donnée ci-dessus est nécessaire et suffisante, mais à condition de considérer toutes les incertitudes $\Delta(s)$ de norme H_∞ inférieure ou égale à un. En pratique, cet ensemble contient nécessairement des incertitudes de modélisation irréalisables physiquement ou, dont on sait qu'elles n'ont pas à être considérées. Les résultats déduits de cette analyse conduiront nécessairement à des appréciations pessimistes de la robustesse.

Cette limitation provient du fait qu'on rassemble toutes les incertitudes dans un seul bloc $\Delta(s)$. Des résultats moins conservatifs peuvent être obtenus en tenant compte de la structure des incertitudes de modèle.

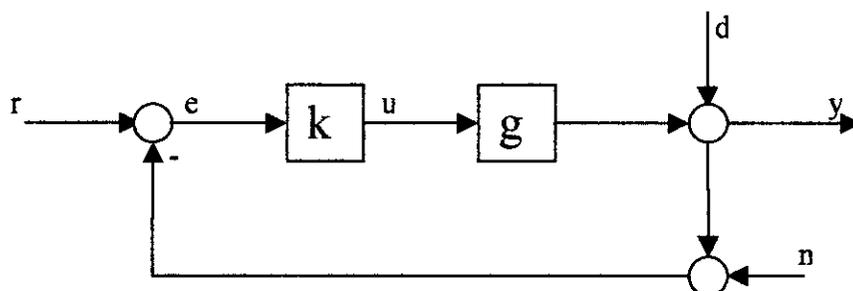
II.4 OBJECTIFS DE LA SYNTHÈSE :

Nous allons dans le présent chapitre traduire les objectifs qu'on désire atteindre sous forme de normes infinies. On définira auparavant des fonctions ou matrices selon le cas MIMO ou SISO qui jouent un rôle prépondérant dans la traduction du cahier de charges en normes infinies et la synthèse du contrôleur adéquat.

Les principaux objectifs à atteindre sont :

- La poursuite : c'est à dire que la sortie doit suivre la consigne appliquée.
- Le rejet des perturbations : l'effet sur la sortie dû à toute perturbation ou entrée exogène, ainsi que la consigne, doit être atténué au maximum.
- L'atténuation des bruits de mesure.

Soit à présent le système de commande suivant :



Figure(II.3) forme générale de la boucle de commande

Où

r : la référence

d : perturbation

n : bruit de mesure

II.4.1 Fonction de sensibilité :

la sensibilité de la sortie y par rapport aux variations d'un paramètre a est définie par :

$$S_a^y = \left(\frac{\partial y / y}{\partial a / a} \right) \quad (\text{II.8})$$

Bode fut le premier à étudier la sensibilité d'un système en boucle fermée par rapport aux variations du modèle.

Pour le système de la figure(II.3) on a :

$$y = [gk / (1 + gk)] r \quad (\text{II.9})$$

$$\text{D'ou} \quad S_g^y = 1 / (1 + gk) \quad (\text{II.10})$$

On définit la fonction sensibilité par :

$$S = 1 / (1 + gk) \quad (\text{II.11})$$

On définit aussi la fonction de sensibilité complémentaire par :

$$T = gk / (1 + gk) \quad (\text{II.12})$$

S est l'inverse de la fonction de différence retour $(1 + gk)$

T est la fonction de transfert du système en boucle fermée.

Nous avons $S + T = 1$

La relation (II.8) montre que l'effet des variations du modèle sur la sortie du système en boucle fermée dépend de la fonction de sensibilité S

Les relations (II.8) et (II.10) nous permettent d'écrire :

$$|S| \ll 1 \Rightarrow \text{le système est insensible aux variations de } g.$$

$$|S| > 1 \Rightarrow \text{les variations de } g \text{ sont amplifiées à la sortie.}$$

Mais il ne faut pas oublier que la fonction de sensibilité est définie par une dérivée c'est à dire que la relation (II.10) n'est valable que pour les faibles variations paramétriques de g .

Dans le cas MIMO, le modèle et le contrôleur sont représentés par les matrices de transfert G , K , S , et T qui sont définies par :

$$S = (I + GK)^{-1} \quad (\text{II.13})$$

$$T = GK (I + GK)^{-1} \quad (\text{II.14})$$

En multivariable les 'gains' de S et de T n'étant plus définis, on utilise les valeurs singulières afin de voir l'effet des faibles variations paramétriques de G sur le système en boucle fermée.

La relation (II.12) nous permet d'écrire que :

- La sensibilité est faible si sa plus grande valeur singulière est faible.
- La sensibilité est élevée si sa plus petite valeur singulière est grande.

$$\overline{\sigma}(S) \ll 1 \Rightarrow \text{La sensibilité est faible.}$$

$$\underline{\sigma}(S) > 1 \Rightarrow \text{Le système est plus sensible qu'en boucle ouverte.}$$

Mais généralement à cause du fait que $S + T = I$.

On ne peut avoir $\overline{\sigma}(S(j\omega)) < 1, \forall \omega$.

Il y a d'autres spécifications à vérifier, c'est ce que nous allons voir dans les sections suivantes.

II.4.2 Performances nominales :

On utilise les valeurs singulières pour analyser les performances d'un système multivariable.

Du système de la figure (II.3) nous avons les relations suivantes :

$$y(s) = GK(I+GK)^{-1}(r - n) + (I+GK)^{-1}d \quad (\text{II.15})$$

$$u(s) = K(I+GK)^{-1}(r - n - d) \quad (\text{II.16})$$

$$e(s) = (I+GK)^{-1}(r - n - d) \quad (\text{II.17})$$

En utilisant S et T :

$$y(s) = T(r - n) + Sd \quad (\text{II.18})$$

$$u(s) = KS(r - n - d) \quad (\text{II.19})$$

$$e(s) = S(r - n - d) \quad (\text{II.20})$$

- **Poursuite :**

La poursuite de r est équivalente à ce que l'erreur soit faible

A partir de (II.20), l'erreur est faible si $\overline{\sigma}(S) \ll 1$

$$\overline{\sigma}(S) \ll 1 \Rightarrow \text{une bonne poursuite.} \quad (\text{I})$$

- **Rejet des perturbations :**

A partir de (II.18), la perturbation d est rejetée si le gain de la fonction de la sensibilité est faible et donc :

$$\overline{\sigma}(S) \ll 1 \Rightarrow \text{rejet des perturbations en sortie.} \quad (\text{II})$$

- **Bruit de mesure :**

A partir de (II.18) le bruit de mesure est atténué si T est faible :

$$\bar{\sigma}(T) \ll 1 \Rightarrow \text{atténuation du bruit de mesure} \quad (\text{III})$$

mais pour une bonne poursuite il faut que $\bar{\sigma}(s) \rightarrow 0$ donc $\bar{\sigma}(T) \rightarrow 1$

Ainsi lorsque le système suit la référence, le bruit de mesure est directement transmis à la sortie.

Remarques :

De (I) et (II) on peut voir que la poursuite de la référence r et le rejet des perturbations d sont compatibles.

A partir de (I) et (II) et (III) et se rappelant que $S + T = I$ on voit qu'il n'est pas possible de réaliser une bonne poursuite et un rejet des bruits de mesure simultanément.

Heureusement que les bruits de mesures sont généralement de haute fréquence, et les consignes à suivre et les perturbations sont de basses fréquence.

Un bon comportement du système revient à avoir :

$$\begin{cases} \bar{\sigma}(S) \text{ faible pour } \omega < \omega_B \\ \bar{\sigma}(T) \text{ faible pour } \omega > \omega_B \end{cases} \quad (\text{II.21})$$

Où ω_B délimite les basses fréquences des hautes fréquences, c'est une sorte de bande passante du système.

Ou en utilisant S seulement

$$\bar{\sigma}(W_s S) < 1 \quad (\text{II.22})$$

Avec W_s une fonction de transfert ayant un gain élevé en basse fréquences, permettant ainsi de garantir l'objectif (II.21)

Plusieurs méthodes dans le cadre H_∞ ont été développées pour la synthèse du contrôleur K. Nous allons présenter celui de H_∞ standard d'une manière très brève, puis celui des facteurs coprimes qui sera d'ailleurs utilisé pour notre application. D'autres méthodes existent elles sont développées dans plusieurs ouvrages, [32] par exemple.

II.5 PROBLEME STANDARD :

Soit le schéma général de la figure (II.4) avec :

y : le vecteur de mesures.

w : les entrées externes (consigne, perturbation).

u : les commandes.

z : les signaux à minimiser (les indicateurs de performances et de robustesse).

Le problème standard est posé de la manière suivante :

Trouver le compensateur $K(S)$ qui stabilise P et minimise $\|T_{zw}\|_\infty$

T_{zw} est la matrice de transfert entre w et z .

$P(S)$ est le système augmenté qui nous permet de générer z correspondant aux objectifs de la commande.

Où P est donnée par sa matrice de transfert :

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \quad \begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \quad (\text{II.23})$$

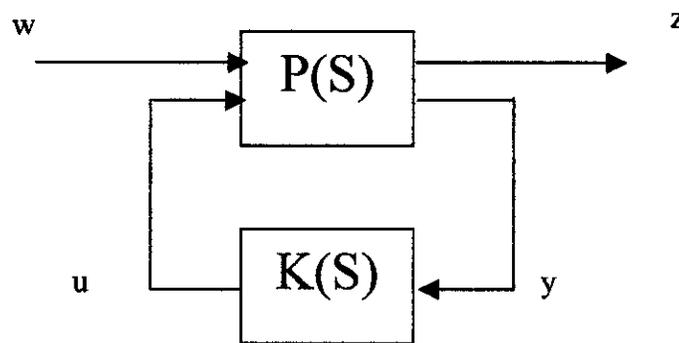


Figure (II.4) problème standard

où sa représentation dans l'espace d'état :

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix} \quad (\text{II.24})$$

Puisque les solutions du problème standard sont connues, alors le seul problème qui reste est de transformer notre problème de commande en un problème standard.

- **Mise des problèmes sous forme standard :**

Soit le système de la figure (II.5), les objectifs de la commande sont

- Rejet de la perturbation d et poursuite de la référence r

$$\Rightarrow \text{minimiser } \|W_S S\|_\infty$$

- Atténuation du bruit de mesure

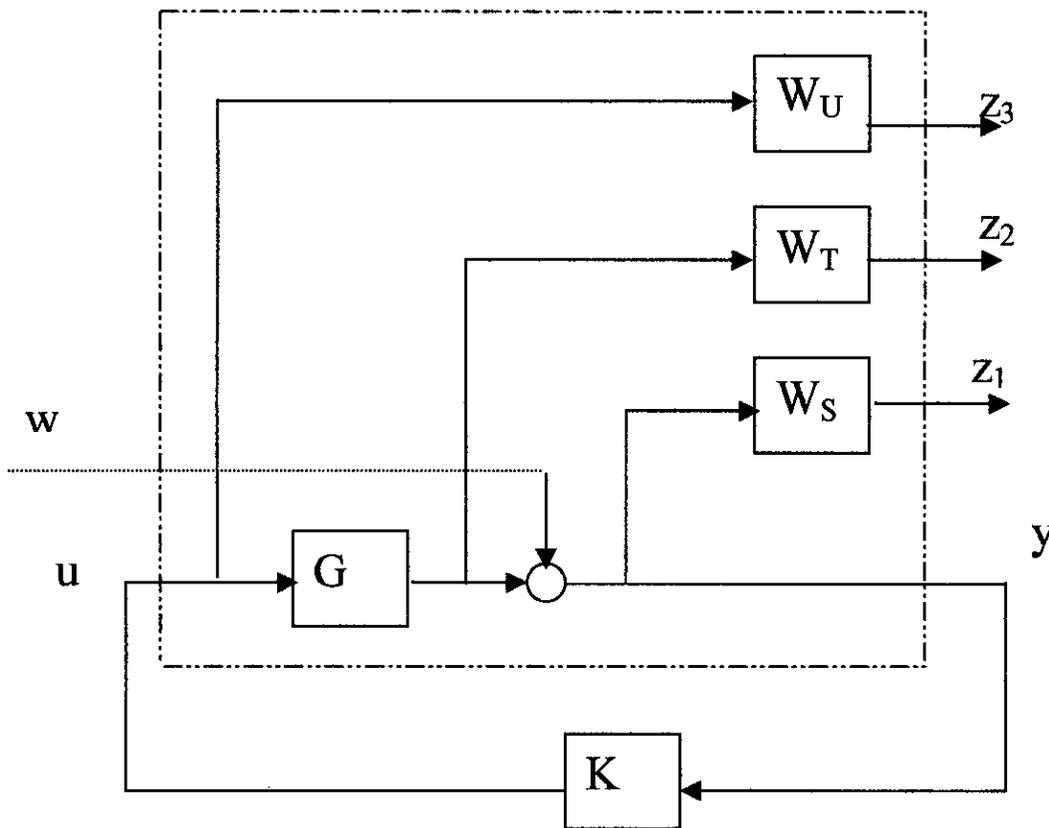
$$\Rightarrow \text{minimiser } \|W_T T\|_\infty$$

- Maximisation de la marge de stabilité additive

$$\Rightarrow \text{minimiser } \|W_U K S\|_\infty$$

On veut donc minimiser la norme $\left\| \begin{pmatrix} W_S S \\ W_T T \\ W_U K S \end{pmatrix} \right\|_\infty$

On peut représenter le problème standard équivalent par la figure(II.5) :



Figure(II. 5) forme générale du système pondéré

De la figure (II.5) on déduit les équations suivantes

$$y = -Gu + w \tag{II.25}$$

$$z_1 = W_S(-Gu + w) \tag{II.26}$$

$$z_2 = W_T(Gu) \tag{II.27}$$

$$z_3 = W_U u \tag{II.28}$$

ce qui, en écriture matricielle, donne :

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} W_S & -W_S G \\ 0 & W_T G \\ 0 & W_U \\ I & -G \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \quad (\text{II.29})$$

De cette dernière on en déduit que

$$T_{zw} = \begin{bmatrix} W_S S \\ W_T T \\ W_U K S \end{bmatrix} \quad (\text{II.30})$$

C'est ce dernier problème qui est appelé '*mixed sensitivity problem*'.

On peut avoir d'autres représentations du système augmenté, par exemple, au lieu d'utiliser une entrée exogène w et plusieurs z_i , on peut avoir w_1, w_2, w_3 , et un seul z .

En résumé :

La synthèse de la commande H_∞ standard passe par les étapes suivantes :

1. Traduire les objectifs $\| \cdot \|_\infty$.
2. Choisir les pondérations fréquentielles.
3. Trouver la matrice P du problème équivalent.
4. Résoudre le problème d'optimisation.
5. Tester les performances, si ce n'est pas bon alors revenir à l'étape 2.

La solution du problème standard passe par le choix des fonctions de pondération W_S, W_T et W_U . Puis la résolution du problème d'optimisation. On ne donnera pas la méthode de résolution de ce dernier, mais le lecteur pourra la consulter dans plusieurs ouvrages ou thèses [9] par exemple.

II.6 APPROCHE PAR FACTORISATION PREMIERE :

II.6.1 Définition :

Les deux matrices M et N propres et stables sont dites *facteurs coprime normalisés à gauche* de G si et seulement si :

$$G = M^{-1}N \quad (\text{II.31})$$

$$MM^* + NN^* = I \quad (\text{II.32})$$

Où M est une matrice inversible.

M et N étant des matrices de transfert propres et stables, sont dite coprimés à gauche si

$$\exists : U \text{ et } V \text{ propres et stables tels que } MV - NU = I$$

G étant défini par ces facteurs coprimés à gauche, on définit le système perturbé par l'ensemble suivant :

$$G_P = \{(M+\Delta_M)^{-1}(N+\Delta_N), (\Delta_N+\Delta_M) \in RH_\infty, \|\Delta_M, \Delta_N\|_\infty < \varepsilon\} \quad (\text{II.33})$$

On pose :

$$\varepsilon = \gamma^{-1}$$

RH_∞ est l'espace des fonctions de transfert stables et propres.

II.6.2 Position du problème :

A partir de la figure(II.2-b), nous avons les relations suivantes :

En boucle ouverte :

$$y = (M+\Delta_M)^{-1} (N+\Delta_N) \quad (\text{II.34})$$

donc
$$M y = -\Delta_M y + \Delta_N u + N u \quad (\text{II.35})$$

alors
$$M y = N u + [-\Delta_M, \Delta_N] \begin{bmatrix} y \\ u \end{bmatrix} \quad (\text{II.36})$$

$$M y = N u + \phi \quad (\text{II.37})$$

$$\phi = [-\Delta_M, \Delta_N] \begin{bmatrix} y \\ u \end{bmatrix} \quad (\text{II.38})$$

ϕ Représente l'incertitude du système.

En boucle fermée :

on peut montrer facilement que :

$$\begin{bmatrix} y \\ u \end{bmatrix} = \begin{bmatrix} K \\ I \end{bmatrix} (I-GK)^{-1} M^{-1} \phi = H \phi \quad (\text{II.39})$$

donc :

$$\phi = [-\Delta_M, \Delta_N] \begin{bmatrix} y \\ u \end{bmatrix} = \Delta \begin{bmatrix} y \\ u \end{bmatrix} \quad (\text{II.40})$$

le système est donc équivalent à la figure (II.7)

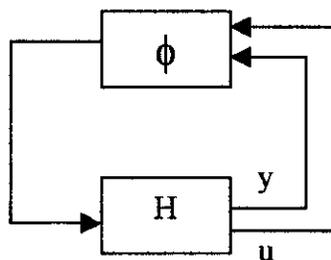


Figure (II.7) forme équivalente de décomposition en facteur coprime

Le théorème du petit gain nous assure que si le compensateur K vérifie

$$\|H\|_\infty \|\Delta\|_\infty < 1$$

Alors le système bouclé reste stable pour toute incertitude de norme infinie bornée.

En tenant compte de l'équation (II.33) le compensateur doit vérifier

$$\left\| \begin{bmatrix} K \\ I \end{bmatrix} [I - GK]^{-1} M^{-1} \right\|_\infty < \gamma \quad (\text{II.41})$$

Pour que le système perturbé représenté par G_p soit toujours stable.

Puisque $[M, N]$ est normalisée, alors $\|H\|_\infty = \|H[M, N]\|_\infty$

Et comme $M^{-1} [M \ N] = [G \ I]$

Le problème de la commande robuste optimale qui consiste à maximiser ϵ , par conséquent minimiser γ , ce qui revient à minimiser :

$$J = \left\| \begin{bmatrix} K \\ I \end{bmatrix} [I - GK]^{-1} [G \ I] \right\|_\infty \quad (\text{II.42})$$

L'avantage de la représentation par facteurs premiers est que le γ optimal est directement donné, sans les γ itérations (ce problème est rencontré dans la résolution du problème standard, nous ne l'avons pas exposé mais le lecteur pourra le trouver facilement dans plusieurs ouvrages [32] par exemple).

II.6.3 La solution :

- **Théorème [27] :**

Soit la représentation d'état du système G donnée par les matrices A, B, C et D , et soit X, Y les solutions définies positives données par les équations de Riccati suivantes :

$$A^T X + X A - X B B^T X + C^T C = 0 \quad (\text{II.43})$$

$$A Y + Y A^T - Y C^T C Y + B B^T = 0 \quad (\text{II.44})$$

Alors

$$\begin{cases} \gamma_{\min} = \sqrt{1 + \lambda_{\max}(XY)} \\ \epsilon_{\max} = \gamma_{\min}^{-1} \end{cases} \quad (\text{II.45})$$

Et pour tout $\epsilon < \epsilon_{\max}$ le compensateur qui stabilise G et vérifie $J < \gamma$, est donné par sa représentation d'état :

$$\begin{cases} \begin{bmatrix} A + B B^T + \epsilon^{-2} Z^{-1} Y C^T C & \epsilon^{-2} Z^{-1} Y C^T \\ B X^T & 0 \end{bmatrix} \\ Z = [I + X Y - \gamma^{-1} I]^{-1} \end{cases} \quad (\text{II.46})$$

- **Implémentation par loop-shaping :**

Dans la synthèse de sensibilité mixte le problème consistait à maintenir

$$\begin{cases} \overline{\sigma}(S) & \text{faible pour } \omega < \omega_B \\ \overline{\sigma}(T) & \text{faible pour } \omega > \omega_B \end{cases}$$

en utilisant le fait que :

$$\begin{aligned} T &= GK(I + GK)^{-1} \\ S &= (I + GK)^{-1} \end{aligned}$$

On remarque que :

$$\begin{aligned} \overline{\sigma}(S) & \text{faible pour } \underline{\sigma}(GK) \text{ grand} \\ \overline{\sigma}(T) & \text{faible pour } \underline{\sigma}(GK) \text{ faible} \end{aligned}$$

Ainsi on en déduit que, pour avoir une bonne poursuite, les valeurs singulières du système doivent être grandes en basses fréquences et faibles en hautes fréquences pour avoir un bon rejet de la perturbation.

Cependant une propriété remarquable du compensateur synthétisé par la méthode des facteurs coprimés, est que les valeurs singulières du système compensé sont très proches du système réel. Ce qui veut dire que le système corrigé avec le compensateur K assurant la stabilité robuste du système G a les mêmes performances que celles du système G tout seul.

Donc si on veut atteindre les performances fixées par le cahier de charge, les valeurs singulières de G doivent avoir des allures bien déterminées, ce qui n'est pas possible car on ne peut changer les paramètres du système à notre guise.

Pour y remédier, on insère un pré-compensateur ou filtre W de telle sorte que les valeurs de WG aient les caractéristiques désirées, ensuite synthétiser le correcteur K stabilisant le système WG .

- **Algorithme de Mc Farlane Glover [14] :**

- Choisir un pré-compensateur W de telle sorte que les valeurs singulières de WG aient les allures désirées.
- Calculer le ε_{max} pour WG , si sa valeur est trop faible, modifier le WG et répéter jusqu'à obtention du ε_{max} adéquat, en général sa valeur doit être dans l'intervalle $[0.3, 0.5]$. par suite synthétiser le correcteur K .
- Le régulateur final sera KW .

Remarques :

- Durant l'étape une, toutes les connaissances issues de l'automatique traditionnelle peuvent être utilisées.
- Pour calculer le ε_{max} et la synthèse du correcteur K on utilisera la commande matlab *ncfsyn*.

- Il est à noter que la valeur de ε_{max} est un indicateur de robustesse.

II.7 APPLICATION AU ROBOT PUMA 560 :

Après avoir donné un bref historique de la commande H_∞ , développé la théorie de la commande et donné la solution générale dans le cadre de l'implémentation par loop-shaping, on se propose à présent d'appliquer cette commande d'une manière décentralisée à un bras manipulateur pour la poursuite d'une trajectoire cycloïdale.

Pour ce faire, il nous faudra avoir un modèle représentatif du manipulateur. Or, comme nous le savons, le modèle de ce dernier est non linéaire et fortement interconnecté (annexe I). Pour contourner cet obstacle, nous aurions pu linéariser autour de plusieurs points de fonctionnement et synthétiser les contrôleurs. Cependant, reste le problème de la décentralisation. Dans les travaux déjà effectués [26], [27], on se contentait de négliger les termes d'interconnexion, ce qui est possible étant donné la robustesse de la commande vis à vis des incertitudes du modèle, mais une fois appliqué au modèle réel, le contrôleur perd sa robustesse vis à vis des variations paramétriques (l'effet de la charge portée), et supporte mal le test de décentralisation [27].

Alors nous avons décidé d'approximer le modèle réel à l'aide des systèmes flous, qui permettent de faire apparaître l'effet des couplages dans les paramètres des matrices A , B , C et D de l'approximateur à base de logique floue. Ceci permettra, comme nous le verrons plus tard, une meilleure robustesse vis à vis des variations de la masse et de la décentralisation.

II.7.1 Synthèse des systèmes flous :

Dans le chapitre I nous avons montré qu'on peut approximer notre système réel par trois modèles linéaires et indépendants, en n'importe quel point de fonctionnement. Il faut juste bien choisir les valeurs initiales des paramètres de Takagi-Sugeno et le gain de départ.

Sachant que notre manipulateur sera amené à suivre une trajectoire bien définie. On choisira donc, trois points de fonctionnement sur cette trajectoire. Nous nous sommes contentés de trois car c'est en régime dynamique qu'il faudra avoir une bonne approximation du système.

Les trois points de fonctionnements choisis sont donc,

$$q_1 = \frac{\pi}{12} \quad q_2 = \frac{3\pi}{12} \quad q_3 = \frac{5\pi}{12}$$

Pour éviter toute confusion, ces points ont été choisis pour chaque articulation et non un par articulation. On abouti finalement à neuf systèmes linéaires. Trois sous-systèmes par articulation.

Première articulation :

<i>Point de fonctionnement</i>	a_0	a_1	a_2	b
$\pi/12$	0.3302	0.1561	0.0386	0.0302
$3\pi/12$	1.0589	0.6209	1.0601	0.7589
$5\pi/12$	0.6482	0.3841	0.6008	0.7589

Tableau(II.1) valeurs des paramètres de Sugeno pour l'articulation 1

Deuxième articulation :

<i>Point de fonctionnement</i>	a_0	a_1	a_2	b
$\pi/12$	1.3706	0.4379	1.7346	1.0706
$3\pi/12$	0.7475	0.3956	0.9136	0.4475
$5\pi/12$	0.4565	0.1437	0.0555	0.1565

Tableau(II.2) valeurs des paramètres de Sugeno pour l'articulation 2

Troisième articulation :

<i>Point de fonctionnement</i>	a_0	a_1	a_2	b
$\pi/12$	1.8842	0.5728	3.2459	1.5842
$3\pi/12$	0.8760	0.5017	1.7100	0.5760
$5\pi/12$	0.5122	0.2156	0.2563	0.2122

Tableau(II.3) valeurs des paramètres de Sugeno pour l'articulation 3

II.7.2 Synthèse des correcteurs

Une fois les sous systèmes choisis, on commence tout d'abord par le choix des pondérations ou filtres. Ces derniers, combinés au système linéaire, doivent avoir des valeurs singulières élevées en basse fréquence afin d'assurer une bonne poursuite de la trajectoire et le rejet des perturbations, et très petites en hautes fréquences afin d'atténuer l'effet perturbateur des bruits de mesure.

Notre choix c'est donc fixé sur les pondérations suivantes :

Première articulation

$$W_1 = 400 \frac{s+10}{s+60} \quad W_2 = 400 \frac{s+6}{s+30} \quad W_3 = 400 \frac{s+10}{s+30}$$

Deuxième articulation

$$W_1 = 400 \frac{s+10}{s+30} \quad W_2 = 400 \frac{s+10}{s+50} \quad W_3 = 400 \frac{s+12}{s+50}$$

Troisième articulation

$$W_1 = 400 \frac{s+10}{s+50} \quad W_2 = 400 \frac{s+10}{s+50} \quad W_3 = 900 \frac{s+10}{s+50}$$

Les expressions des contrôleurs dans le plan de Laplace sont :

Première articulation

$$k_1 = \frac{-1.782s^2 - 113s + 355.3}{s^2 + 81.1s - 629.2} \quad k_2 = \frac{-1.82s^2 - 66s + 241.5}{s^2 + 78.98s + 436.1} \quad k_3 = \frac{-1.95s^2 - 75.04s - 399.4}{s^2 + 88.15s + 776.5}$$

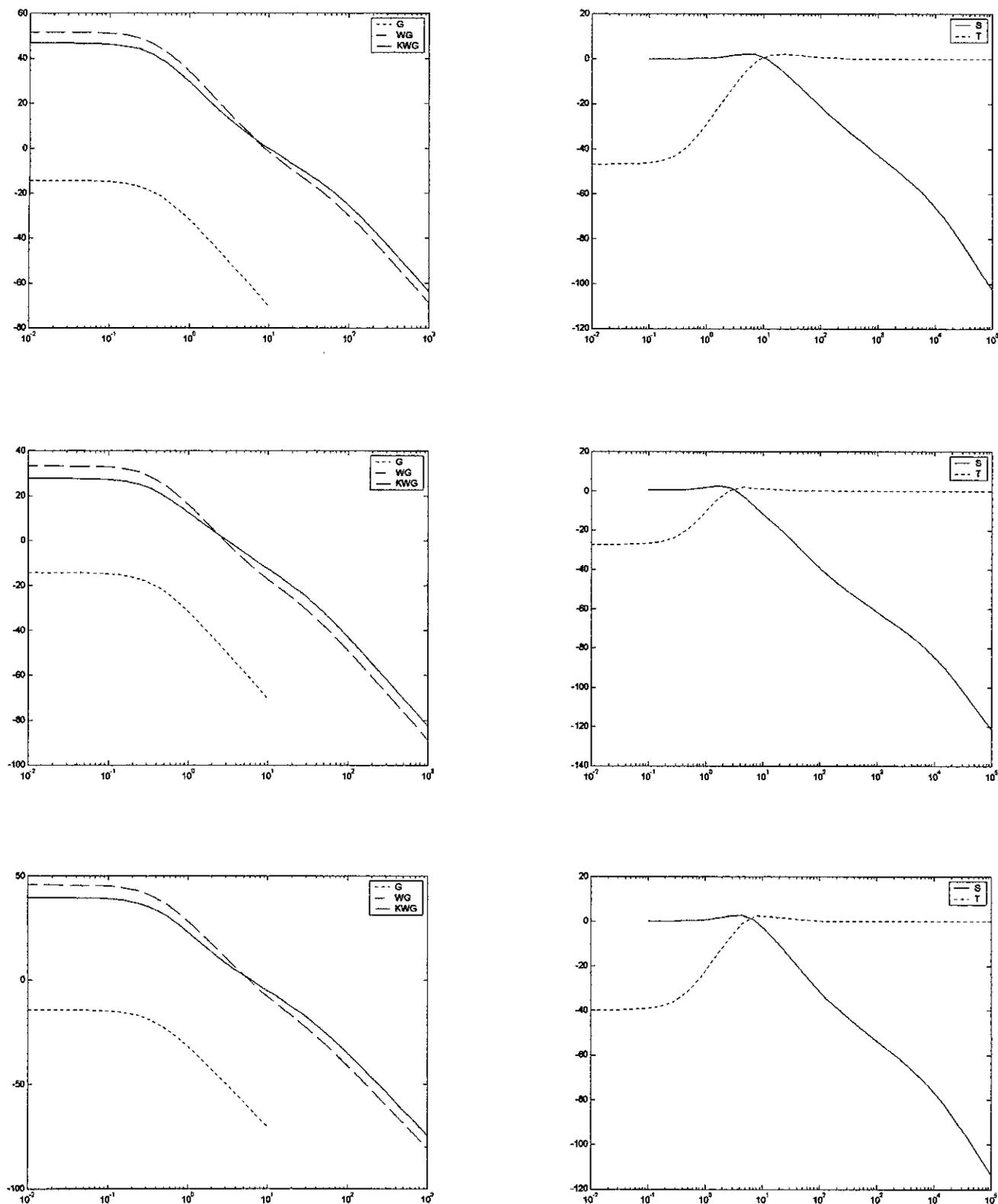
Deuxième articulation

$$k_1 = \frac{-2.017s^2 - 78.92s - 427.5}{s^2 + 96.35s + 859.9} \quad k_2 = \frac{-1.836s^2 - 99.57s - 365.2}{s^2 + 80.18s + 664.8} \quad k_3 = \frac{-1.949s^2 - 103.1s - 280.}{s^2 + 67.23s + 541.7}$$

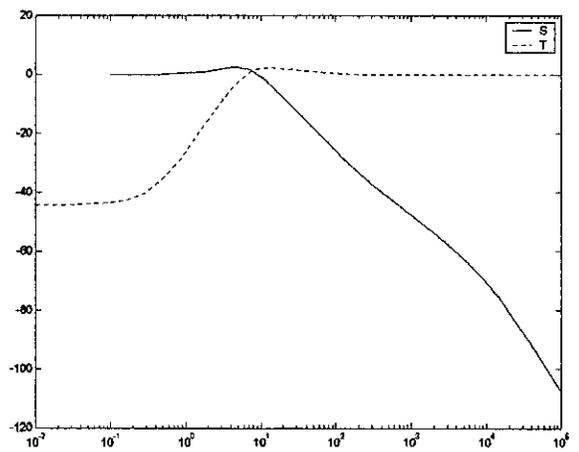
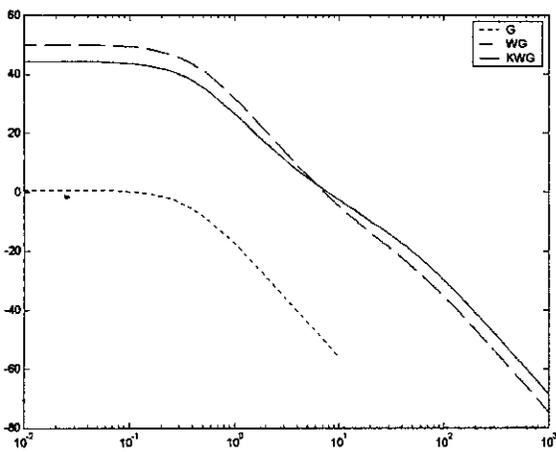
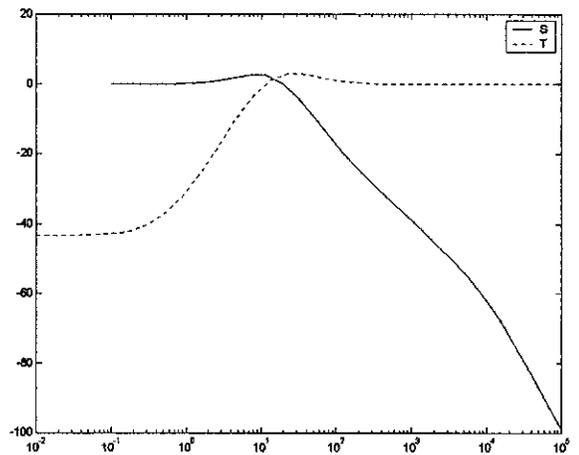
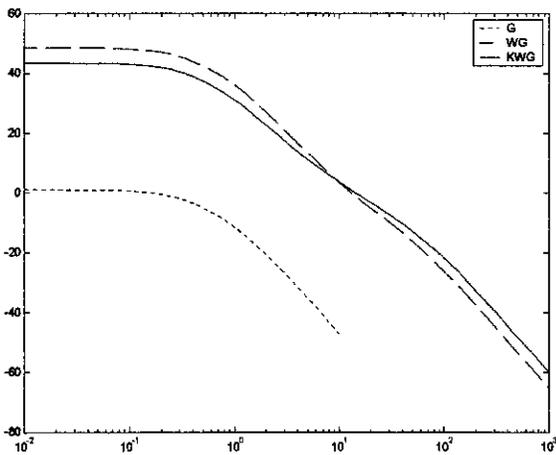
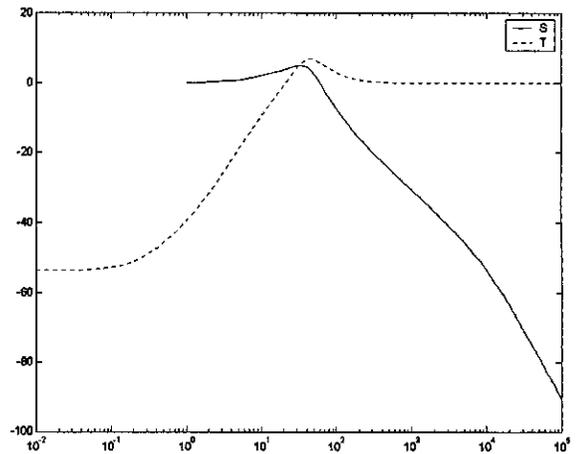
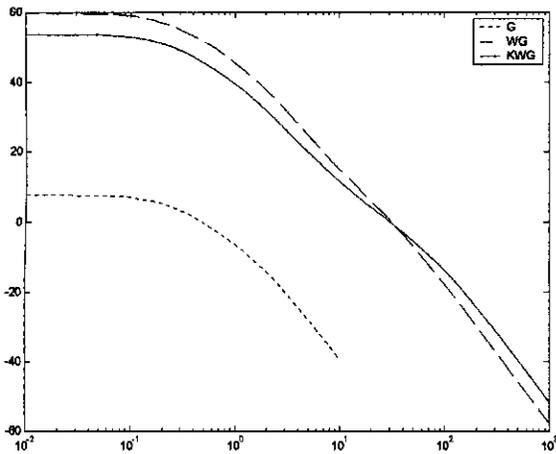
Troisième articulation

$$k_1 = \frac{-1.875s^2 - 108.9s - 577.7}{s^2 + 118.6s - 1079} \quad k_2 = \frac{-1.872s^2 - 102.1s - 393.3}{s^2 + 85.69s + 729.8} \quad k_3 = \frac{-1.767s^2 - 96.52s - 383.8}{s^2 + 81.08s + 675.1}$$

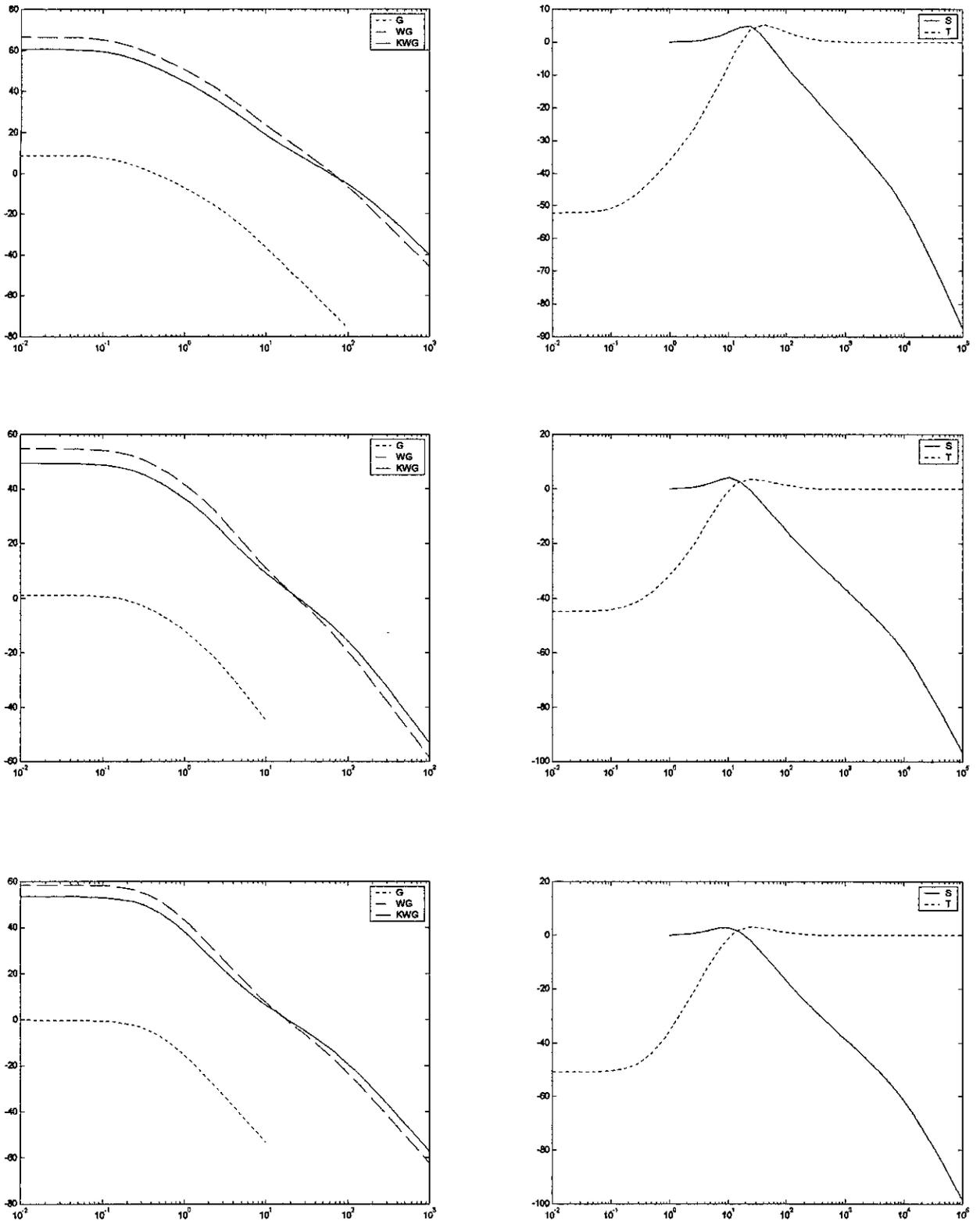
L'allure des valeurs singulières et fonctions sensibilité et sensibilité complémentaire est représentée sur les figures (II.8-9-10).



Figure(II.8) valeurs singulières et fonctions sensibilités pour la première articulation



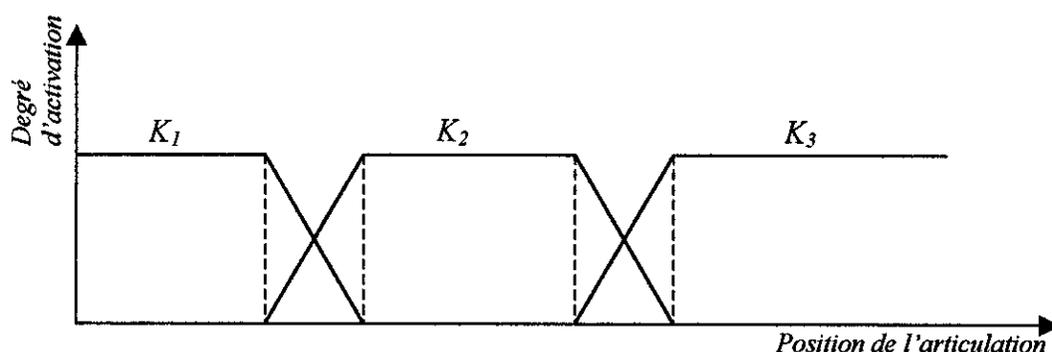
Figure(II.9) valeurs singulières et fonctions sensibilité pour la deuxième articulation



Figure(II.10) valeurs singulières et fonctions sensibilité pour la troisième articulation

Les figures précédentes montrent une caractéristique très intéressante du contrôleur H_∞ . On remarque bien que les valeurs singulières du système avec filtre ne diffèrent pas trop de celle du système avec filtre et compensateur. Les fonctions sensibilité et sensibilité complémentaire ont la forme désirée, ce qui nous assure une bonne poursuite des trajectoires en basses fréquences et un rejet des bruits de mesure en hautes fréquences.

Une fois les contrôleurs obtenus, il faudra les combiner de manière à ce que chaque contrôleur agit dans la plage où le système linéaire approximant le modèle réel est le plus valide. En d'autres termes chaque contrôleur se verra associer une 'fonction d'appartenance'. Cette structure de commande est appelée commande par réseaux de modèles locaux (*linear model network*) [27]. Les fonctions d'appartenances ont donc la forme représentée sur la figure(II.11).



Figure(II.11) stratégie de commutation entre les compensateurs

Ainsi on voit que chaque contrôleur sera prépondérant autour du point de fonctionnement autour duquel il a été synthétisé. On aura un meilleur contrôle et sur une plus grande plage de fonctionnement.

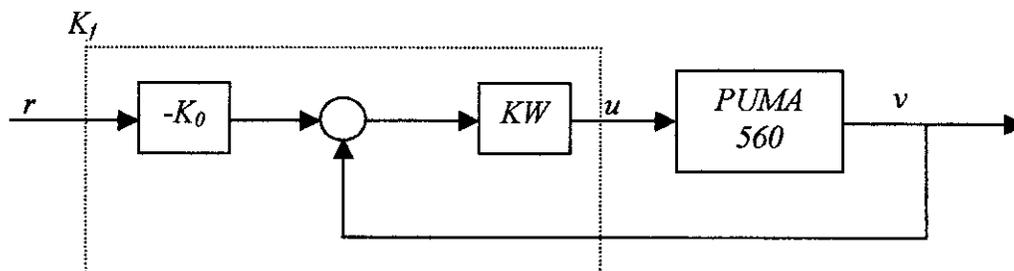
Le contrôleur final est donc :

$$K_f = \mu_1 K_1 + \mu_2 K_2 + \mu_3 K_3$$

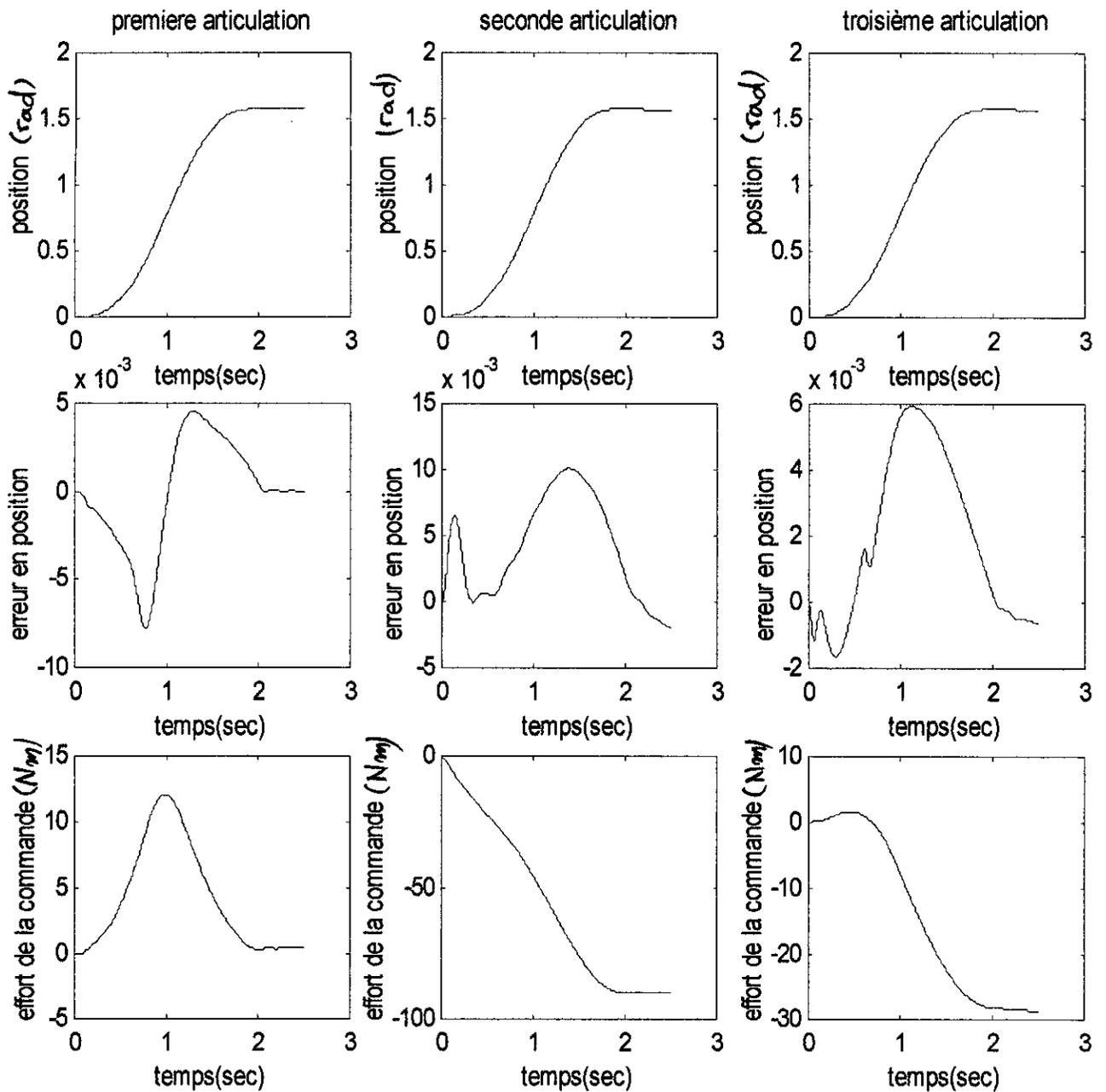
Où : μ_i est le degré d'activation ou de participation de chaque contrôleur.

II.7.3 Simulation non linéaire :

Après avoir déterminé le contrôleur décentralisé, il faut à présent l'appliquer sur le système non linéaire suivant le schéma de la figure(II.12), où K_0 est le gain statique de K_f .



Figure(II.12) schéma de commande en boucle fermée



Figure(II.13) réponses temporelles des articulations du manipulateur

Les simulations non linéaires, ont été effectuées en choisissant comme référence une trajectoire cycloïde pour chaque articulation :

$$q_d^i(t) = \begin{cases} q_d^i(0) + \frac{D_i}{2\pi} \left[2\pi \frac{t}{t_f} - \sin\left(2\pi \frac{t}{t_f}\right) \right] & \text{pour } t \leq t_f \\ q_d^i(t_f) & \text{pour } t \geq t_f \end{cases} \quad (\text{II.47})$$

où : $D_i = q_d^i(t_f) - q_d^i(0)$, t_f est le temps d'arrivée.

Notons que les trajectoires sont en (*rad*) et les commandes sont en (*Nm*).

La figure (II.13) montre le suivi des trajectoires cycloïdales par les trois articulations, les commandes sont lisses et admissibles, l'erreur est pratiquement négligeable même en régime dynamique.

II.8 CONCLUSION :

Dans ce chapitre nous avons exposé la notion de robustesse pour des incertitudes non structurées avec des conditions de stabilité. Nous avons vu qu'il était nécessaire de garantir une certaine robustesse a priori. La commande H_∞ traduit les objectifs de synthèse d'une manière très intuitive. Enfin nous avons vu que Mc Farlane et Glover ont bien posé le problème de stabilité robuste par facteurs coprimes normalisés. En plus, ils ont donné une technique de synthèse très efficace qui utilise le concept de loop-shaping. Un autre avantage de cette technique est qu'en monovariante, toutes les connaissances de base de l'automatique peuvent être utilisées pour le choix des pondérations. Cette technique de synthèse est ensuite appliquée pour astreindre un bras manipulateur à suivre une trajectoire de référence.

La poursuite est remarquable bien que l'effet de la masse soulevée n'était pas pris en compte lors de la modélisation par les systèmes flous. Ceci peut être considéré comme un premier test de robustesse. Nous effectuerons d'autres tests ultérieurement, afin de voir les limites de tolérance de cette commande et la comparer avec les autres commandes qui seront développées dans les chapitres suivants.

CHAPITRE III

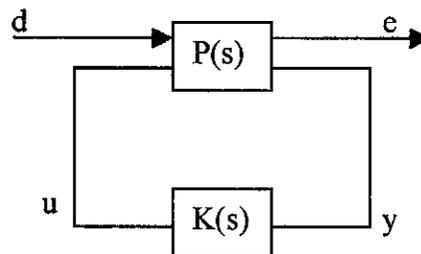
COMMANDE PAR RETOUR D'ETAT

Nous introduisons Dans ce chapitre la technique de commande par retour d'état d'abord dans le sens optimal (H_2) et puis un retour d'état permettant un placement des pôles. Comme La forme standard du problème a été déjà posée, on donnera directement les conditions d'existence d'un gain permettant de stabiliser le système et de satisfaire au performances exigées. Ce dernier sera appliqué au robot PUMA 560

III.1 INTRODUCTION :

Dans le cadre des systèmes linéaires invariants et propres où nous nous situons, le système est représenté par une matrice de transfert $G(s)$. Un certain nombre d'éléments perturbateurs liés à l'environnement du système physique et aux erreurs de modélisation peuvent être décrits à travers le signal d (figure(III.1)).

L'objectif de la compensation est d'assurer la stabilité du système en boucle fermée et un comportement satisfaisant en nominal. L'ensemble de ces qualités devra être préservé autant que possible en présence des perturbations externes.



Figure(III.1) forme standard du problème

Une grande classe de problèmes peut être ainsi formalisée. L'objectif consiste à trouver un compensateur K tel que la stabilité interne du système soit assurée et tel qu'une certaine norme du transfert noté $F_l(P, K)$ de d vers e , caractérisant des critères de performance et/ou de robustesse, soit minimisée.

III.2 POSITION DU PROBLEME :

Après avoir donné la forme standard du problème posé, on peut alors chercher des solutions de la forme suivante :

$$\min_{K \text{ stabilisant}} \|F_l(P, K)\|$$

Où

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$$

Les dimensions de chaque sous-matrice respectant les nombres d'entrées et de sorties du système. L'expression de $F_l(P, K)$ est alors donnée par :

$$e = F_l(P, K)d = (P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21})d \quad (\text{III.1})$$

La stabilisation du système P par le compensateur K est l'un des objectifs de la compensation. Cependant, P est un système multivariable, la notion de stabilité doit être précisée. C'est l'objectif du prochain paragraphe.

III.3 ETUDE DE LA STABILITE :

Pour un système multivariable, du fait d'éventuelles simplifications entre pôles et zéros, la stabilité n'est pas équivalente à celle des pôles de sa matrice de transfert. Deux types de stabilité sont alors étudiées : la stabilité interne et la stabilité externe.

La stabilité externe exprime la stabilité des modes de la matrice de transfert tandis que la stabilité interne exprime la stabilité des pôles de la matrice dynamique du système. Cependant il est possible d'exprimer, sous certaines conditions, les conditions de la stabilité interne en termes de fonctions de transfert [29].

Nous présentons une analyse de la stabilité interne dans le cas des systèmes caractérisés par une représentation d'état, pour l'utilisation des matrices de transfert le lecteur pourra se référer à [29] pour plus de détails.

La stabilité interne du système bouclé est liée aux valeurs propres de la matrice dynamique du système en boucle fermée. Nous allons donc calculer cette matrice. Soit la matrice d'état suivante :

$$P = \left(\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array} \right) \quad (\text{III.2})$$

on suppose que c 'est une représentation minimale du système P .

On dira alors que le compensateur K stabilise P ou de manière équivalente que la stabilité interne est assurée si le vecteur d'état associé à la représentation d'état du système en boucle fermée tend asymptotiquement vers zéro à partir d'une condition initiale quelconque.

Si l'on tient compte de la représentation d'état de P donnée en (III.2), celle de chaque sous système P_{ij} pour $i=1,2$ et $j=1,2$ est alors :

$$\left(\begin{array}{c|c} A & B_j \\ \hline C_i & D_{ij} \end{array} \right) \quad (\text{III.3})$$

soit maintenant

$$\left(\begin{array}{c|c} A & B_k \\ \hline C_k & D_k \end{array} \right) \quad (\text{III.4})$$

une représentation d'état minimale du compensateur K .

Nous noterons x le vecteur d'état associé au système P et x_k celui associé au compensateur K .

Pour calculer la matrice dynamique du système en boucle fermée, les entrées exogènes d sont annulées et le signal d'erreur e est ignoré. Les équations d'état qui régissent alors le comportement du système de la figure(III.1) sont les suivantes :

$$\begin{cases} \dot{x} = Ax + B_2 u \\ y = C_2 x + D_{22} u \end{cases} \quad (\text{III.5})$$

$$\begin{cases} \dot{x}_k = A_k x_k + B_k y \\ u = C_k x_k + D_k y \end{cases} \quad (\text{III.6})$$

Pour éliminer les variables u et y dans les systèmes d'équations (III.5) et (III.6), il est nécessaire de résoudre l'équation suivante :

$$\begin{pmatrix} I & -D_k \\ -D_{22} & I \end{pmatrix} \begin{pmatrix} u \\ y \end{pmatrix} = \begin{pmatrix} 0 & C_k \\ C_2 & 0 \end{pmatrix} \begin{pmatrix} x \\ x_k \end{pmatrix} \quad (\text{III.7})$$

Ceci nécessite l'inversion de la matrice

$$\begin{pmatrix} I & -D_k \\ -D_{22} & I \end{pmatrix} \quad (\text{III.8})$$

Donc pour pouvoir calculer la matrice dynamique du système en boucle fermée, la matrice (III.8) doit être inversible. C'est la raison pour laquelle, nous supposons que le problème représenté par la figure (III.1) est bien posé. Ou de manière équivalente que les trois propositions équivalentes suivantes sont vérifiées[29]:

i. La matrice $\begin{pmatrix} I & -D_k \\ -D_{22} & I \end{pmatrix}$ est inversible.

ii. La matrice $(I - D_k D_{22})$ est inversible.

iii. La matrice $(I - D_{22} D_k)$ est inversible.

Notons que le problème est bien posé lorsque la transmission directe P_{22} est nulle. L'expression de l'équation différentielle régissant la dynamique de l'état est donnée par :

$$\begin{aligned} \begin{pmatrix} \dot{x} \\ \dot{x}_k \end{pmatrix} &= \left[\begin{pmatrix} A & 0 \\ 0 & A_k \end{pmatrix} + \begin{pmatrix} B_2 & 0 \\ 0 & B_k \end{pmatrix} \begin{pmatrix} I & -D_k \\ -D_{22} & I \end{pmatrix}^{-1} \begin{pmatrix} 0 & C_k \\ C_2 & 0 \end{pmatrix} \right] \begin{pmatrix} x \\ x_k \end{pmatrix} \\ &= \bar{A} \begin{pmatrix} x \\ x_k \end{pmatrix} \end{aligned} \quad (\text{III.9})$$

La stabilité est alors équivalente à celle de \bar{A} . Celle ci dépend uniquement du sous système P_{22} ; la stabilité interne n'est liée qu'à P_{22} . on donne alors le théorème suivant :

• **Théorème[7] :**

Il existe un correcteur propre tel que la stabilité interne du système soit assurée si et seulement si (A, B_2) est commandable et (A, C_2) est détectable

L'existence d'un compensateur stabilisant n'est liée donc qu'à la stabilité et la détectabilité du sous système P_{22} . Dans la pratique, celui ci représente généralement le système à commander.

III.4 H_2 SYNTHÈSE DANS LE CAS D'UN RETOUR D'ÉTAT :

Dans ce paragraphe, nous considérons que les états sont mesurables. La représentation d'état relative au problème posé et déjà décrite en (III.2) étant prise en compte, les équations qui régissent alors le comportement dynamique se simplifient et deviennent :

$$\begin{cases} \dot{x} = Ax + B_1 d + B_2 u \\ e = C_1 x + D_{12} u \\ y = x \end{cases} \quad (\text{III.10})$$

d étant le vecteur des perturbations (consignes, bruit et incertitudes de modèle...), u le vecteur des entrées, y le vecteur des sorties mesurées et e l'erreur à minimiser.

Compte tenu de la mesurabilité de l'état, la représentation d'état du système P s'écrit :

$$P = \left(\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ I & 0 & 0 \end{array} \right) \quad (\text{III.11})$$

Le problème d'optimisation est donc de trouver un compensateur stabilisant et minimisant le critère suivant :

$$\min_K \|F_1(P, K)\|_2$$

La validité des expressions des correcteurs optimaux n'est garantie que sous les hypothèses suivantes [29]:

- i. (C_1, A) est observable
- ii. (A, B_2) est commandable
- iii. $D_{12}^T [C_1 \ D_{12}] = [0 \ I]$

Ces hypothèses ne sont pas restrictives mais permettent de simplifier la représentation d'état des correcteurs optimaux. Le problème étant posé dans sa forme standard, nous sommes maintenant en mesure de présenter les principaux résultats relatifs à la synthèse H_2 .

Pour de plus amples informations sur le calcul de la norme H_2 d'une matrice, le lecteur se refera à l'annexe II.

- **Synthèse d'un retour d'état H_2 :**

Classiquement, le régulateur LQ (Linéaire Quadratique) est optimale et nécessite la résolution d'une équation de Riccati standard dont l'Hamiltonien est donné par :

$$H_2 = \begin{bmatrix} A & -B_2 B_2^T \\ -C_1 C_1^T & -A^T \end{bmatrix} \quad (\text{III.12})$$

ce qui donne l'équation de Riccati suivante :

$$A^T X + XA - X B_2 B_2^T X + C_1 C_1^T = 0 \quad (\text{III.13})$$

D'après les hypothèses faites plus haut, la solution de cette équation existe et est unique[7]. De plus elle est semi-définie positive. Le retour d'état optimal est donné par :

$$K = -B_2^T X_2 \quad (\text{III.14})$$

Où X_2 est la solution de l'équation de Riccati dont le Hamiltonien est donné par l'expression (III.12).

D'où l'on déduit la matrice dynamique en boucle fermée

$$A_K = A + B_2 K \quad (\text{III.15})$$

Ainsi la nouvelle matrice associée au signal d'erreur

$$C_{1K} = C_1 + D_{12} K \quad (\text{III.16})$$

Considérons le système dont la représentation d'état est la suivante :

$$G_C = \left(\begin{array}{c|c} A_K & B_2 \\ \hline C_{1K} & 0 \end{array} \right) \quad (\text{III.17})$$

La valeur optimale du critère est alors donnée par :

$$\min_K \|F_1(P, K)\|_2 = \|G_C\|_2 \quad (\text{III.18})$$

Il apparaît que cette technique de résolution, bien connue par les automaticiens, est très simple car elle nécessite la résolution d'une seule équation de Riccati.

III.5 H₂ SYNTHÈSE DANS LE CAS D'UN RETOUR DE SORTIE :

Dans ce paragraphe nous présentons la synthèse H_2 dans le cas où le signal d'entrée y du compensateur est une combinaison linéaire du vecteur d'état et de la perturbation :

$$y = C_2 x + D_{21} d \quad (\text{III.19})$$

La forme standard relative à ce problème est donnée par la figure (III.1) et le représentation de la matrice P par l'équation (III.2).

Les hypothèses suivantes sont supposées vérifiées (ces hypothèses ne sont pas restrictives, elles permettent d'assurer l'existence du compensateur et la simplification de ses expressions) :

- i. (A, B_2) commandable et (C_2, A) observable
- ii. (A, B_1) commandable et (C_1, A) observable
- iii. $D_{12}^T [C_1 \ D_{12}] = [0 \ I]$
- iv. $\begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} D_{21}^T = \begin{bmatrix} 0 \\ I \end{bmatrix}$

• **Synthèse du compensateur :**

Dans ce cadre le principe de séparation (commande - estimation) est vérifié. On a donc d'une part, un retour d'état comme nous l'avons décrit précédemment et d'autre part un estimateur dont le gain est obtenu par la résolution d'une équation de Riccati J_2 duale de celle intervenant dans le problème de commande :

$$J_2 = \begin{bmatrix} A^T & -C_2^T C_2 \\ -B_1 B_1^T & -A \end{bmatrix} \quad (\text{III.20})$$

La matrice de gain d'estimation est alors donnée par :

$$L_2 = -Y_2 C_2^T \quad (\text{III.21})$$

Ainsi, la matrice dynamique du système en boucle fermée est :

$$A_{L_2} = A + L_2 D_{21} \quad (\text{III.22})$$

Et la nouvelle matrice de perturbation est donnée par :

$$B_{1L_2} = B_2 + L_2 D_{21} \quad (\text{III.23})$$

Considérons le système G_f dont une représentation d'état est donnée par :

$$G_f = \left(\begin{array}{c|c} A_{L_2} & B_{1L_2} \\ \hline F_2 & 0 \end{array} \right) \quad (\text{III.24})$$

L'unique compensateur optimal K_{opt} a pour représentation d'état :

$$K_{opt} = \left(\begin{array}{c|c} A + B_2 F_2 + L_2 C_2 & -L_2 \\ \hline F_2 & 0 \end{array} \right) \quad (\text{III.25})$$

Et la valeur optimale du critère est donnée par :

$$\min_K \|F_l(P, K)\|_2 = \sqrt{\|G_c\|_2^2 + \|G_f\|_2^2} \quad (\text{III.26})$$

Ainsi, la synthèse d'un compensateur H_2 dans le cas d'un retour de sortie est directe et nécessite la résolution de deux équations de Riccati standards. La valeur optimale est atteinte et c'est le compensateur correspondant, noté K_{opt} qui est calculé.

Sa représentation d'état peut être réécrite sous la forme suivante :

$$\begin{cases} \dot{\hat{x}} = A \hat{x} + B_2 u + L_2 (C_2 \hat{x} - y) \\ u = F_2 \hat{x} \end{cases} \quad (\text{III.27})$$

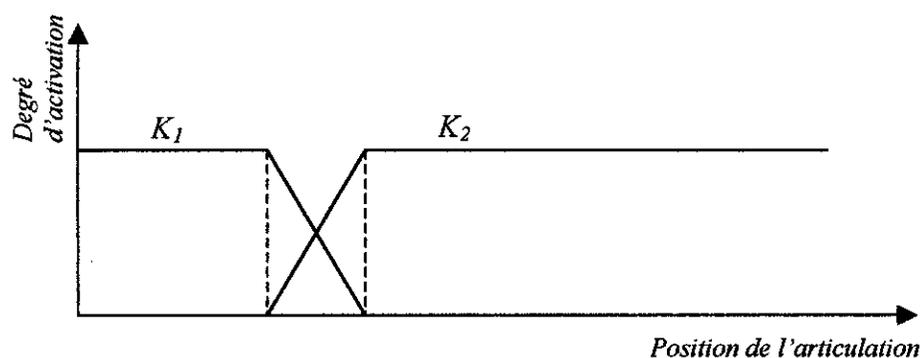
Cette représentation met en évidence le principe de séparation commande estimation. Où L_2 est le gain optimal de l'observateur, F_2 la matrice de gain appliquée à l'état estimé \hat{x} .

III.6 APPLICATION AU ROBOT PUMA 560 :

Après avoir donné les notions théoriques nécessaires pour la synthèse d'un retour d'état et d'un retour de sortie optimal au sens H_2 , on s'apprête à présent à synthétiser les gains de retour d'état. Pour cela, comme dans le chapitre précédent, on utilisera les systèmes flous comme modèles représentatifs de notre système réel. Ceci nous permettra aussi d'éviter le recours à des observateurs, car les systèmes flous nous fournissent l'état de chaque sous système à tout instant.

Comme pour la commande H_∞ , on utilisera une stratégie de commutation entre les différents gains de retour d'état. La seule différence est que dans ce chapitre, on se limitera uniquement à deux sous systèmes par articulation. Ce choix a été imposé afin d'avoir une forme plus lisse de la commande et éviter les pics qui sont généralement dus aux commutations. Néanmoins, ces comparaisons feront l'objet du prochain chapitre, où l'on effectuera une étude comparative entre les différentes méthodes.

Ainsi la stratégie de commutation est telle que représentée sur la figure(III.2).



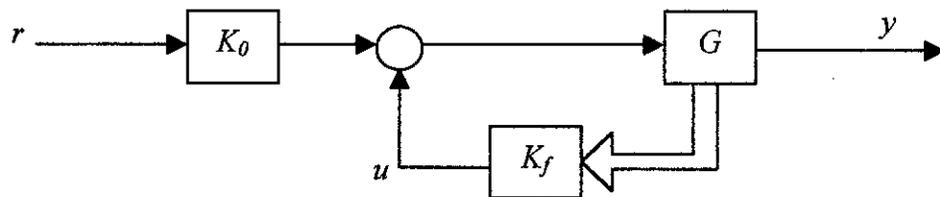
Figure(III.2) stratégie de commutation

Le gain du retour d'état appliqué au modèle non linéaire sera donc :

$$K_f = \mu_{u_1} K_1 + \mu_{u_2} K_2$$

Où μ_{u_i} représente le degré de contribution de chaque gain de retour.

La boucle de commande est représentée sur la figure (III.3).



Figure(III.3) boucle de commande

La pondération d'entrée K_0 est calculée de telle sorte qu'en régime statique le gain entre la consigne et la grandeur de sortie soit unitaire. Comme la représentation d'état du système de la figure (III.3) est donnée par :

$$\begin{cases} \dot{x} = Ax + Be \\ e = u + K_0 r \\ u = K_f x \\ y = Cx \end{cases} \quad (\text{III.28})$$

La fonction de transfert en boucle ouverte est donc égale à :

$$Y(s) = C(sI - A - BK_f)^{-1} BK_0 R(s) \quad (\text{III.29})$$

Comme en régime statique la consigne est un échelon, et sachant que :

$$y(\infty) = \lim_{s \rightarrow 0} sY(s) \quad (\text{III.30})$$

Alors le gain de transfert entre l'entrée et la sortie est donc :

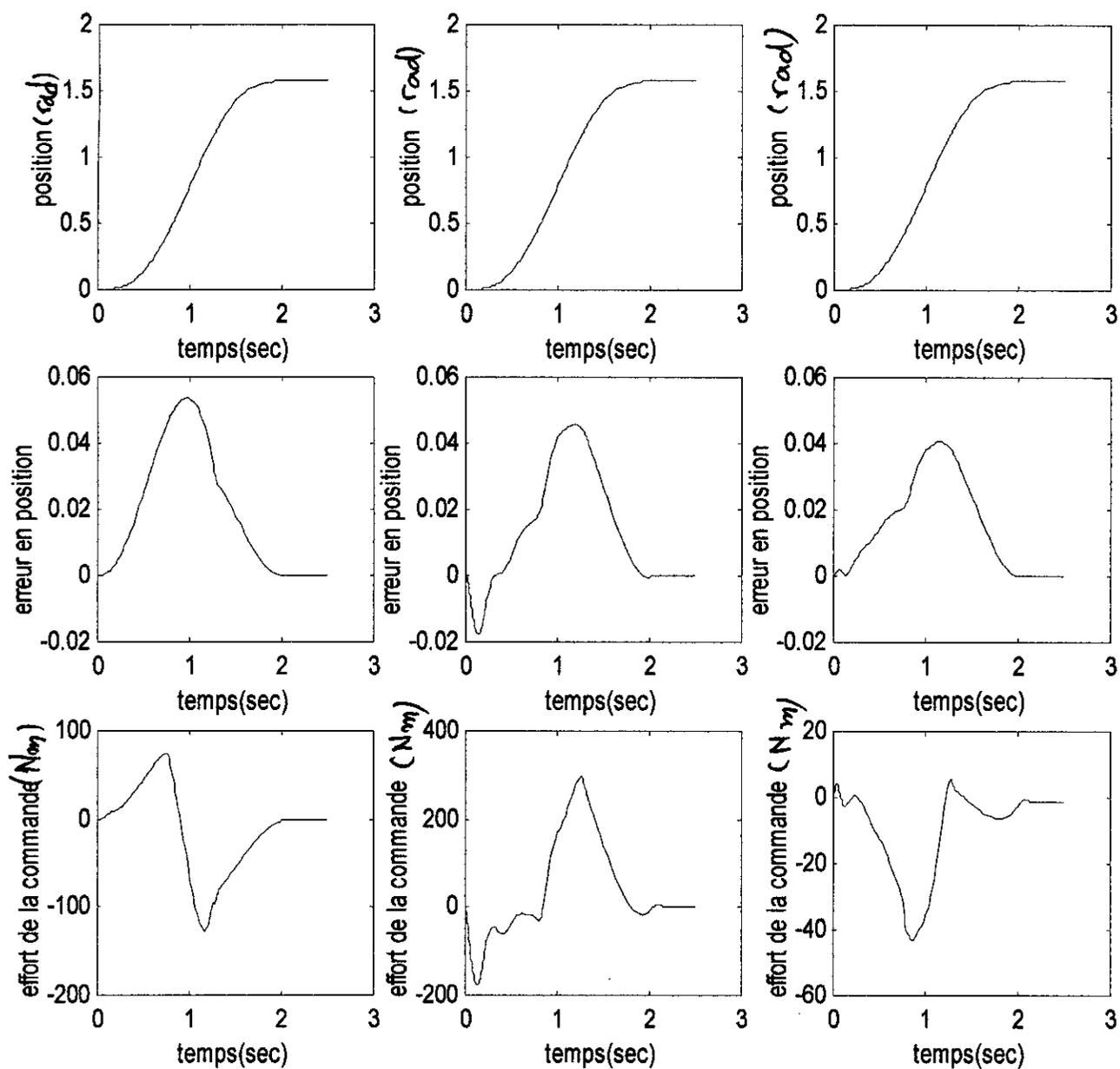
$$\lim_{s \rightarrow 0} C(sI - A - BK_f)^{-1} BK_0 = C(-A - BK_f)^{-1} BK_0$$

Ce dernier doit être unitaire. D'où l'on déduit la valeur de la pondération de l'entrée :

$$K_0 = \frac{1}{[C(-A - BK_f)^{-1} B]} \quad (\text{III.31})$$

Où les matrices A , B , C sont données par le système flou.

La trajectoire imposée (consigne) est toujours la trajectoire cycloïdale et est donnée par l'équation (II.56).



Figure(III.4) Réponses temporelles des trois articulations du robot PUMA560

La figure (III.4) montre les simulations non linéaires des trois articulation du robot PUMA 560. On constate que les trajectoires sont bien suivies, les commandes sont admissibles et l'erreur est relativement négligeable. Il est à noter que les systèmes flous utilisés pour le calcul des gains sont les mêmes utilisés pour la synthèse H_∞ . Comme ces derniers sont l'approximation du système réel lorsqu'il est à vide, on apprécie donc la robustesse de la commande face aux erreurs de modélisations dues à la fois à l'approximation par les systèmes flous et l'effet de la charge qui a été négligé lors du calcul des paramètres des systèmes flous.

III.7 PLACEMENT DE POLES PAR RETOUR D'ETAT :

Dans ce paragraphe, nous abordons la deuxième partie de notre chapitre. Il s'agit à présent de trouver le retour d'état permettant d'effectuer un placement des pôles et donc imposer au système de suivre une certaine dynamique.

En premier lieu, nous donnons les notions théoriques nécessaires pour la synthèse d'un tel retour d'état. On présente d'abord les relations de bases, où l'on mentionne les principes adoptés pour le dimensionnement du gain de retour d'état. Néanmoins, plusieurs similitudes existent entre les deux méthodes de commande car elles sont toutes les deux basées sur un retour d'état. Nous utilisons donc la même stratégie de commutation, la même pondération d'entrée et bien sure les mêmes systèmes flous.

Le principe du placement de pôles est très simple, il s'agit de trouver un gain K permettant d'avoir une fonction de transfert en boucle fermée bien déterminée.

Soit la représentation d'état du système en boucle ouverte :

$$\begin{cases} \dot{x} = Ax + Be \\ y = Cx \end{cases} \quad (\text{III.32})$$

Et soit le retour d'état suivant :

$$u = Kx \quad (\text{III.33})$$

L'entrée appliquée au système est l'erreur définie par :

$$e = K_0 r + u = K_0 r + Kx \quad (\text{III.34})$$

La fonction de transfert en boucle fermée est donc :

$$F(s) = C(sI - A - BK)^{-1} BK_0 \quad (\text{III.35})$$

On désire que cette dernière ait une dynamique bien déterminée. Pour ce, nous choisirons le gain K afin d'imposer les pôles de la fonction de transfert en boucle fermée.

Comme nous sommes en monovariante et connaissant la forme générale des matrices A , B , C on peut écrire :

$$A = \begin{pmatrix} 0 & 1 \\ a_1 & a_2 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ b \end{pmatrix} \quad C = (1 \quad 0)$$

et comme $K = (k_1 \quad k_2)$ l'équation (III.35) s'écrit :

$$F(s) = \frac{bK_0}{s^2 - (a_2 + bk_2)s - (a_1 + bk_1)} \quad (\text{III.36})$$

Ainsi imposer une dynamique à une articulation donnée revient à imposer le dénominateur de la fonction de transfert en boucle fermée.

Généralement la fonction de transfert désirée est de la forme :

$$F_d(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (\text{III.37})$$

Pourquoi cette forme ? Car un système du deuxième ordre est un système qu'on connaît très bien et qu'on sait très bien réguler.

Par simple identification entre les équations (III.36) et (III.37) on déduit les expressions des gain du retour d'état :

$$\begin{cases} k_1 = -\frac{\omega_n^2 + a_1}{b} \\ k_2 = -\frac{2\xi\omega_n + a_2}{b} \\ K_0 = \frac{\omega_n^2}{b} \end{cases} \quad (\text{III.38})$$

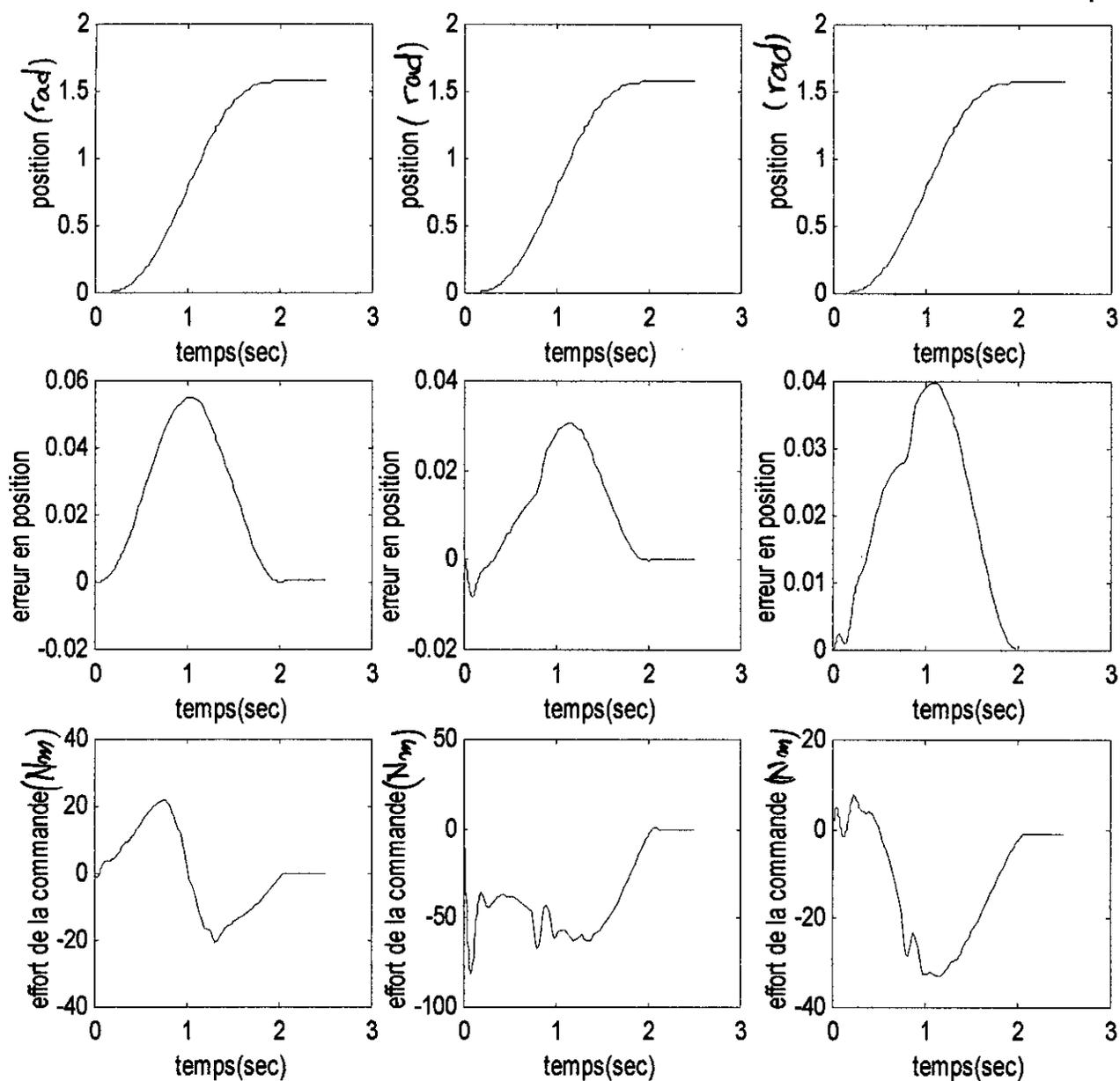
Il suffit donc d'imposer un coefficient d'amortissement ξ et une fréquence de coupure ω_n pour avoir la dynamique recherchée.

Pour notre application nous avons choisi un même coefficient d'amortissement pour les trois articulations égal à 0.7, pour la fréquence de coupure elle varie d'une articulation à une autre. Nous avons même pensé à choisir une fréquence de coupure différente pour chaque sous système de chaque articulation mais cela n'avait pas de grande influence sur l'erreur en régime dynamique qui est le principale critère de performance. Alors nous nous contenterons d'une même fréquence de coupure pour tous les sous systèmes de chaque articulation.

Finalement les résultats les plus satisfaisants ont été atteints avec les fréquences de coupures suivantes :

$$\omega_{n1} = 40 \text{ rad s}^{-1} \quad \omega_{n2} = 50 \text{ rad s}^{-1} \quad \omega_{n3} = 60 \text{ rad s}^{-1}$$

La figure(III.5) donne les différentes réponses temporelles du système réel à une entrée cycloïdale. On constate le suivi parfait des trajectoires, l'erreur en régime dynamique est presque négligeable et les commandes sont assez lisses.



Figure(III.5) Réponses temporelles des trois articulations du robot PUMA560

III.8 CONCLUSION :

Dans ce chapitre, nous avons commencé par introduire l'idée de la commande par retour d'état. Nous avons d'abord mis le problème sous forme standard. Puis nous avons vu que le gain recherché appartient à un ensemble de compensateurs stabilisants le système. Celui qu'on allait choisir est le gain qui permet en plus d'atteindre un certain niveau de performances. On a vu aussi que le régulateur H_2 permet de minimiser l'énergie de transfert entre les signaux perturbateurs et le signal d'erreur. Ceci nous assure un comportement satisfaisant même en présence de perturbations externes et/ou d'erreurs de modélisation. De plus le calcul du régulateur se fait en une seule étape, et le résultat est optimal.

Dans la deuxième partie, nous avons présenté le placement des pôles par retour d'état. Cette technique permet d'imposer une dynamique au système à commander. Elle présente plusieurs similitudes avec la commande H_2 , dans le sens où l'on utilise un retour d'état pour stabiliser et commander le système, ainsi que dans le choix de la pondération de l'entrée qui nous assure un gain unitaire entre l'entrée et la sortie en régime permanent, et ceci afin d'annuler l'erreur statique. Mais le placement des pôles n'est pas systématique, il appartient donc à l'utilisateur de choisir le placement qui le juge satisfaisant. Donc le gain n'est plus calculé d'une manière unique contrairement au retour d'état synthétisé par la technique H_2 .

On remarque aussi, que contrairement à la commande H_∞ , la commande par retour d'état n'utilise que deux modèles flous en régime dynamique au lieu de trois. Nous donnerons plus de détails sur ce choix dans le chapitre suivant où l'on effectuera plusieurs tests de robustesse et l'on discutera sur le choix des sous systèmes.

CHAPITRE IV

TESTS DE ROBUSTESSE

Dans ce chapitre, nous appliquons différents tests de robustesse aux commandes synthétisées dans les chapitres précédents. Ceci nous permettra de comparer les commandes et d'apprécier leurs tolérances face à des variations paramétriques (charge soulevée, valeurs des paramètres du manipulateur... etc.), à des bruits de mesure, à des ruptures de commandes et à la poursuite d'une trajectoire qui varie tout le temps. Des conclusion seront ainsi tirées à la fin de chaque test.

IV.1 INTRODUCTION :

Nous allons à présent soumettre notre système commandé à différents tests de robustesse, et ceci afin de pouvoir conclure et comparer les différents contrôleurs calculés dans les chapitres précédents. Pour ce faire, nous avons choisis plusieurs tests qui sont en fait des cas souvent rencontrés dans la réalité tels que les bruits de mesure, les variations paramétriques et les ruptures de commandes. Nous avons aussi effectué un autre test, celui de l'actionneur. C'est à dire que la commande ne sera pas directement appliquée à notre système. Cette dernière passera d'abord par un actionneur ayant une certaine dynamique, à priori on s'attend à ce que les commandes soient plus lisses. Donc tout au long de ce chapitre nous essayerons d'approcher le système tel qu'il est en réalité. Nos conclusions suivent chaque test effectué. On rappelle que la position est donnée en radian et que la commande est en newton.mètre.

IV.2 TEST DE LHEAVY :

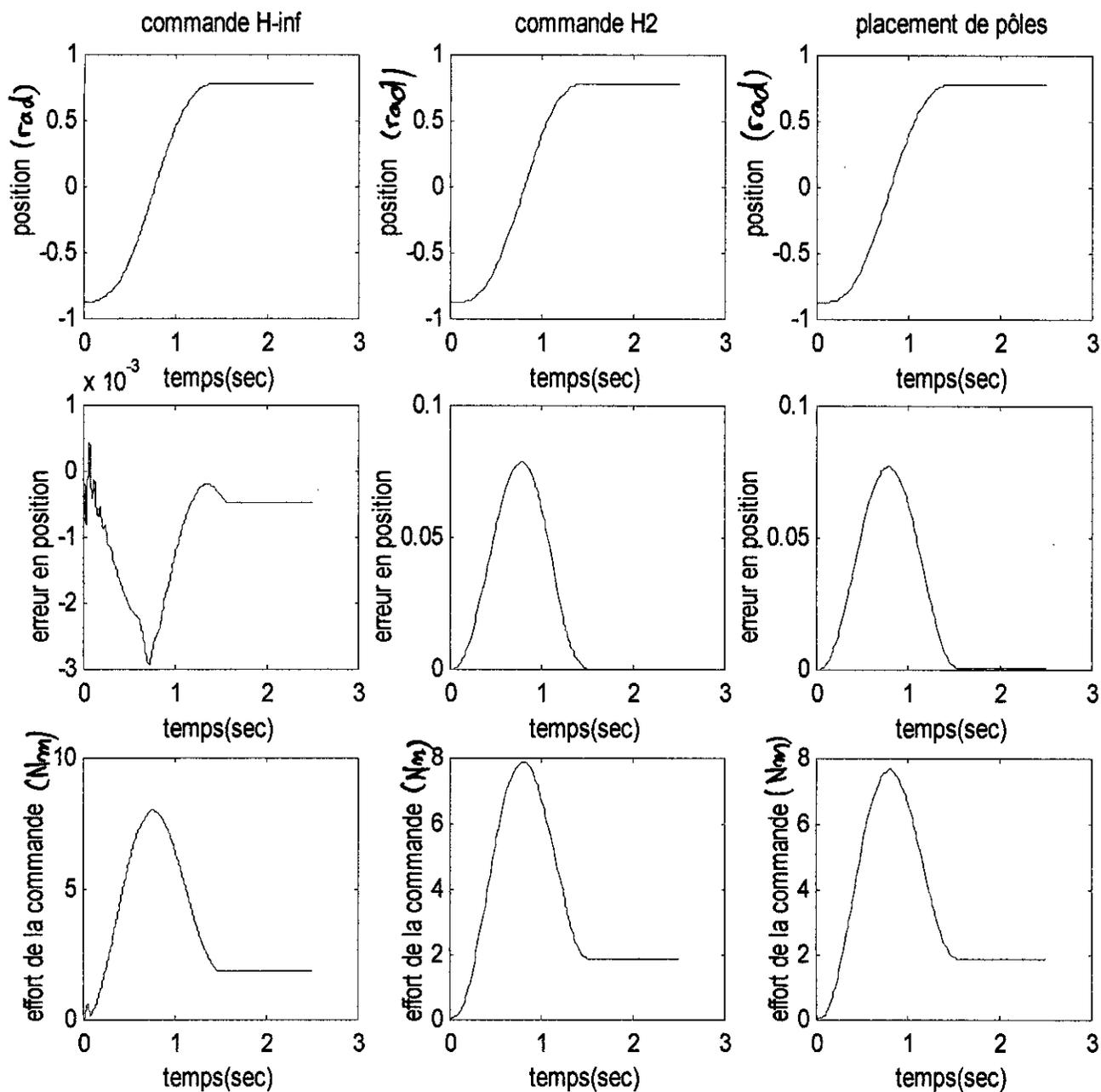
Ce test consiste à changer la consigne appliquée au manipulateur. Il s'agit de changer les points de départ et d'arrivée de chaque articulation, la trajectoire est toujours cycloïdale. Cette trajectoire est choisie car elle excite toutes les dynamiques du bras manipulateur [26].

Les points de départ de chaque articulation sont :

$-5\pi/18$ pour la première, $-3\pi/4$ pour la deuxième et $3\pi/4$ pour la troisième. Les points d'arrivée sont, respectivement, $\pi/4$, $-17\pi/36$ et $\pi/6$. Les articulations du robot doivent se déplacer d'un point à un autre en un temps $t=1.5$ s.

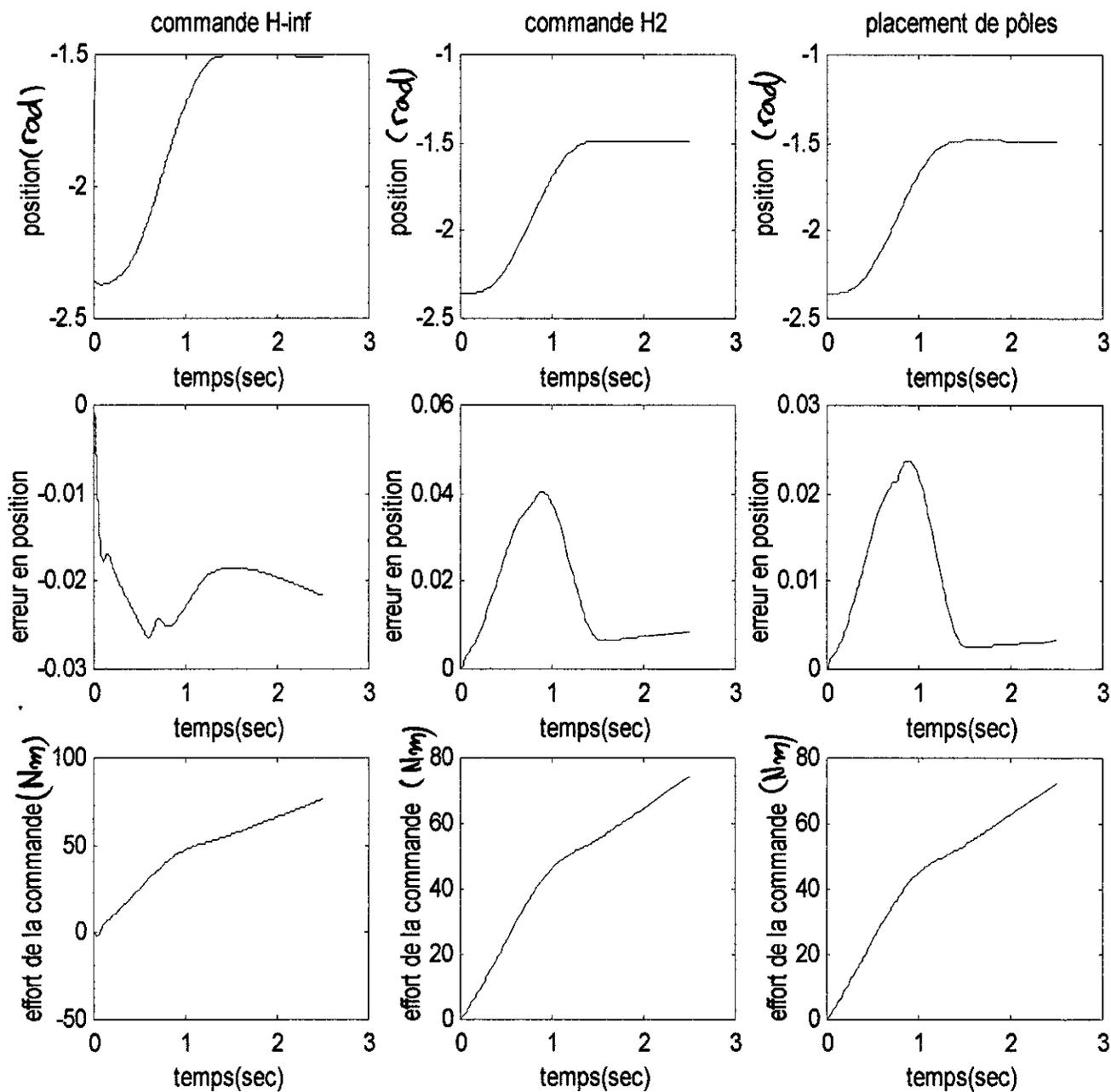
Les figures (IV.1-3) montrent les réponses temporelles des trois articulations.

Première articulation :



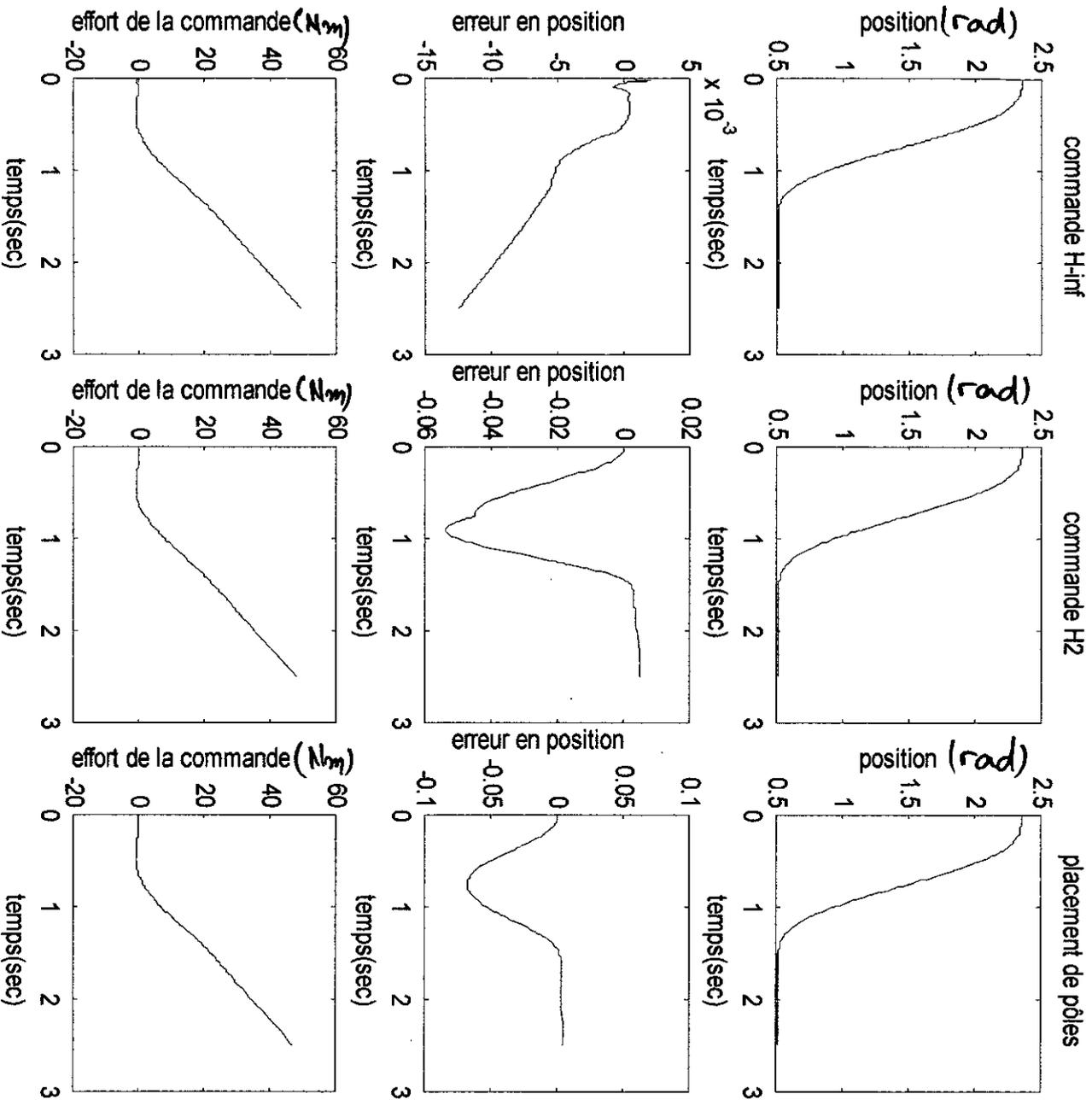
Figure(IV.1) test de Lheavy pour la première articulation

Deuxième articulation :



Figure(IV.2) test de Lheavy pour la deuxième articulation

Troisième articulation :



Figure(IV.3) test de Lheavy pour la troisième articulation

IV.2.1 Conclusion :

Dans ce test la référence a été changée de manière à exciter toutes les dynamiques du manipulateur. Nous constatons une bonne poursuite des trajectoire de références. Néanmoins, l'erreur dynamique est nettement plus importantes que celle obtenue avec une trajectoire 'ordinaire', l'effort développé par les contrôleur est aussi assez important. Ceci s'explique par deux faits :

- La trajectoire de référence est étudiée pour pousser les contrôleur a fond[26].
- Les points de fonctionnements sur lesquels on s'est basés pour approximer le modèle du robot par les systèmes flous sont hors des trajectoires de référence. Donc le modèle utilisé pour la synthèse des contrôleurs est moins fiable, du moins pour cette référence.

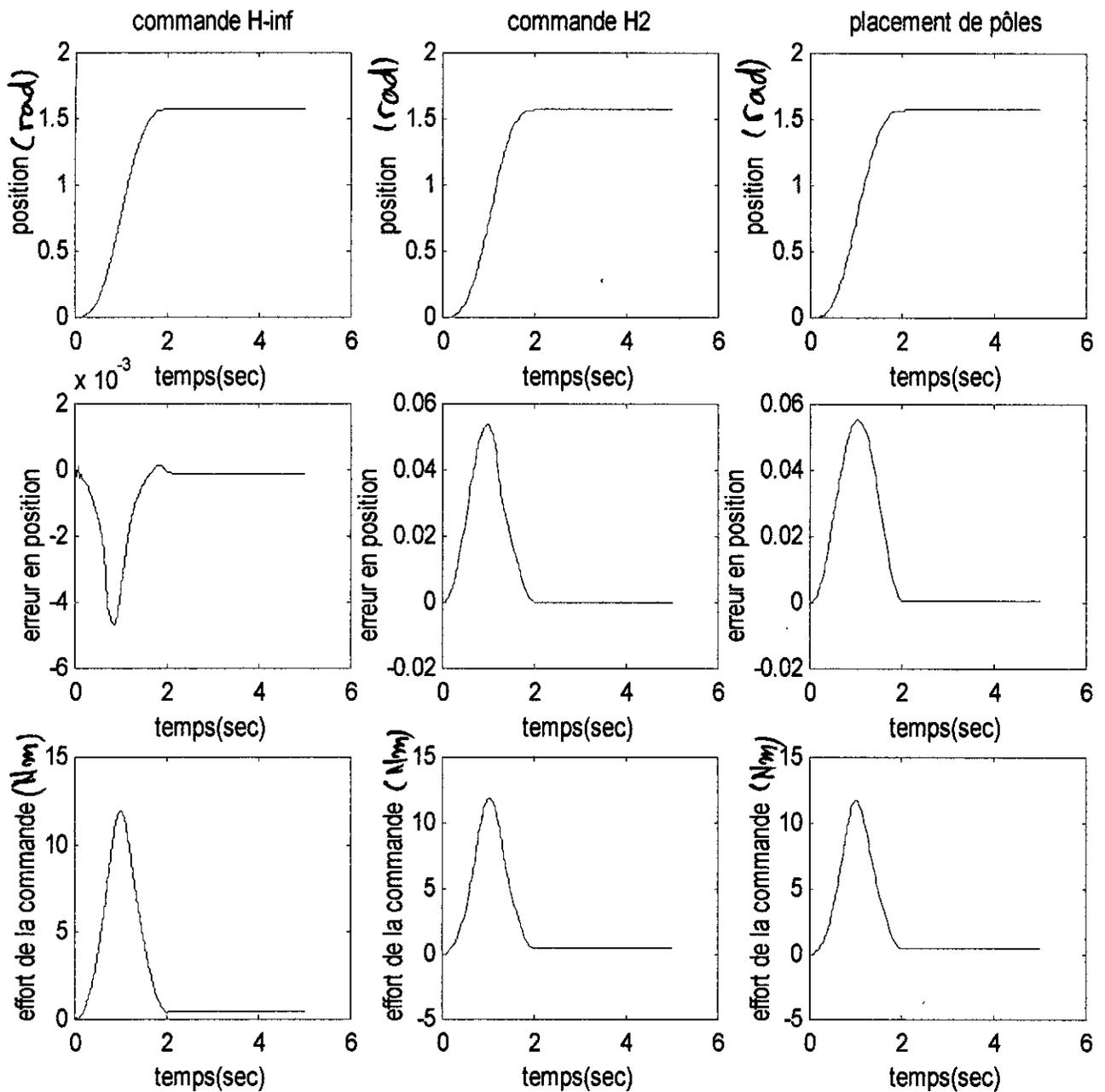
IV.3 TEST DE ROBUSTESSE VIS A VIS DE LA MASSE :

Dans ce test, nous introduisons une sorte de variation paramétrique qui consiste à faire varier la masse soulevée de 5 kg à zéro après que le manipulateur ait atteins sa position finale.

Ceci nous permettra de voir l'influence d'une variation paramétrique due à la masse soulevée. C'est un test standard qu'on retrouve dans plusieurs travaux [27].

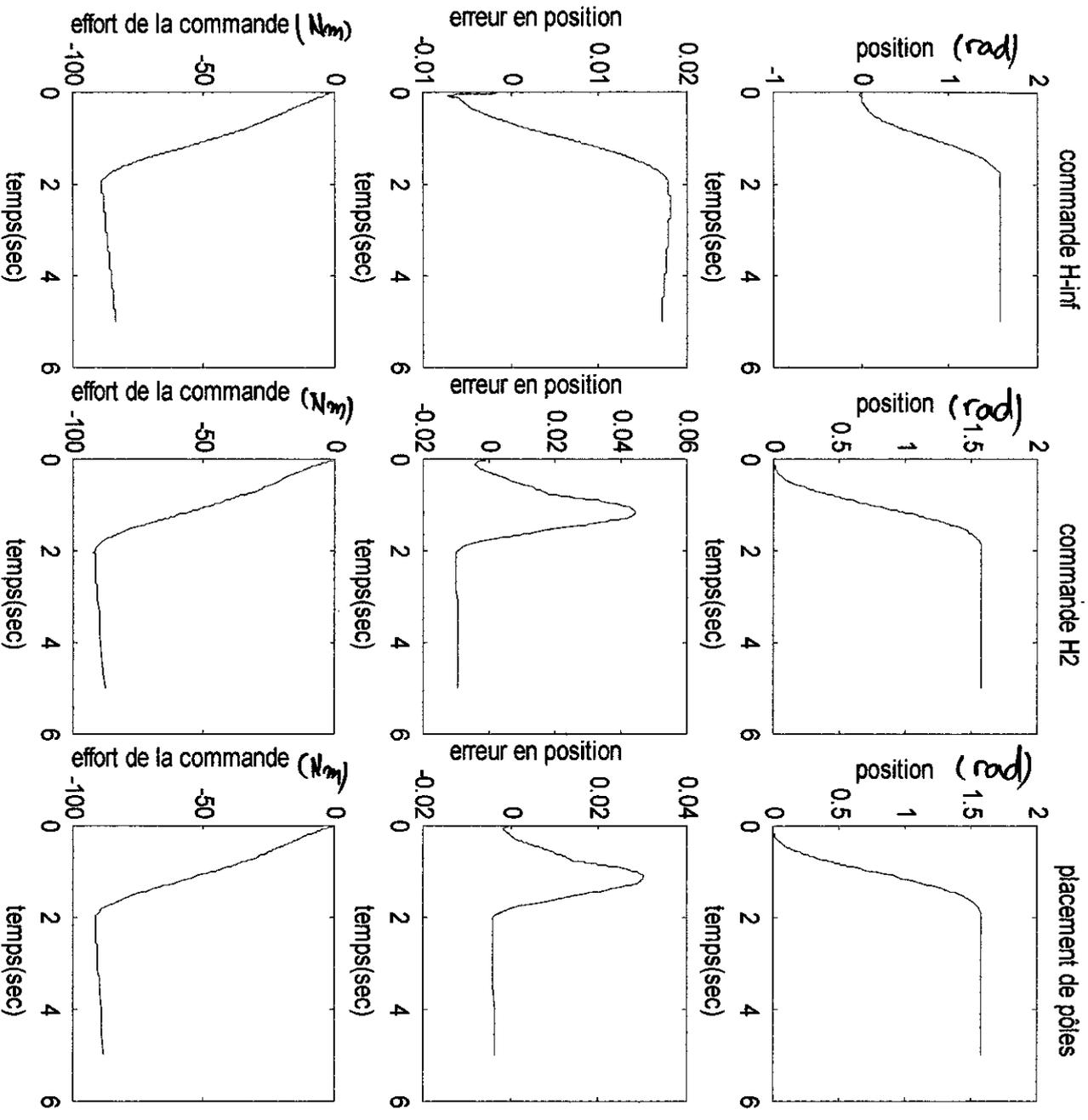
Les figure (IV.4-6) représentent les réponses temporelles des trois articulation à des consignes cycloïdales.

Première articulation :



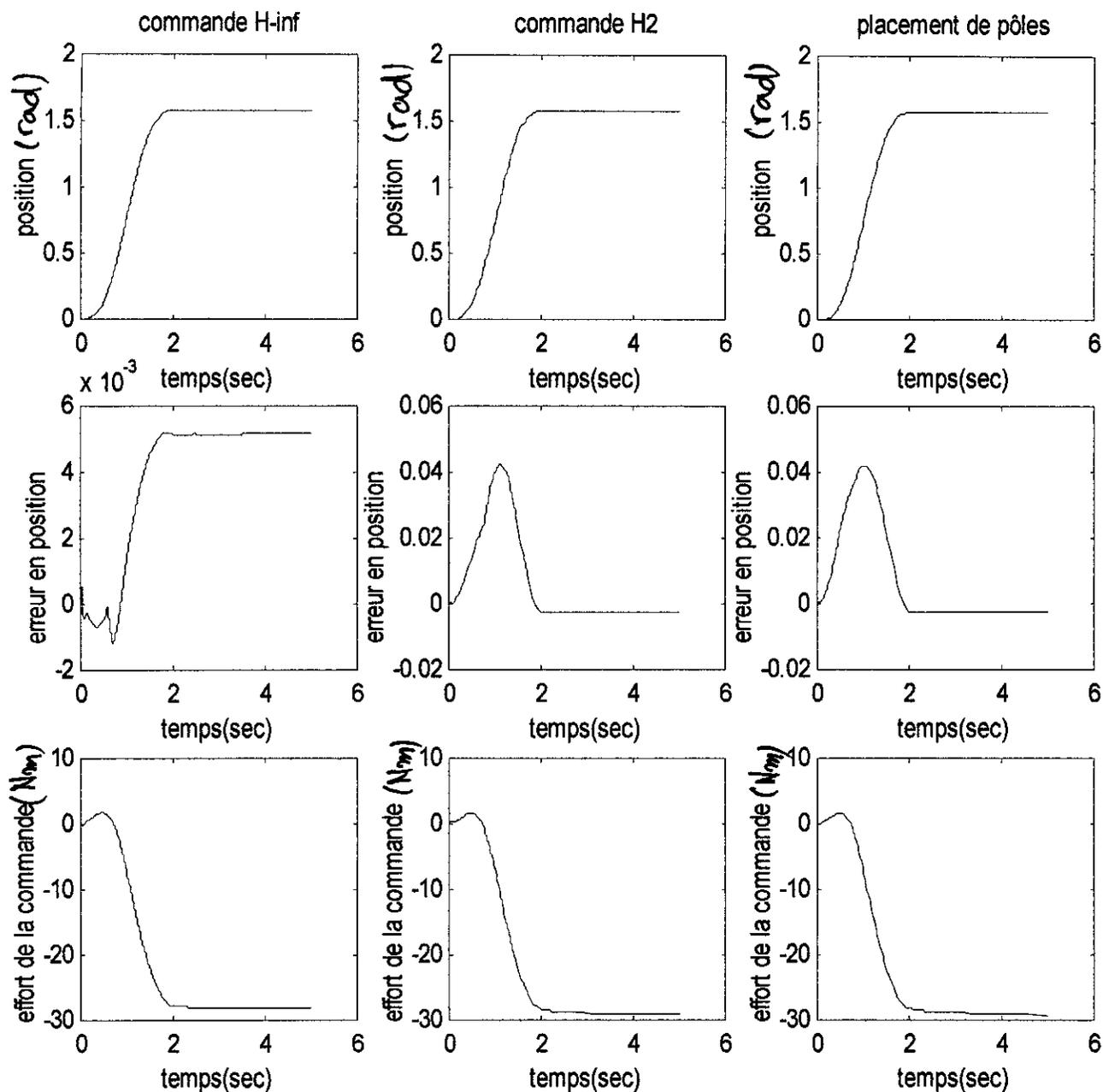
Figure(IV.4) effet de la chute de la masse sur la première articulation

Deuxième articulation :



Figure(IV.5) effet de la chute de la masse sur la deuxième articulation

Troisième articulation :



Figure(IV.6) effet de la chute de la masse sur la troisième articulation

IV.3.1 Conclusion :

Les figures (IV.4-6) montrent l'effet de la variation de la masse soulevée par le manipulateur sur les différentes articulations. On remarque que les trajectoires ne sont pas vraiment perturbées. On remarque juste un pic de l'erreur à l'instant où la masse tombe. Chose prévisible étant donné la variation brusque de sa valeur.

En générale, nous avons un bon suivi des trajectoires. Les commandes sont lisses et admissibles. La commande H_∞ est celle qui donne les meilleurs résultats d'un point de vue précision. Pour ce qui est de l'effort développé toutes les commandes sont équivalentes.

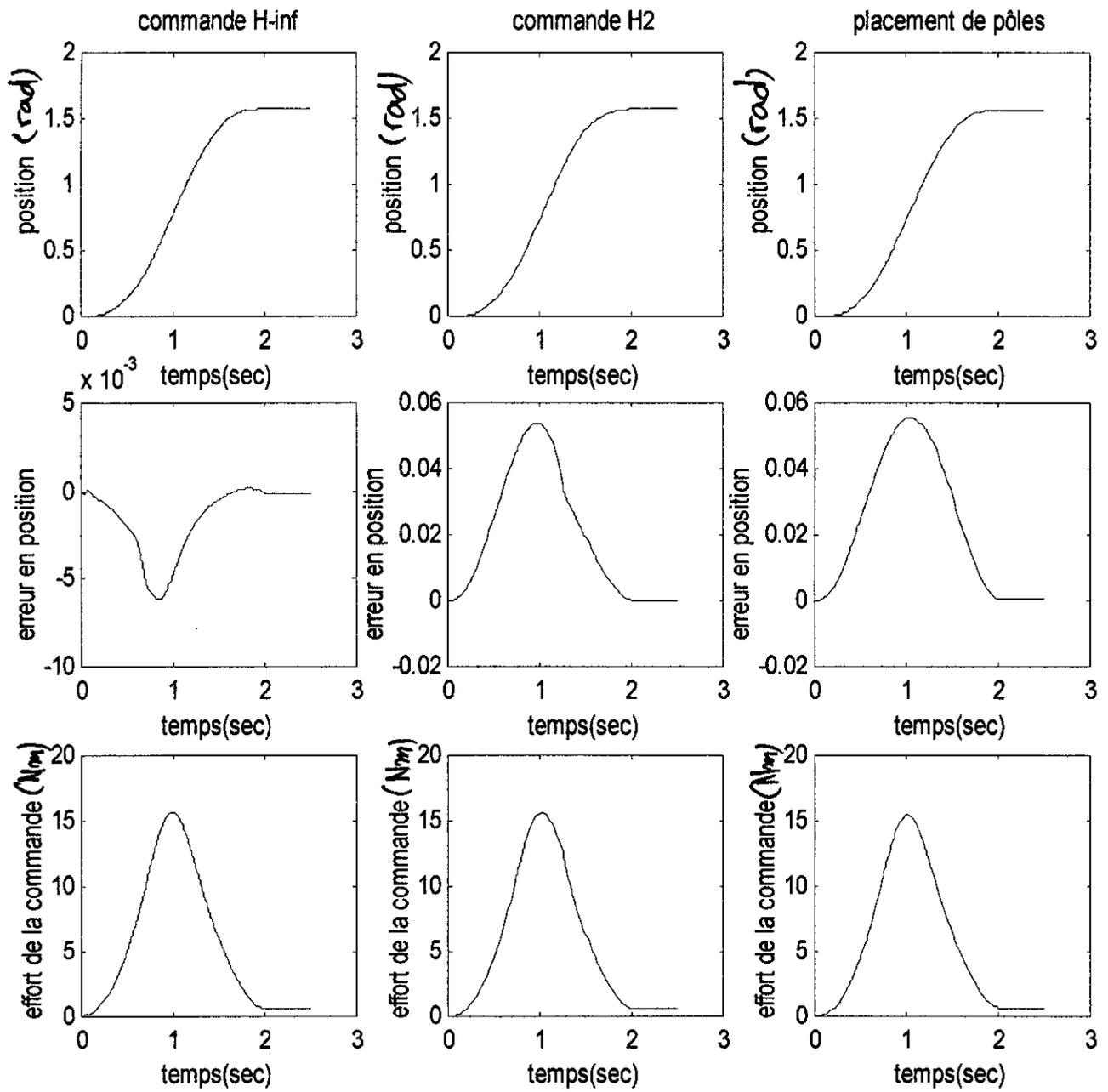
IV.4 TEST DE ROBUSTESSE VIS A VIS DES VARIATIONS PARAMETRIQUES :

Dans ce test on se propose de faire subir à notre modèle mathématique une variation paramétrique. On augmente tous les paramètres du bras manipulateur de 40%. Donc le modèle a une équation de la forme :

$$T = [(M(q) + \Delta M(q)) + 0.4 * (M(q) + \Delta M(q))] \ddot{q} + [N(q, \dot{q}) + 0.4 * N(q, \dot{q})] + G(q) + T_{m_0} \quad (\text{IV.1})$$

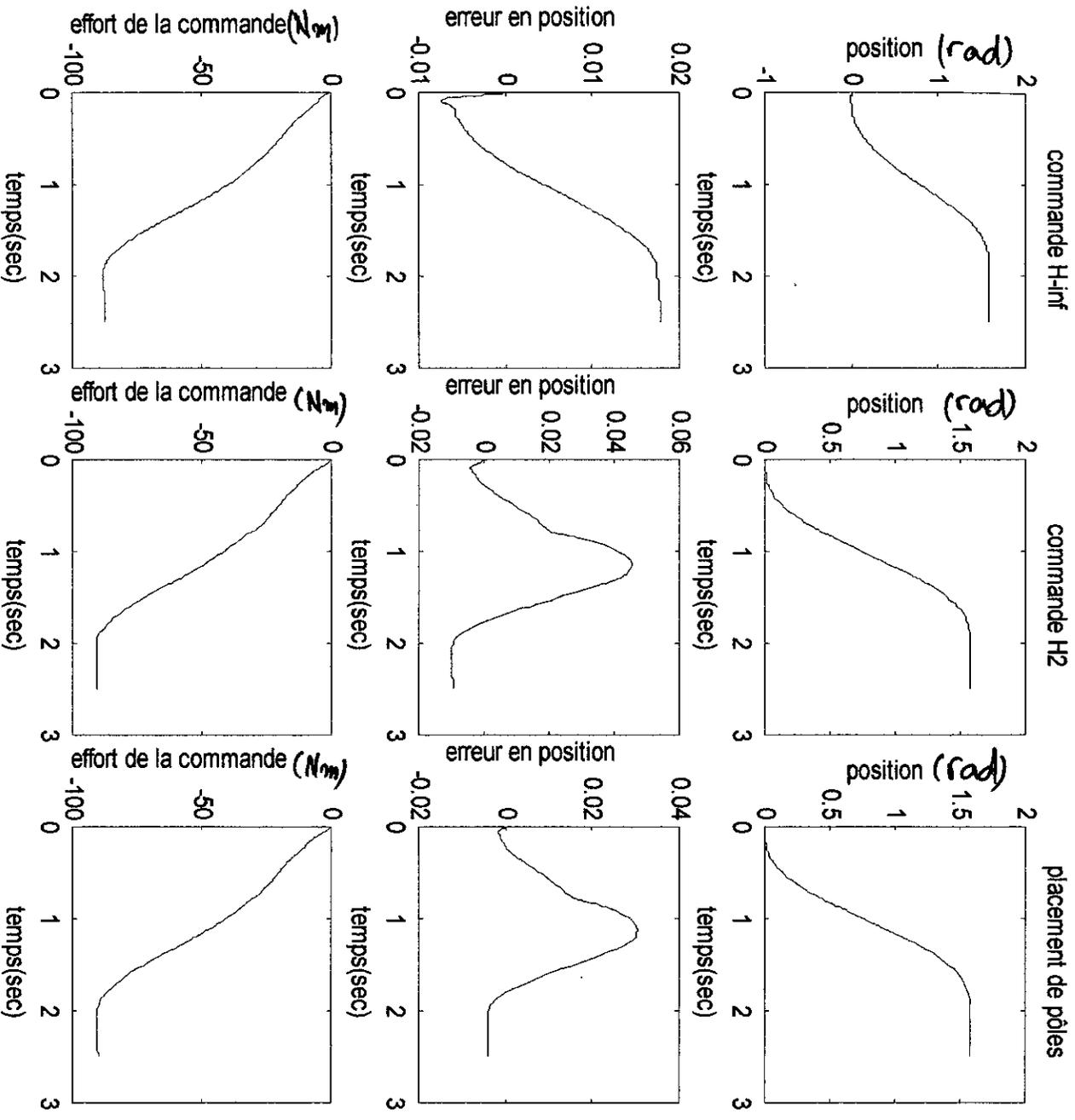
Puis nous appliquons les contrôleurs synthétisés afin de voir leur robustesse face à des variations paramétriques. Les figures qui suivent représentent les réponses des articulations à des consignes cycloïdales.

Première articulation :



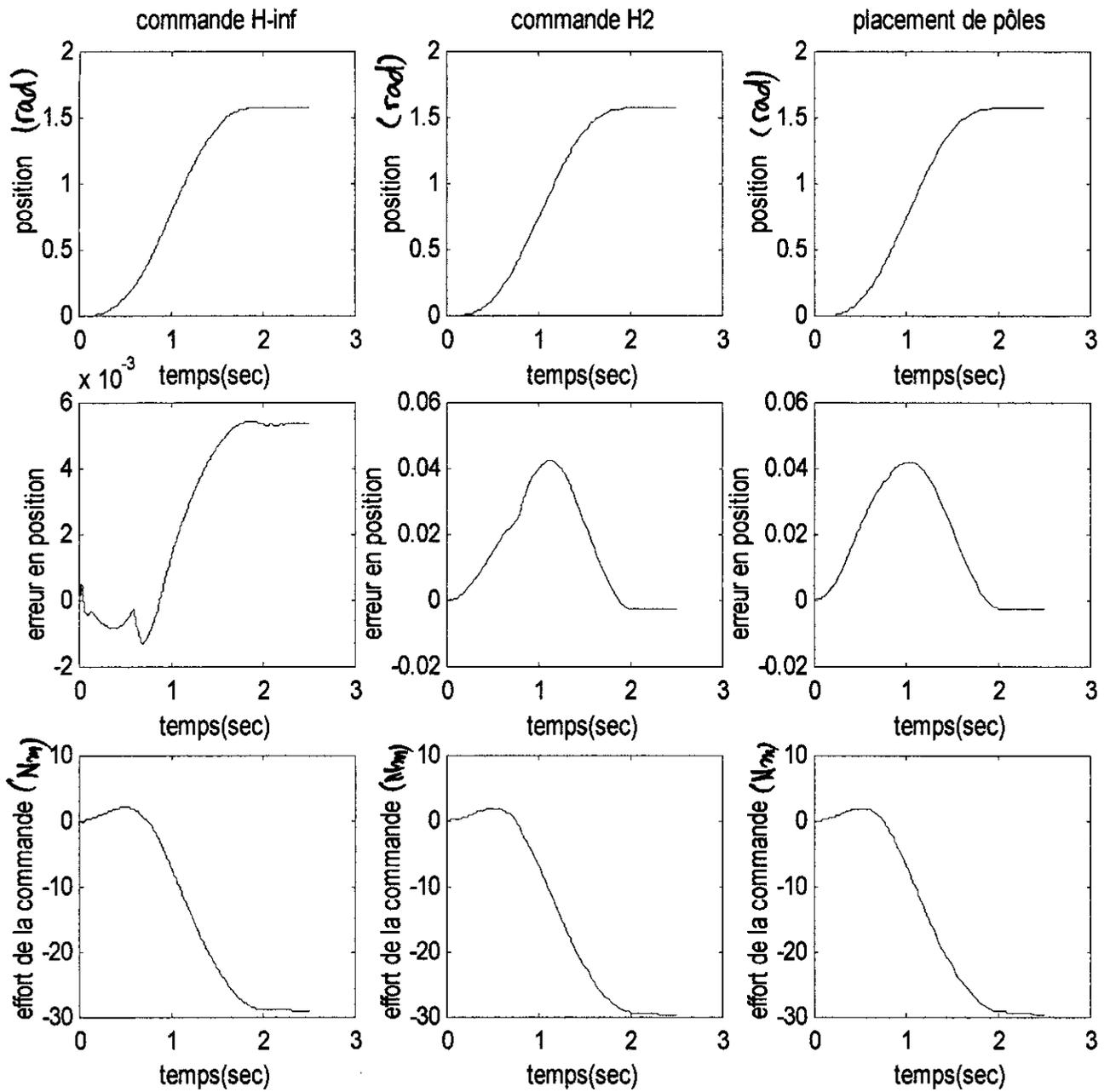
Figure(IV.7) effet des variations paramétriques sur la première articulation

Seconde articulation :



Figure(IV.8) effet des variations paramétriques sur la deuxième articulation

Troisième articulation :



Figure(IV.9) effet des variations paramétriques sur la troisième articulation

IV.4.1 Conclusion :

Les figures précédentes montrent que malgré l'augmentation des paramètres du bras manipulateur de 40% les trois articulations arrivent à suivre la consigne d'une manière assez satisfaisante. Bien que l'erreur dynamique ait augmenté les commandes sont toujours dans un domaine admissible et ont une forme lisse. L'augmentation de l'erreur en régime dynamique est principalement due à l'augmentation de l'inertie du robot. Les contrôleurs ont de plus en plus de mal à assujettir les articulations à suivre leurs trajectoires. Mis à part l'erreur dynamique il n'y a aucun changement notable de la réponse relativement à celle obtenue avec le modèle exacte du robot.

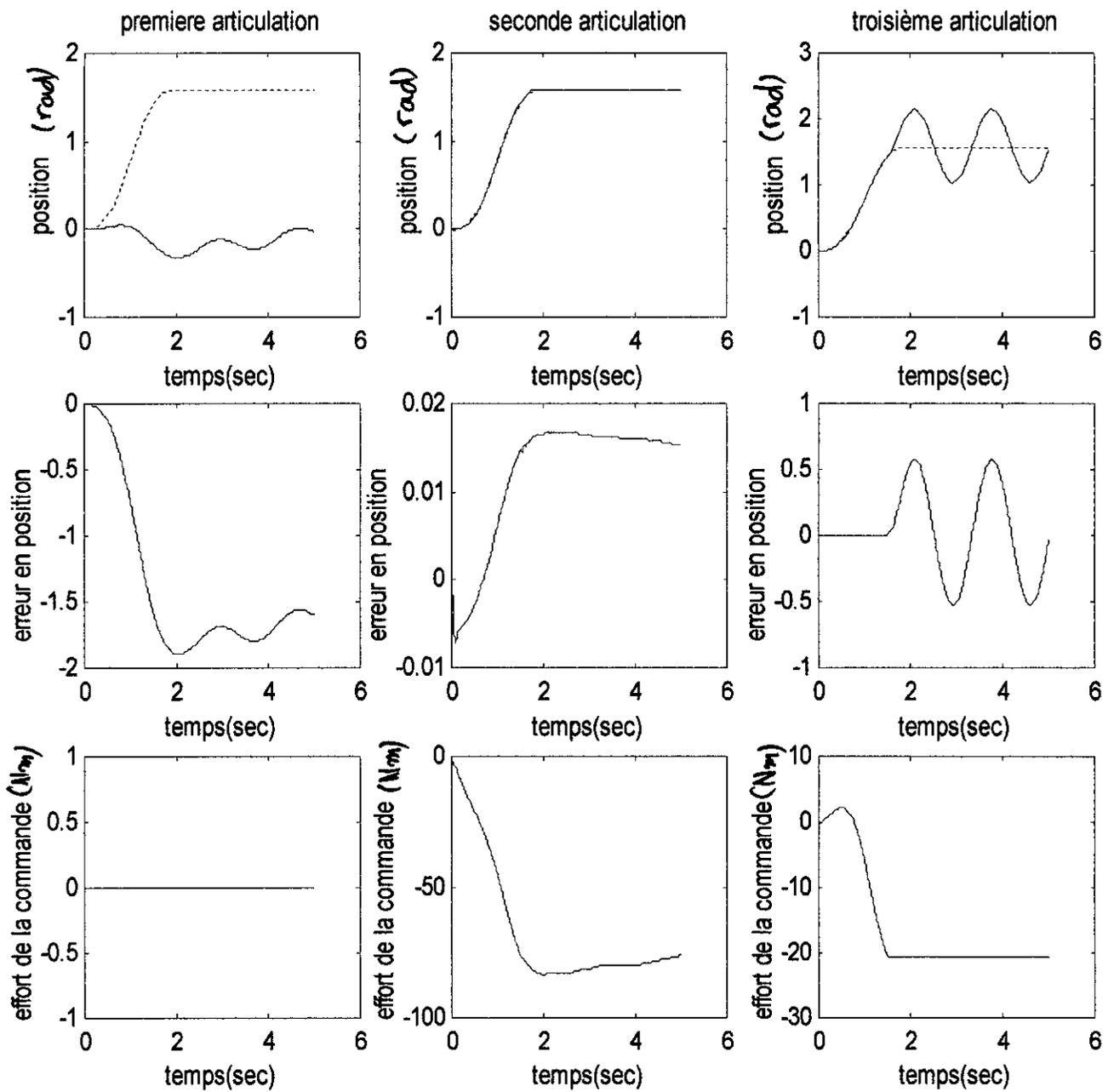
IV.5 TEST DE DECENTRALISATION :

A présent, nous procédons à un autre test de robustesse de la commande. Il s'agit de maintenir une ou deux commandes des trois articulations à zéro. C'est à dire qu'on simule une sorte de rupture de commande survenue lors du fonctionnement du manipulateur. Ce test nous permet de voir jusqu'à quel degré les commandes des articulations sont indépendantes, bien que notre système soit un système couplé.

Les figures qui suivent montrent le comportement de la deuxième articulation quand la commande de la première articulation est nulle, et que celle de la troisième est interrompue après un temps $t = 1.5 \text{ sec}$.

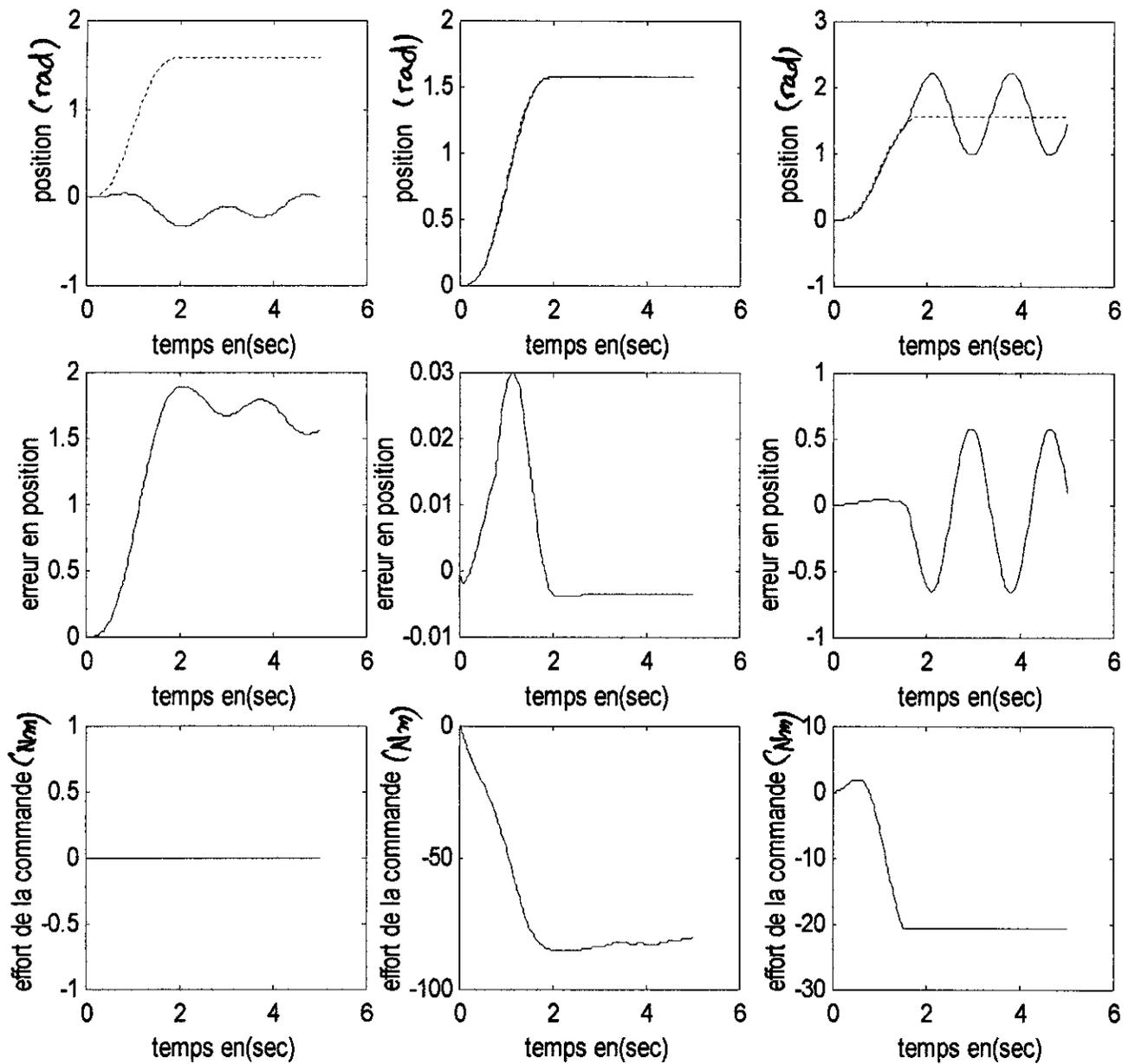
Nous essayons donc de résumer plusieurs tests de décentralisations en un seul afin de ne pas encombrer le document. Nous avons donc le comportement du système quand une seule commande est coupée puis, quand deux commandes sont rompues en même temps. Le choix des commandes à interrompre est fortuit.

Commande H_∞ :



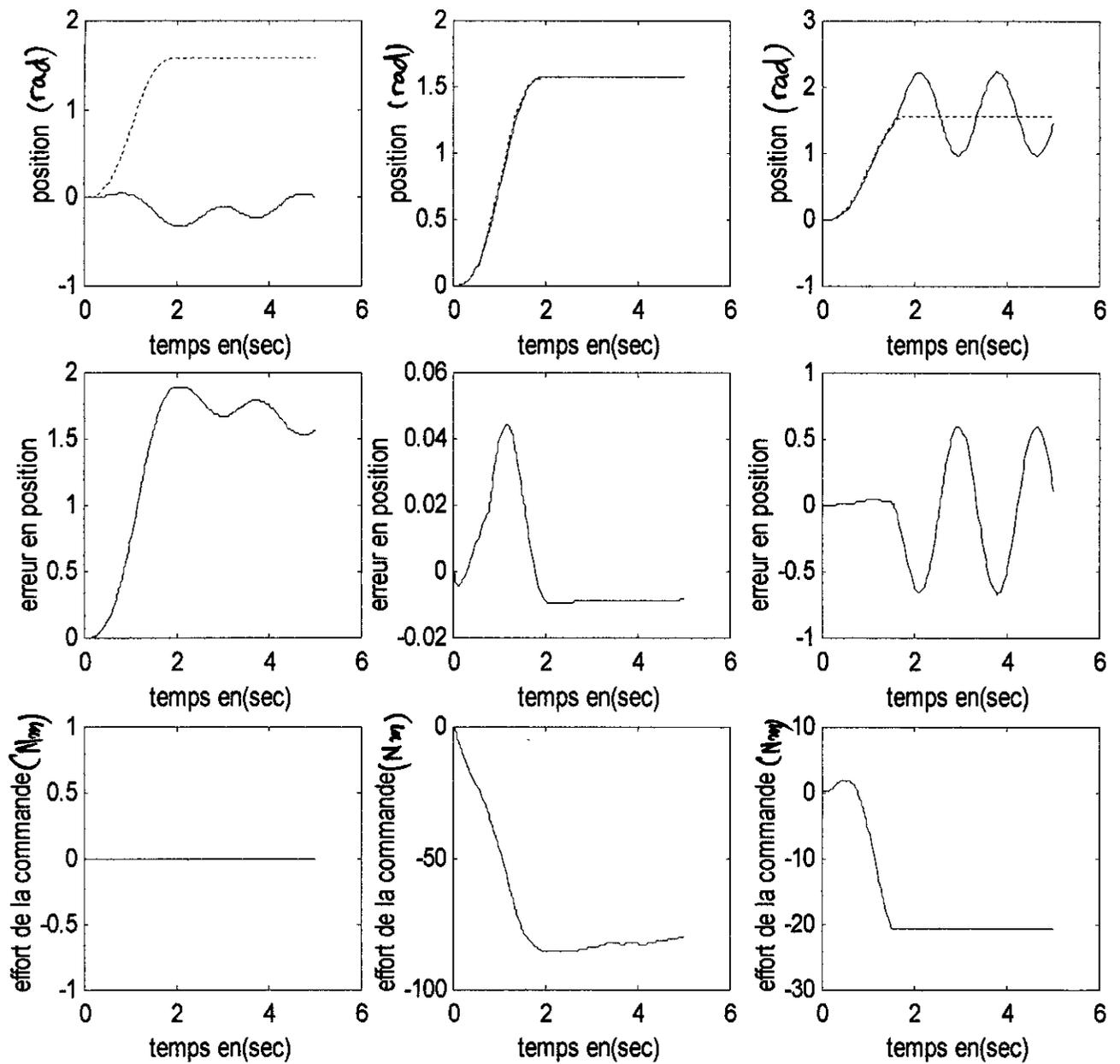
Figure(IV.10) test de décentralisation de la commande H_∞

Placement des pôles :



Figure(IV.11) test de décentralisation de la commande par placement des pôles

Commande H_2 :



Figure(IV.12) test de décentralisation de la commande H_2

IV.5.1 Conclusion :

D'après les tests effectués, on remarque une bonne poursuite de la référence pour les articulations 'contrôlées', bien que l'erreur statique ait sensiblement augmentée. L'allure générale de la commande est acceptable, la valeur maximale de celle-ci reste dans un domaine admissible malgré qu'elle soit plus importante que celle développée durant un fonctionnement normal. Chose tout à fait naturelle car, bien que le contrôleur soit décentralisé, le système lui est fortement couplé. La divergence des variables dont la commande est rompue rend la tâche plus difficile au contrôleur pour minimiser l'erreur de poursuite de l'articulation commandée.

Les erreurs de poursuite pour les articulations commandées sont plus importantes que celles obtenues s'il n'y avait pas rupture de commande. Comme pour les commandes ceci est dû aux interconnexions entre les articulations.

Mais si on comparait nos résultats à ceux obtenus dans [27], on voit une nette amélioration de la poursuite, une erreur moins importante et surtout une meilleure commande d'un point de vue énergétique. Notre résultat est meilleur car la synthèse du contrôleur est basée sur les systèmes flous qui font apparaître l'effet des interconnexions dans les paramètres de Sugeno, alors que dans [27] la synthèse du contrôleur était basée sur un système obtenu par linéarisation autour d'un point de fonctionnement. Les termes qui représentent les interconnexions étaient tout simplement négligés, et donc le contrôleur perd de sa robustesse du fait de la grande erreur de modélisation.

En conclusion, l'interruption de certaines commandes a eu peu d'influence sur les autres articulations à cause de la bonne synthèse des contrôleurs décentralisés qui elle-même est basée sur les systèmes flous de Sugeno.

IV.6 INFLUENCE DE LA DYNAMIQUE DE L'ACTIONNEUR :

Dans les chapitres précédents nous avons vu que les commandes développées pour le contrôle du manipulateur avaient des formes plus ou moins lisses. Afin de remédier à ce problème et dans le soucis d'approcher, le plus possible, la réalité, la sortie du régulateur ne sera plus, directement, appliquée au système mais passera d'abord par un actionneur. Ce dernier aura une certaine dynamique. Nous avons choisi une dynamique de la forme suivante :

$$u_r = \frac{1}{1 + sT} u \quad (\text{IV.2})$$

où :

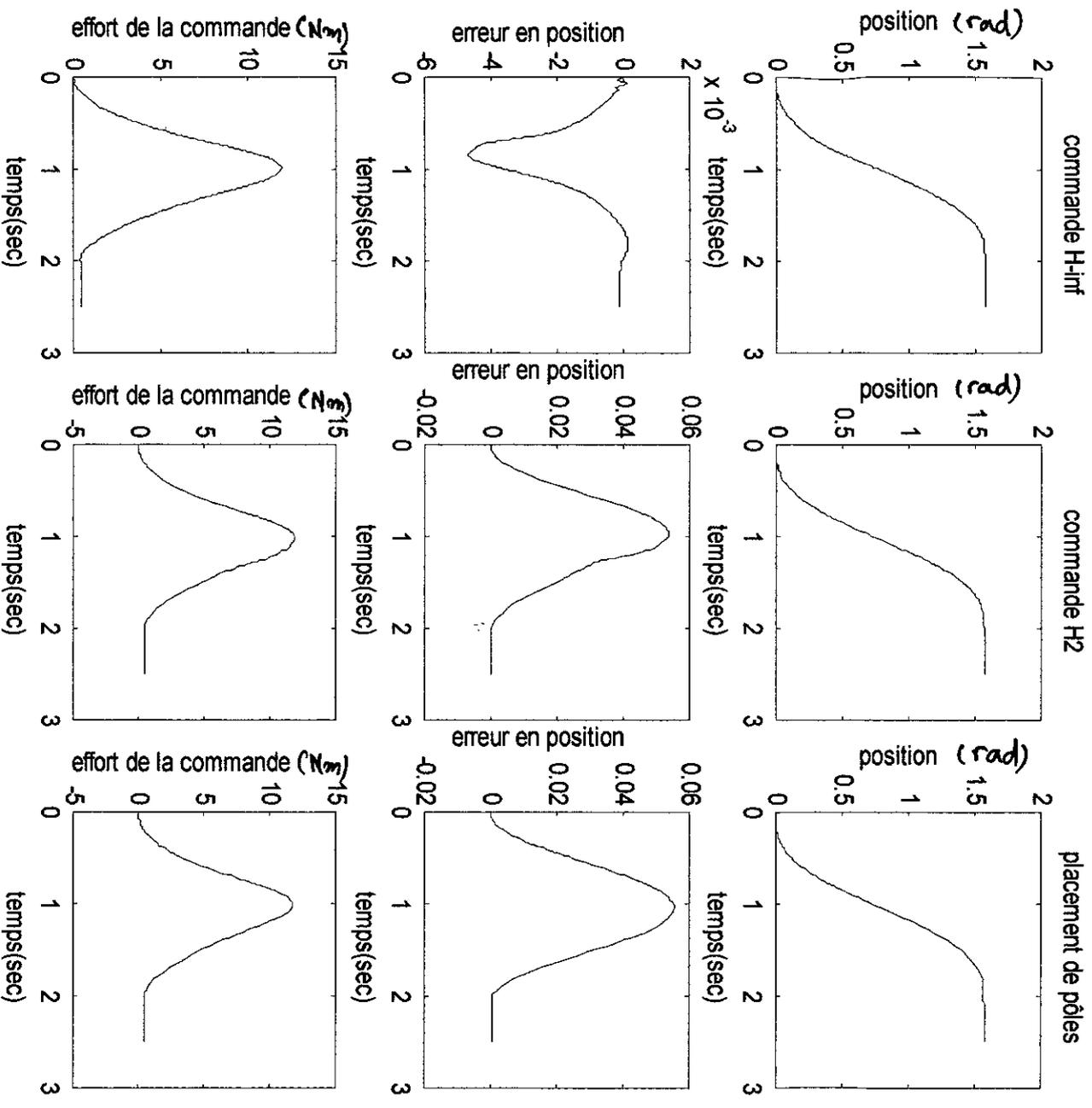
u_r est la commande réelle ou celle qui est appliquée à notre système.

u est la sortie du compensateur.

L'actionneur a une dynamique assez rapide, c'est à dire que la constante de temps T de la fonction de transfert (IV.2) est relativement petite. Pour notre cas nous avons choisi $T = 0.25$ s.

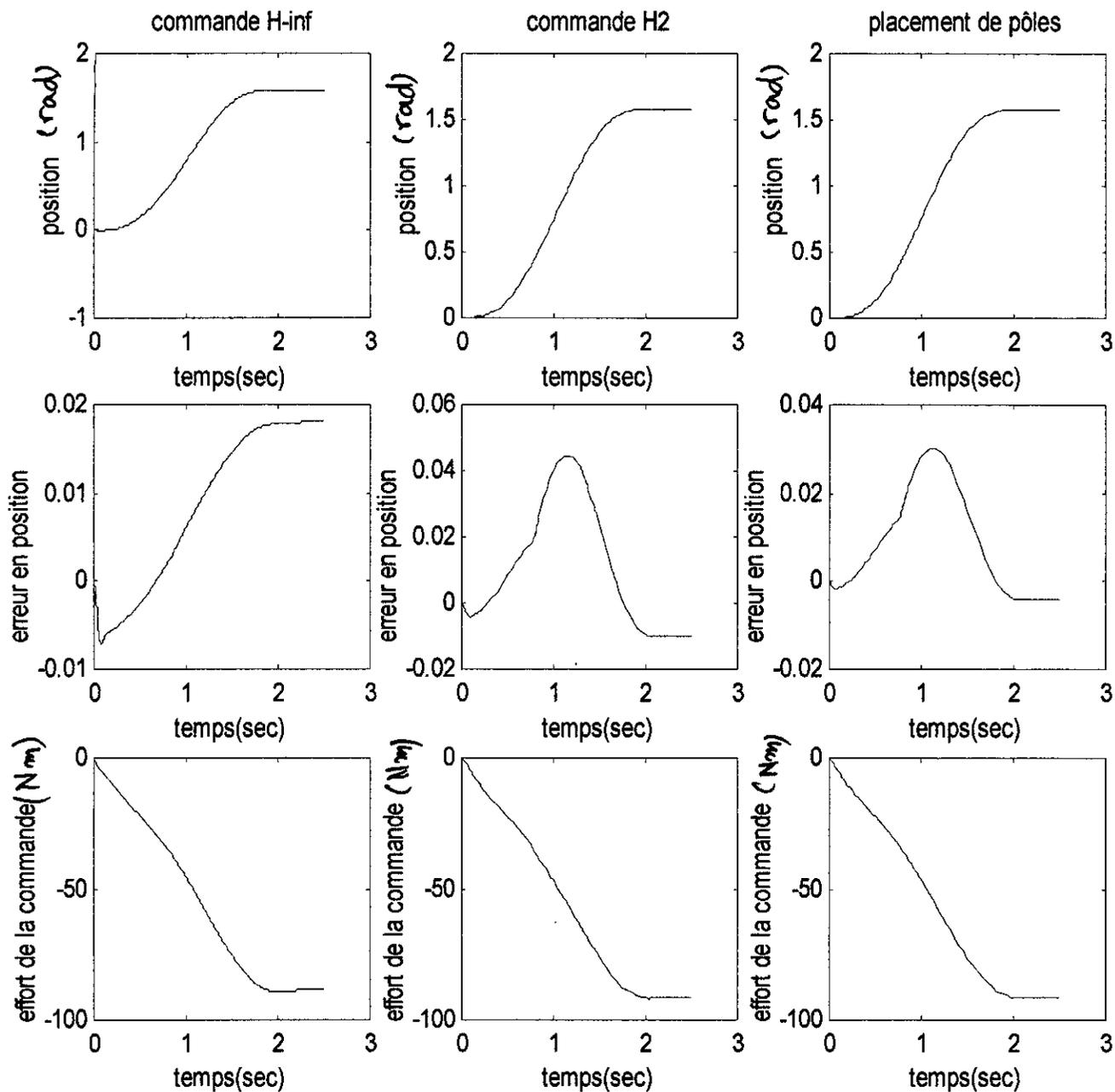
Les figures (IV.13-15) représentent respectivement les trajectoires des trois articulations pour chaque commande.

Première articulation :



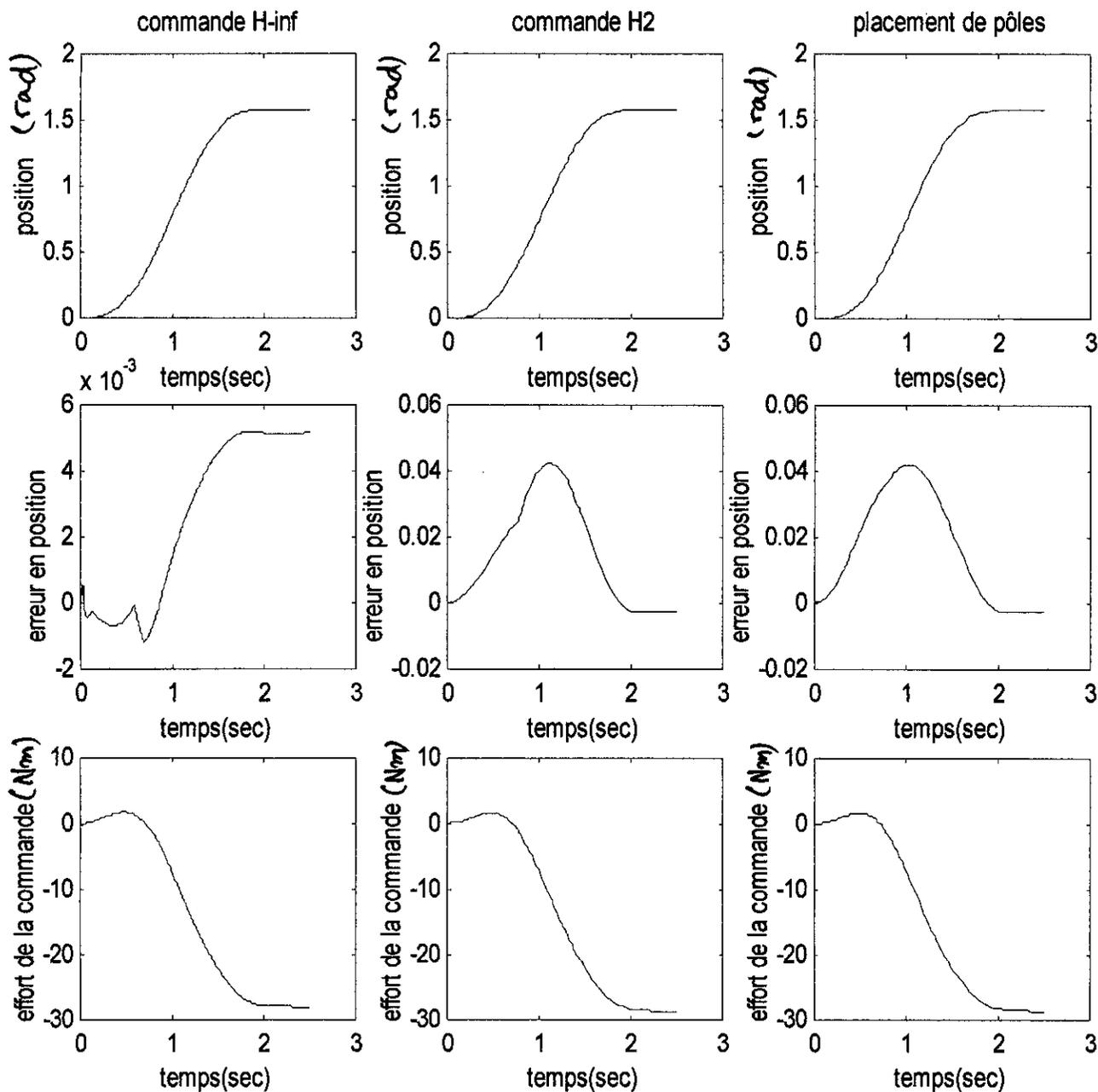
Figure(IV.13) test d'actionneur sur la première articulation

Deuxième articulation :



Figure(IV.14) test d'actionneur sur la seconde articulation

Troisième articulation :



Figure(IV.15) test d'actionneur sur la troisième articulation

IV.6.1 Conclusion :

Nous venons de voir les différentes simulations pour chaque commande, et articulation par articulation. La première remarque que l'on peut faire est le lissage des commandes. En effet, dans toutes les simulations sans actionneur, les variations de la commande sont souvent assez brusques. Mais comme l'actionneur possède une dynamique propre et ne peut dépasser ces limites physiques les variations les plus 'dures' ont été filtrées, car il ne faut pas oublier non plus que la fonction de transfert de l'actionneur est celle d'un filtre passe bas.

Pour ce qui est de la comparaison des commandes entre elles, on remarque que la commande H_∞ est celle qui donne les meilleurs résultats, que ce soit pour l'erreur dynamique ou statique et même l'effort développé par l'actionneur. Ceci n'est pas sans prix, car le contrôleur H_∞ possède un ordre égale à celui du système qui doit être commandé (deux dans notre cas) alors que la commande par retour d'état le contrôleur est un simple gain, soit d'ordre zéro.

D'un point de vue synthèse, le contrôleur H_2 est le plus facile à synthétiser car après avoir mis le système sous forme standard il faut juste résoudre une équation de Riccati. Pour le contrôleur H_∞ , il faut d'abord trouver les poids permettant d'avoir la forme requise des valeurs singulières puis résoudre deux équations de Riccati, pour le placement des pôles la synthèse est assez aisée dans le sens où le contrôleur est dérivé par résolution d'un système d'équations algébriques, mais les résultats obtenus dépendent du bon choix des pôles.

IV.7 INFLUENCES DES PERTURBATIONS EXTERNES :

A présent, on se propose de simuler l'effet d'une perturbation externe que génère généralement l'environnement dans lequel évolue le manipulateur. En effet, dans la pratique, l'état mesuré pour le calcul de la commande est entaché de perturbations. Il est donc judicieux de voir le comportement de notre contrôleur face à une telle perturbation. En s'attend à ce que la commande H_∞ soit la plus robuste face à ce type de perturbation car la synthèse du compensateur nécessite une forme bien spécifique des fonctions de sensibilité pour justement atténuer l'effet de ces perturbations.

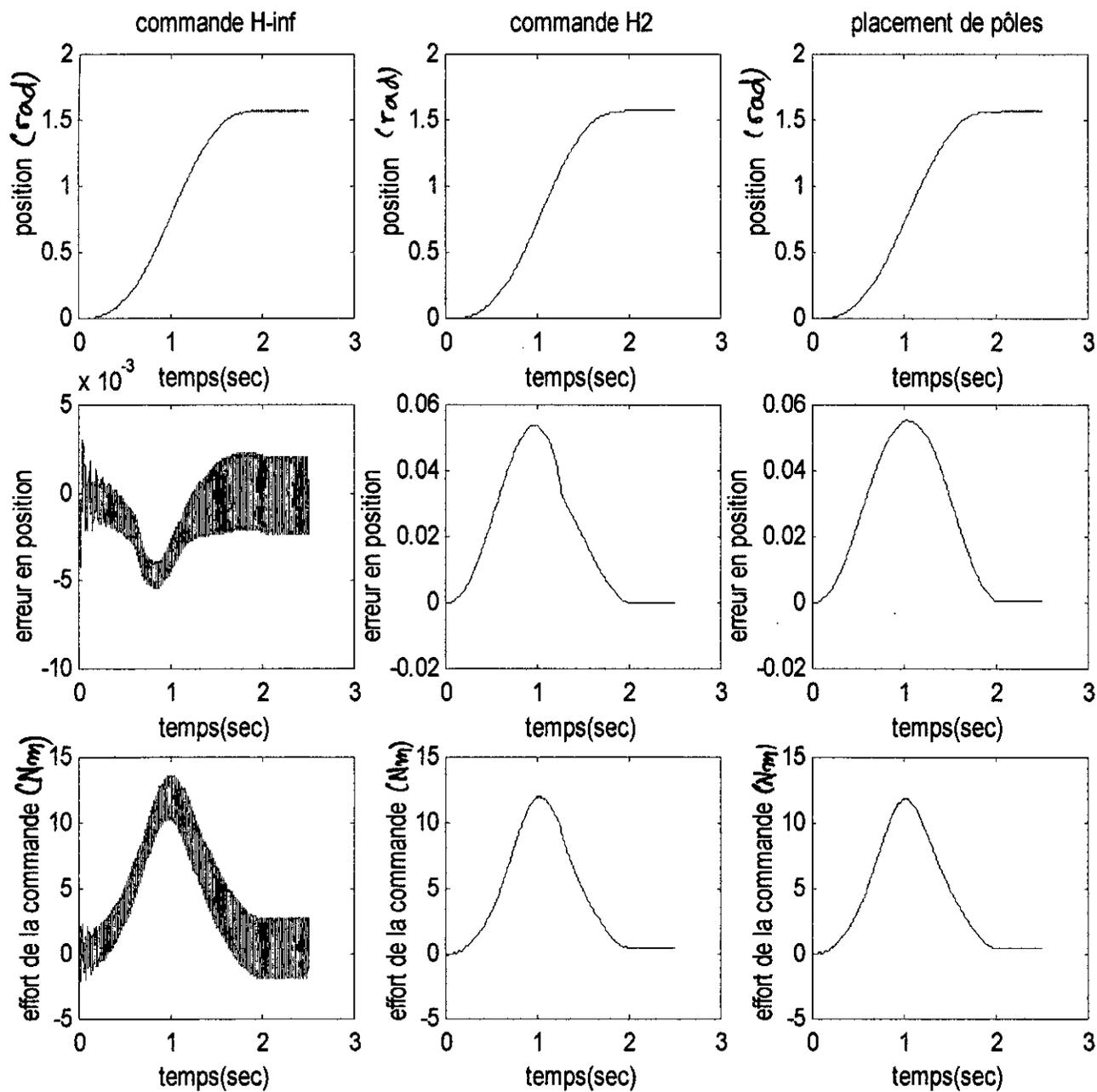
On injecte donc un signal sinusoïdale avec une amplitude relativement basse afin de ne pas noyer la valeur mesurée dans le bruit. Notre choix c'est fixé sur un signal de la forme :

$$b(t) = 0.1 \sin(100\pi t) \quad (\text{IV.3})$$

Ce dernier est ajouté à l'état à l'entrée de chaque contrôleur. Les figures (IV.16-18) montrent les différentes poursuites des consignes, les erreurs de poursuites et les efforts développés par les actionneurs.

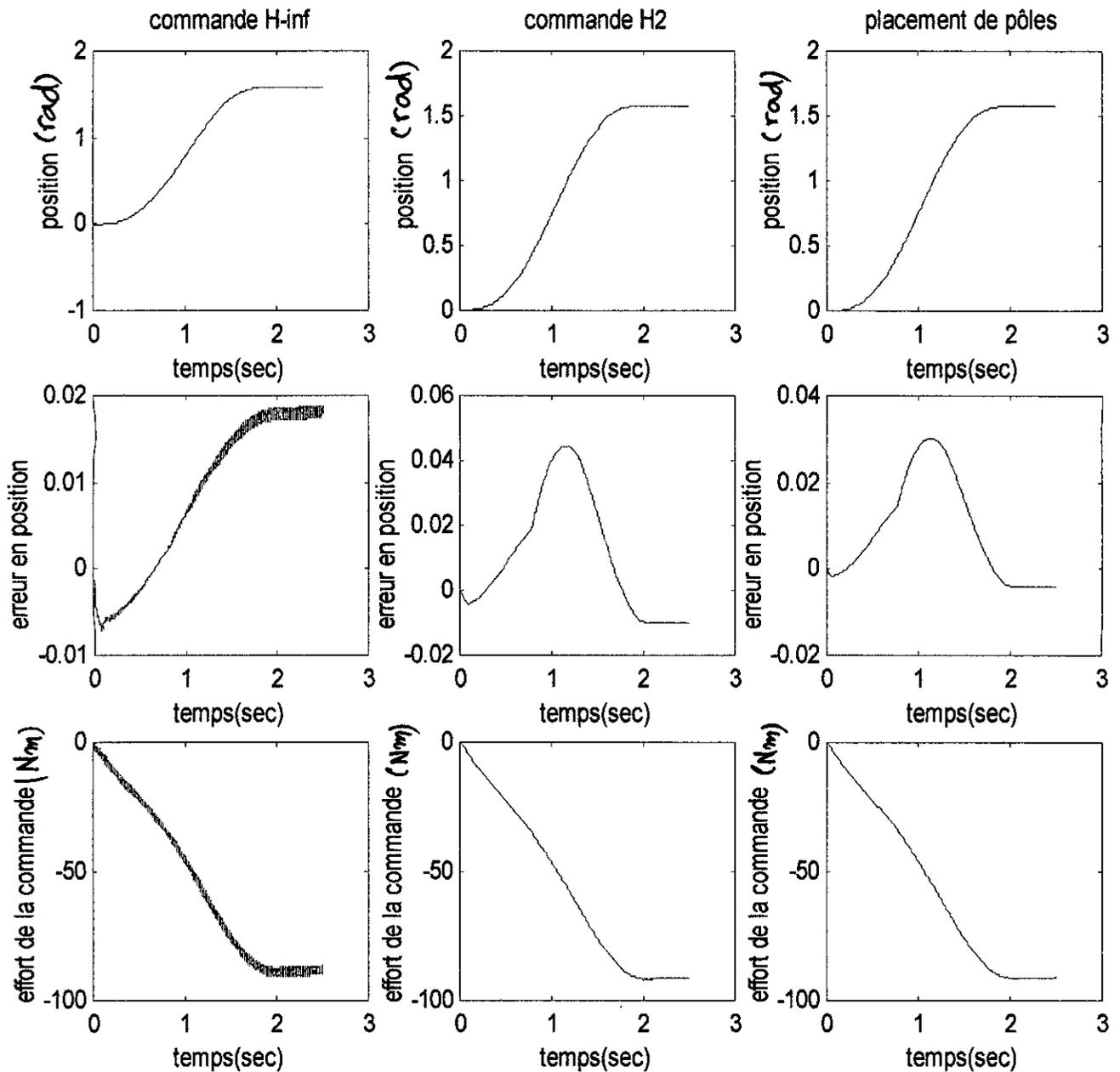
L'effet de l'actionneur est toujours simulé. D'ailleurs il sera ajouté avec tous les autres tests. Et ceci afin d'approcher la réalité au maximum.

Première articulation :



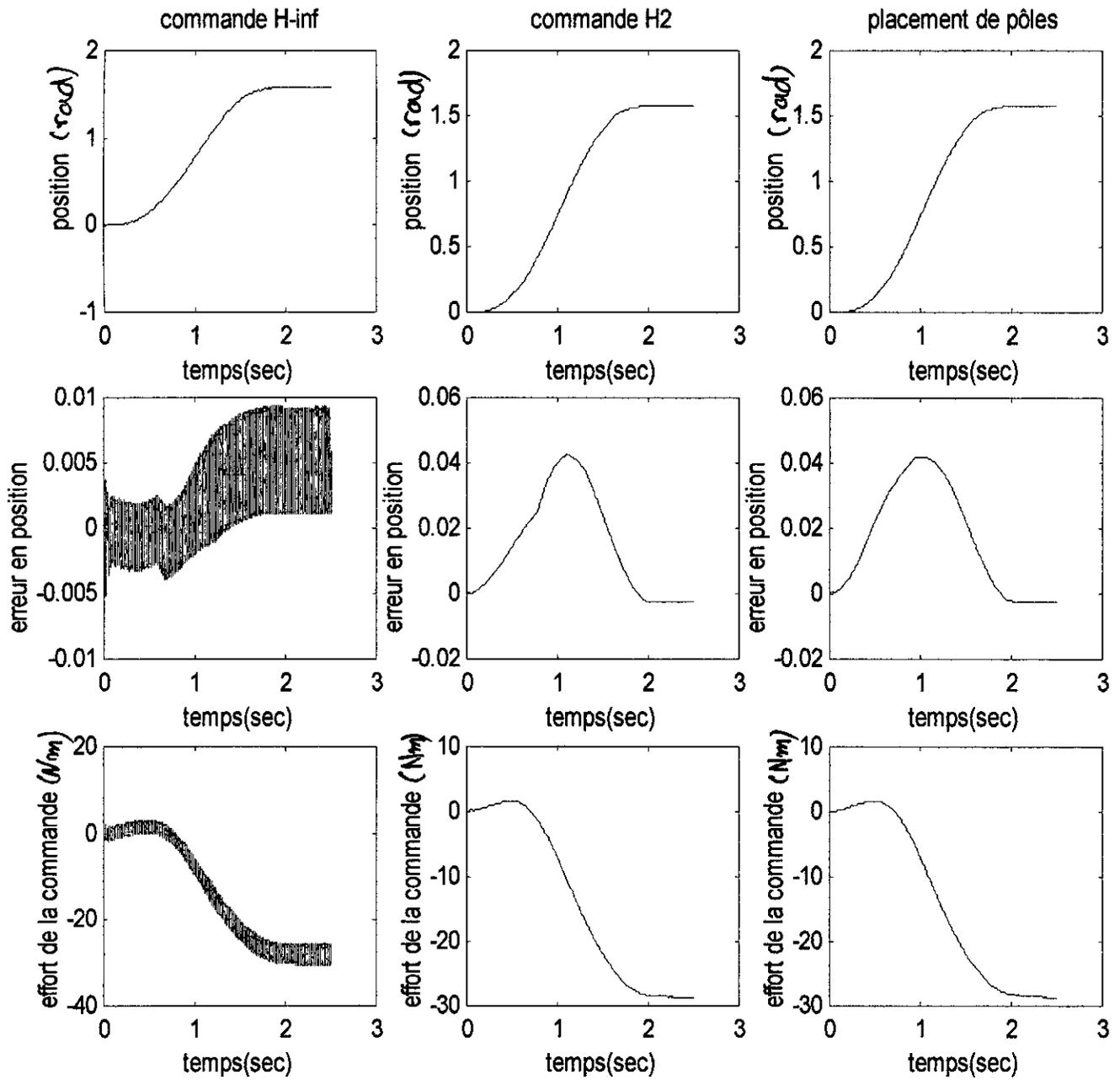
Figure(IV.16) effet des perturbations sur la première articulation

Deuxième articulation :



Figure(IV.17) effet des perturbations sur la deuxième articulation

Troisième articulation :



Figure(IV.18) effet des perturbations sur la troisième articulation

IV.7.1 Conclusion :

Ce test nous a permis de voir le comportement des commandes synthétisées face à des perturbations extérieures. Comme nous l'avons mentionné dans l'introduction de ce paragraphe, on s'attendait à ce que la commande H_∞ ait le meilleur comportement face à de telles perturbations, et les simulations confirment nos prévisions.

On voit aussi que la commande par retour d'état (H_2 ou placement des pôles) a atténué l'effet des perturbations d'une manière plus significative. Ceci est dû au fait qu'en synthétisant le retour d'état c'est la forme générale de la fonction de transfert du système en boucle fermée qu'on imposait. Comme cette dernière est celle d'une fonction de transfert du deuxième ordre avec deux pôles complexes conjugués, alors le système lui-même se comporte comme un filtre passe bas. Donc le système contribue à l'atténuation de l'effet perturbateur des bruits.

De manière générale, les trois commandes ont un comportement satisfaisant face aux bruits de mesure. Toutes les consignes ont été suivies malgré la présence des bruits. Les commandes n'ont pas augmenté et sont maintenues à des niveaux appréciables.

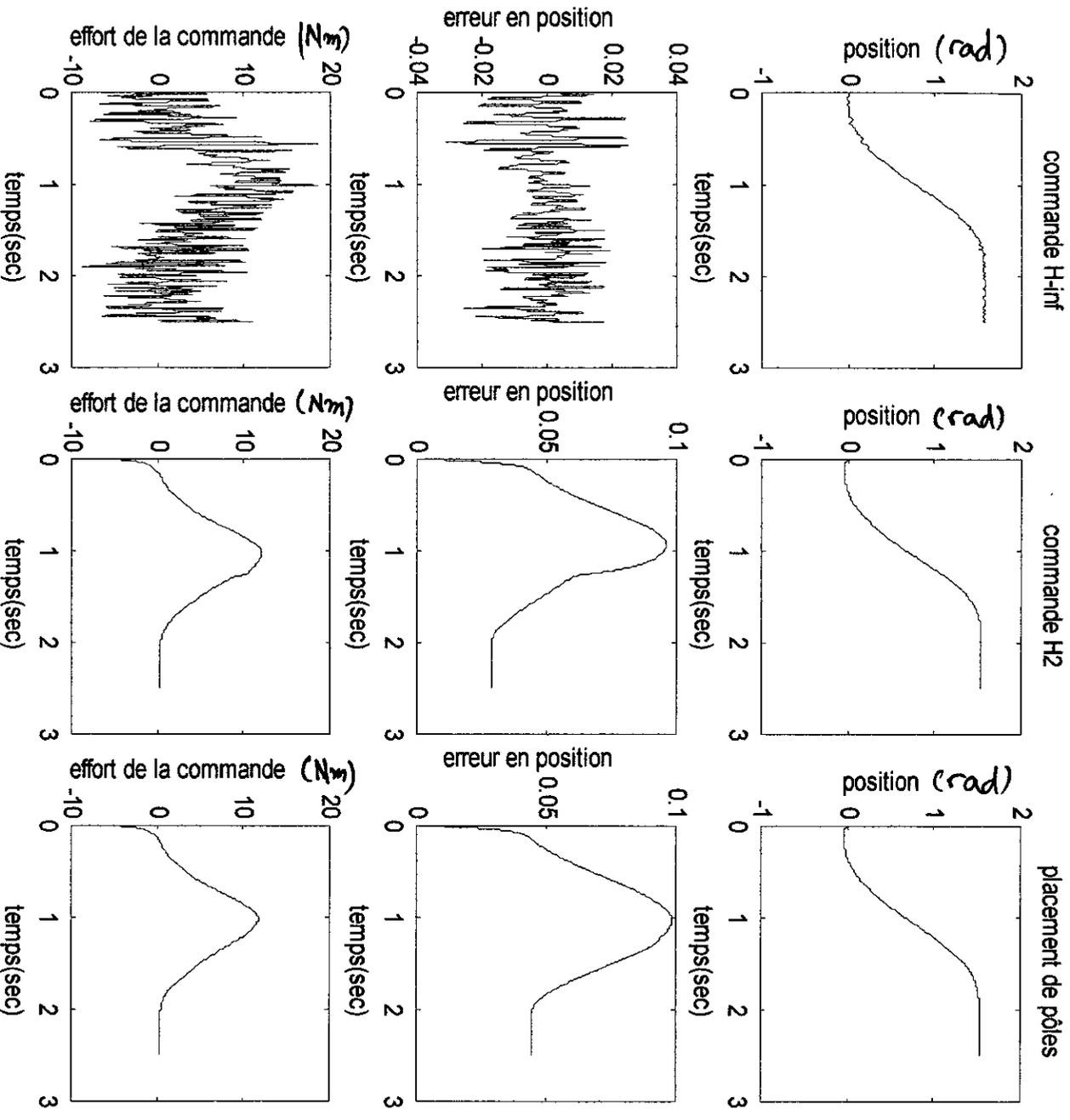
IV.8 INFLUENCE DU BRUIT DE MESURE :

Dans ce paragraphe, nous superposons un bruit blanc à l'état mesuré du système. Dans la pratique, ce bruit est dû aux capteurs. Ces derniers sont généralement à base d'électronique et donc introduisent un bruit de haute fréquence. Son amplitude n'est pas trop grande afin de ne pas noyer l'état mesuré dans le bruit de mesure.

Les figures (IV.19-21) montrent les réponses des différentes articulations à une consigne cycloïdale.

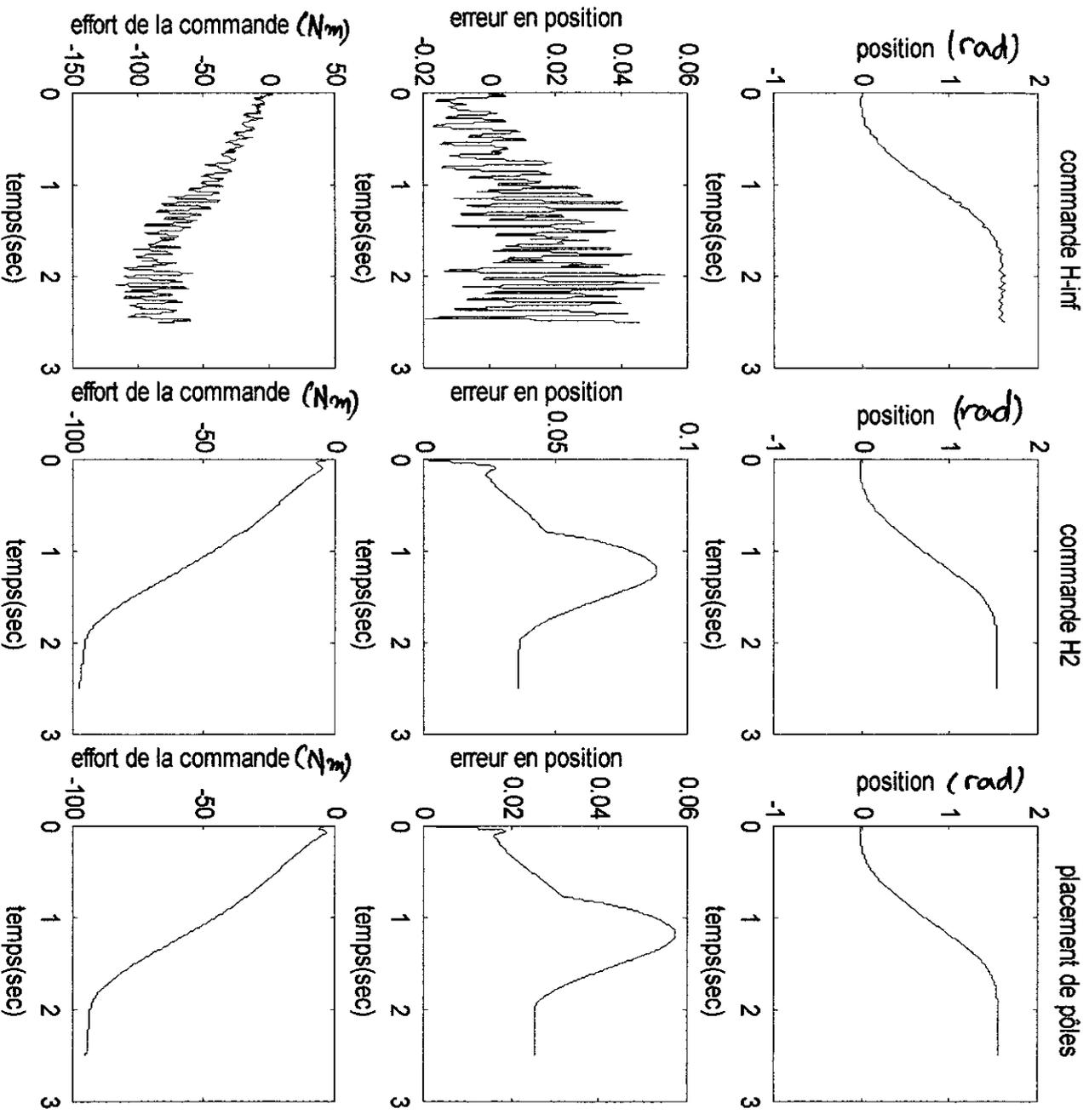
Ce test n'est pas très différent de celui des perturbations juste que le bruit de mesure est généré à l'aide d'une fonction aléatoire.

Première articulation :



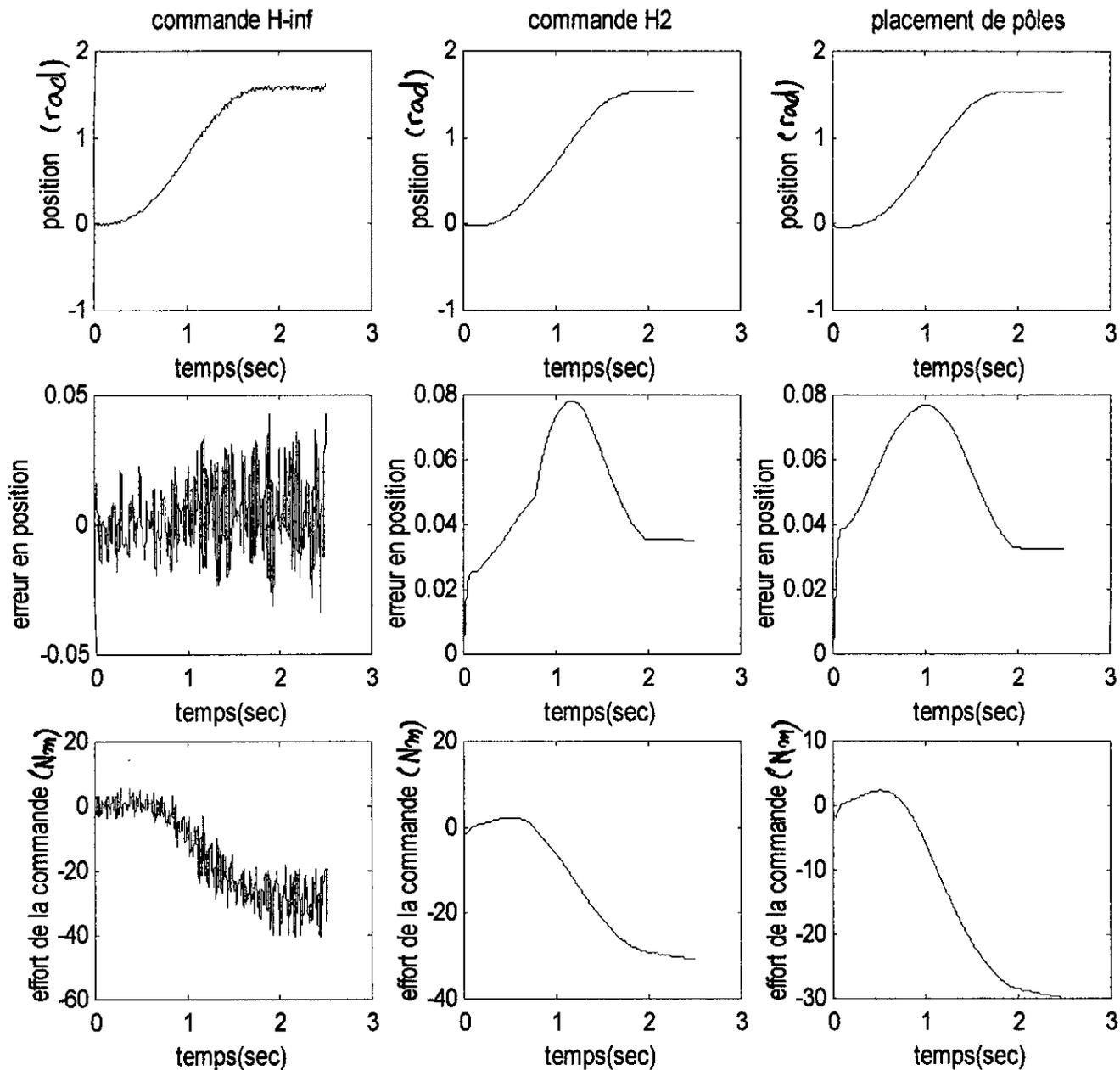
Figure(IV.19) effet du bruit sur la première articulation

Seconde articulation :



Figure(IV.20) effet du bruit sur la deuxième articulation

Troisième articulation :



Figure(IV.21) effet du bruit sur la troisième articulation

IV.8.1 Conclusion :

Nous venons de voir le comportement du système face à des bruit de mesures, généralement dus aux capteurs figures(IV.19-21).

La commande H_∞ est celle qui répond le mieux face à de telles perturbations. L'erreur est très petite. Mais la forme de la réponse en position est moins lisse que celles des positions pour ce qui est des deux autres commandes. Néanmoins l'erreur dynamique est nettement inférieure à celle enregistrée avec la commande H_2 ou la commande par placement des pôles. Mais les deux signaux générés par les contrôleurs par retour d'état sont plus lisses que celui généré par le contrôleur H_∞ .

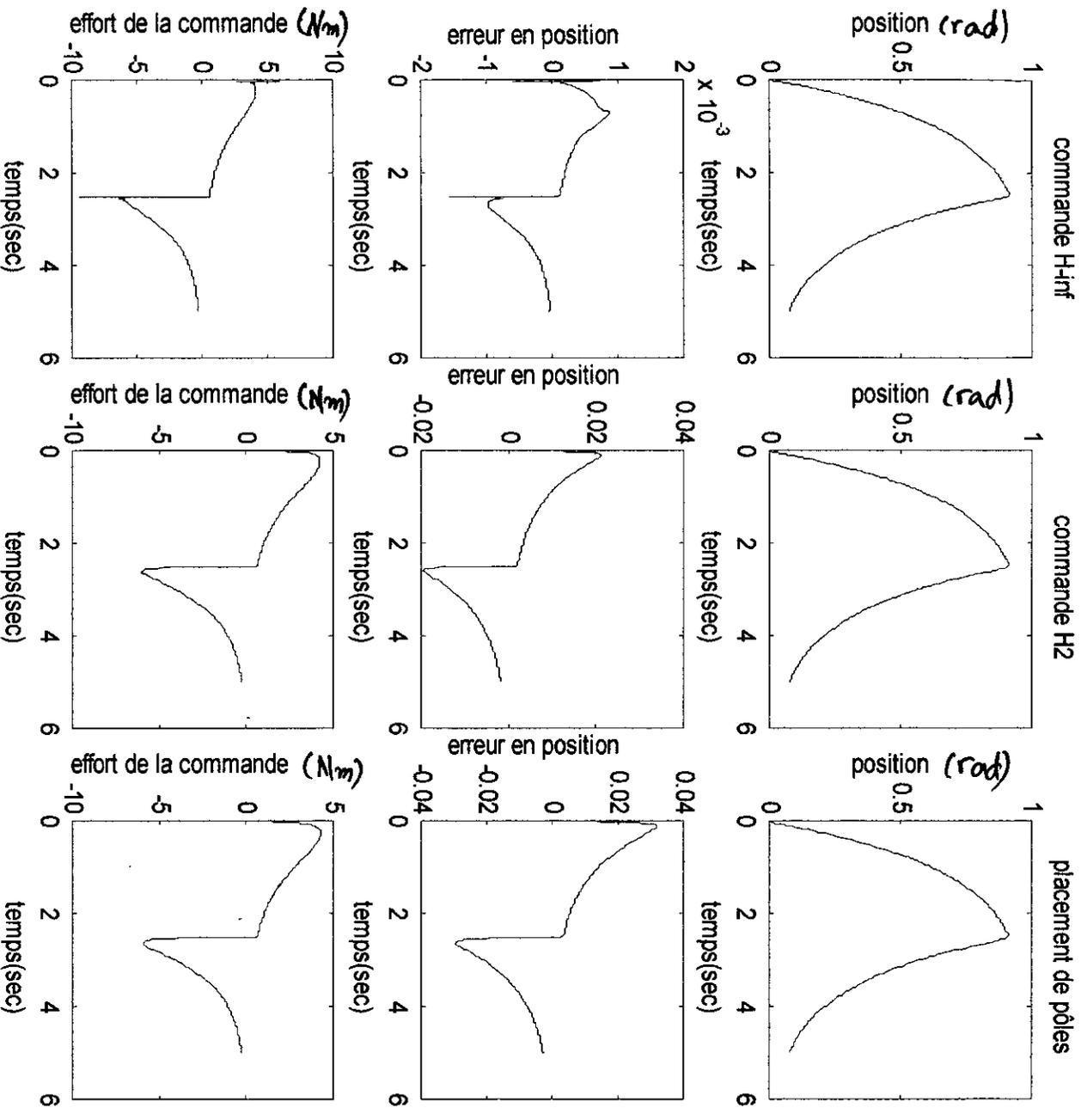
IV.9 POURSUITE D'UNE TRAJECTOIRE CONSTAMMENT VARIABLE :

Dans ce dernier paragraphe, on se propose de changer la consigne. Au lieu d'une trajectoire cycloïdale, les articulations du manipulateurs aurons à suivre une trajectoire qui augmente et diminue d'une manière exponentielle. La vitesse de montée ne sera pas trop importante afin de ne pas dépasser les limites physiques du manipulateur.

La forme générale de la consigne est :

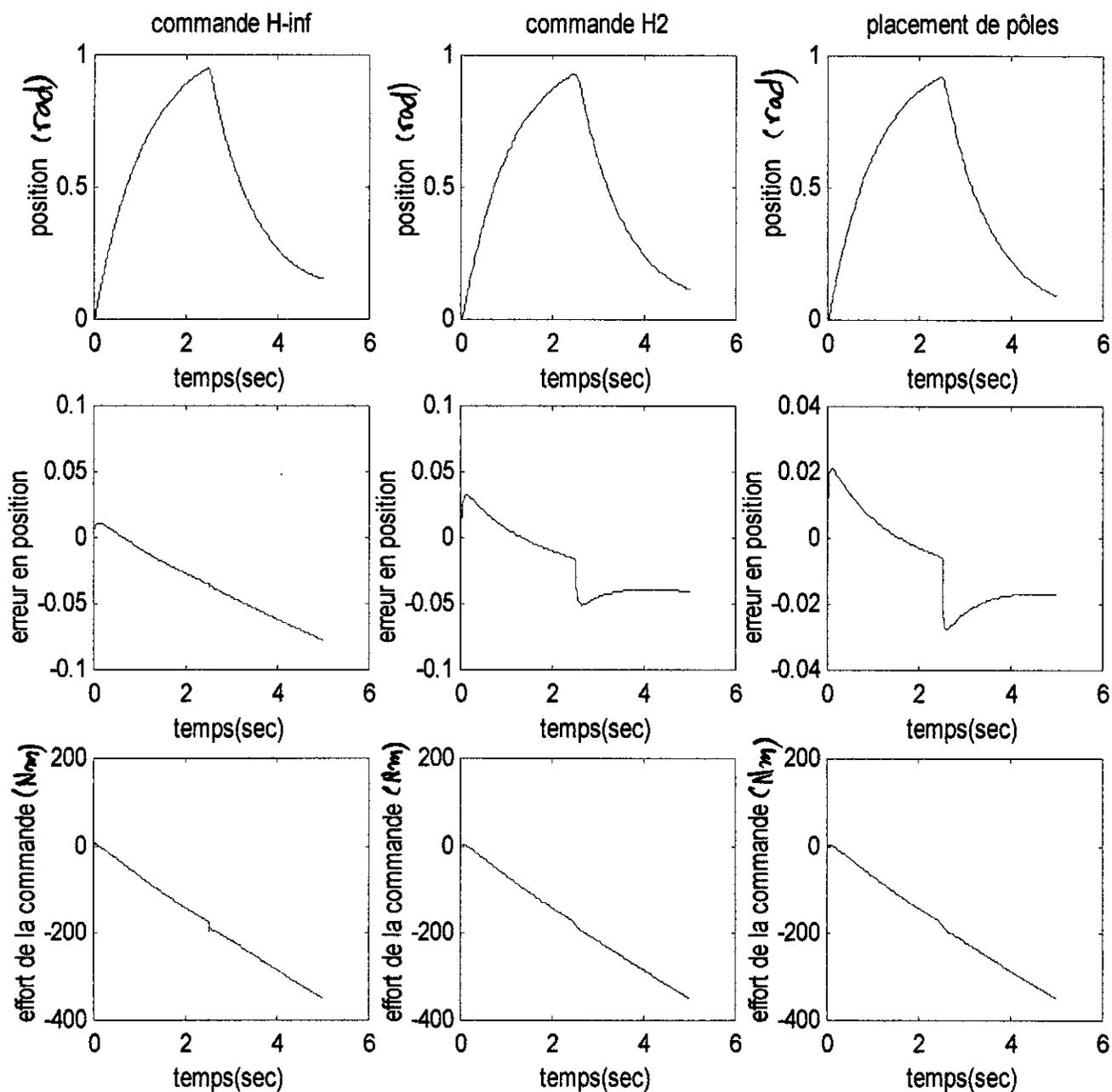
$$r(t) = \begin{cases} r_0 (1 - \exp(-t/\tau)) & t < t_1 \\ r_f \exp(-t/\tau) & t \geq t_1 \end{cases} \quad (\text{IV.4})$$

Première articulation :



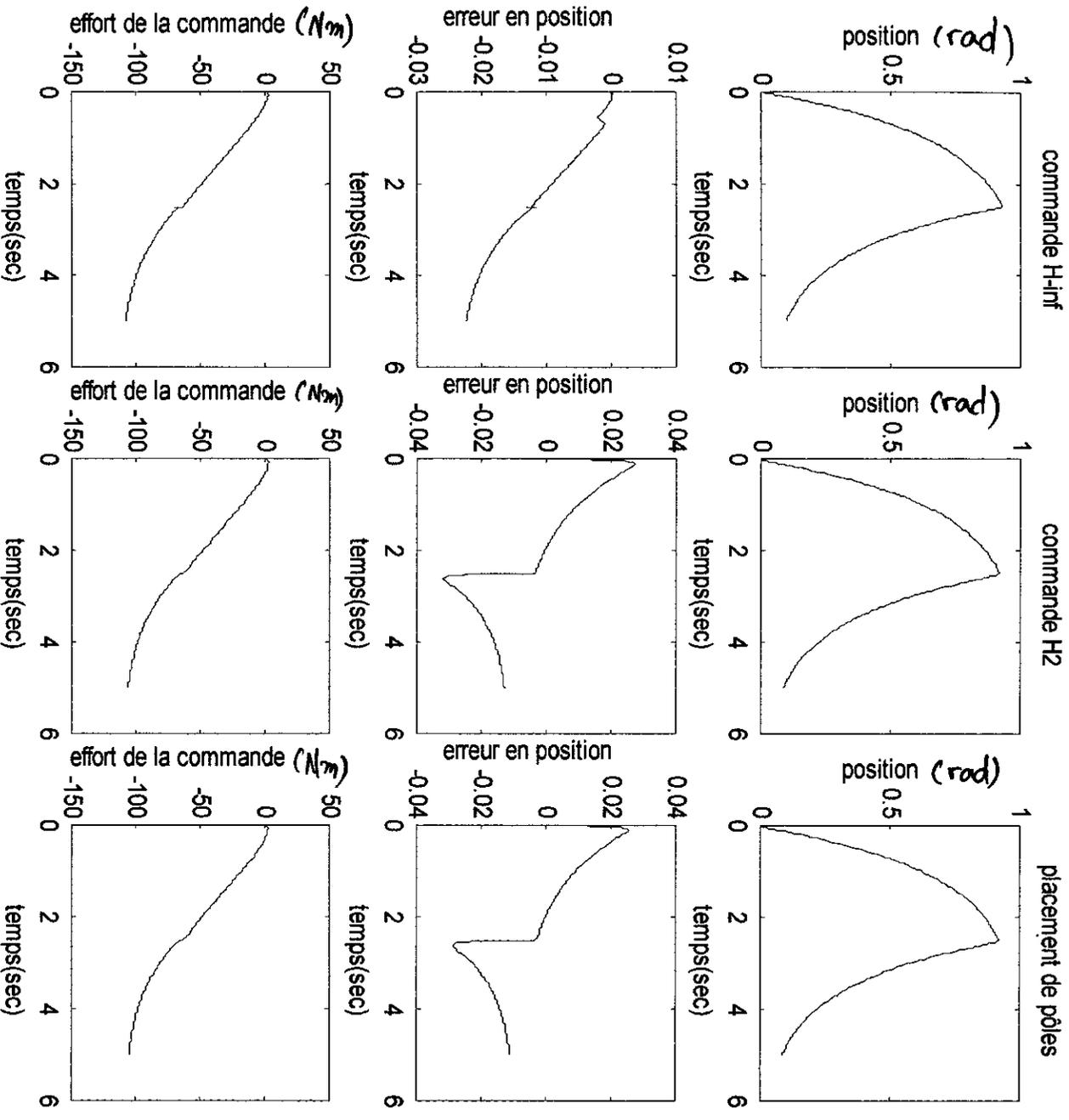
Figure(IV.22) poursuite d'une trajectoire en exponentiel

Deuxième articulation :



Figure(IV.23) poursuite d'une trajectoire en exponentiel

Troisième articulation :



Figure(IV.24) poursuite d'une trajectoire en exponentiel

IV.9.1 Conclusion :

Les simulations ci-dessus montrent une bonne poursuite des trajectoires surtout pour la première articulation. La deuxième et la troisième arrivent à suivre mais avec une erreur plus élevée. En comparant les trois commandes on remarque que c'est la commande H_∞ qui donne les meilleurs résultats d'un point de vue erreur de poursuite et effort développé. Cependant, la forme de la commande est moins lisse malgré l'effet de l'actionneur. En effet, plusieurs fluctuations apparaissent sur le signal de la commande. On constate aussi que l'erreur de poursuite tend à s'accroître avec le temps, on conclut donc que l'articulation a de plus en plus de mal à suivre la consigne surtout que cette dernière varie constamment. C'est vraisemblablement dû à l'inertie du bras manipulateur.

Globalement, on peut dire qu'on a une bonne poursuite des trajectoires et une très bonne robustesse face aux perturbations externes et aux variations brusques de la consigne.

TABLEAU RECAPITULATIF :

	<i>Commande H_∞</i>	<i>Commande H_2</i>	<i>Placement des pôles</i>
<i>Facilité de synthèse et ordre du contrôleur</i>	Nécessite des poids et le contrôleur est du même ordre que le système	Résolution d'une équation de Riccati contrôleur d'ordre zéro	Résolution d'un système d'équations algébriques
<i>Test de Lheavy</i>	Erreur très petite et commande admissible	Erreur petite et commande admissible	Erreur petite et commande admissible
<i>Variation paramétrique</i>	Très bonne robustesse	Bonne robustesse	Bonne robustesse
<i>Décentralisation</i>	Bonne robustesse	Bonne robustesse	Bonne robustesse
<i>Influence de l'actionneur</i>	Lissage des signaux de commande et une erreur très petite	Lissage des signaux de commande et une erreur petite	Lissage des signaux de commande et une erreur petite
<i>Bruit de mesure</i>	Très bon rejet mais des signaux de commandes bruités	Bon rejet mais des signaux de commandes lisses	Bon rejet mais des signaux de commandes lisses
<i>Poursuite d'une trajectoire en exponentiel</i>	Moyenne	Moyenne	Moyenne

CONCLUSION GENERALE

CONCLUSION GENERALE :

Dans ce travail, nous avons synthétisé des contrôleurs décentralisés en se basant sur des systèmes flous. Ces derniers se devaient de trouver des systèmes linéaires approximant le système réel tout en ignorant les interconnexions. En réalité l'effet du couplage apparaît dans les valeurs des paramètres de Sugeno. Nous avons vu que les systèmes flous sont de bons approximateurs, ce qui rend notre système nominal plus fiable. Le défaut majeur de l'adaptation des systèmes flous est celui des minimums locaux. En effet durant l'apprentissage, on ne peut dire que le minimum où on se situe est global. C'est le principal défaut des algorithmes d'adaptation paramétrique.

Dans le deuxième chapitre de notre travail, nous avons synthétisé des contrôleurs H_∞ , ces derniers ont donné de très bons résultats. La poursuite est très bonne, l'erreur statique est négligeable, et les commandes sont admissibles. Les simulations montrent que le manipulateur peut soulever des masses de 5 kg sans perturbation de trajectoire. Les commandes sont robustes vis à vis des variations paramétriques. Nous avons aussi synthétisé un retour d'état optimal au sens H_2 et réalisé un placement des pôles par retour d'état. Ces deux commandes donnent des résultats presque similaires mais moins performants que ceux obtenus avec la commande H_∞ .

Dans le dernier chapitre nous avons soumis le bras manipulateur à divers tests de robustesse. Nous avons constaté que l'actionneur permet d'adoucir les commandes et que les bruits de mesures sont rejetés. Le test de décentralisation a prouvé que les systèmes flous pouvaient éviter la dégradation des performances moyennant un effort supplémentaire dans la commande. Dans le dernier test le manipulateur avait quelques difficultés à suivre une trajectoire en exponentiel du fait de son inertie.

Enfin les points suivants peuvent constituer une suite pour notre travail :

- Résoudre le problème des minimums locaux par des algorithmes d'optimisation plus puissants.
- Comparer la commande H_∞ centralisée et décentralisée d'un point de vue énergétique et robustesse.
- Faire une étude comparative entre la commande H_∞ non linéaire et linéaire.
- Développer une commande par retour d'état adaptative basé sur les systèmes flous.

ANNEXE I

MODELE MATHEMATIQUE DU ROBOT

I.1 INTRODUCTION :

Les modèles dynamiques des bras manipulateurs sont décrits par un ensemble d'équations mathématiques qui portent la dynamique de ceux-ci et peuvent être simulées sur ordinateur dans le but de synthétiser une commande conditionnée par performances désirées. L'ensemble des équations dynamiques peuvent être déterminées par des lois mathématiques classiques Newtoniennes et Lagrangiennes. Les approches d'Euler-Lagrange et Newton-Euler permette aux équations du mouvement des robots.

I.2 APPROCHE D'EULER LAGRANGE [24][31] :

L'approche d'Euler-Lagrange sert à modéliser et présenter la dynamique des robots à travers les équations du mouvement. Elle s'adapte lors des calculs manuels ainsi que pour des calculs assistés par ordinateur. Le formalisme d'Euler-Lagrange et la transformation homogène de Denavit et Hertenberg amène à un algorithme compact pour présenter les équations dynamiques du mouvement.

L'équation d'Euler-Lagrange est :

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_i} \right] - \frac{\partial L}{\partial q_i} + \frac{\partial E_D}{\partial \dot{q}_i} = T_i \quad i = 1, \dots, n \quad (\text{A1.1})$$

Où L est Lagrangien qui s'exprime par :

$$L = E_c - E_p \quad (\text{A1.2})$$

E_C : énergie cinétique totale de toutes les liaisons ;

E_P : énergie potentielle totale de toutes liaisons ;

E_D : énergie de dissipation en cas de présence de frottement visqueux ;

T_i : force généralisée à la $i^{\text{ème}}$ articulation ;

n : nombre de degré de liberté ;

q : coordonnée généralisée d'ordre i ;

\dot{q} : dérivée de la coordonnée généralisée.

• Energie cinétique :

L'énergie cinétique est calculée par l'expression de la vitesse.

$$V_0^i = \frac{dr_0^i}{dt}$$

$$r_0^i = T_0^i r_i^i$$

Où r_i^i est la coordonnée homogène du point (i) exprimée dans le repère R_i . Comme la liaison n'est

pas flexible, on a $\frac{dr_i^i}{dt} = 0$ donc :

$$V_0^i = \sum_{j=0}^i \frac{\partial T_0^i}{\partial q_j} \frac{\partial q_j}{dt} r_i^i \quad (\text{A1.3})$$

Avec T_{i-1}^i est la matrice de transformation homogène. Sous une forme plus compacte, on peut écrire :

$$V_0^i = \sum_{j=0}^i [U_{ij} \dot{q}_j] r_i^i \quad (\text{AI.4})$$

et

$$U_{ij} = \begin{cases} T_0^{j-1} Q_j T_{j-1}^i & j \leq i \\ 0 & j \geq i \end{cases} \quad (\text{AI.5})$$

pour une liaison rotative on a :

$$Q_j = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{AI.6})$$

et pour une liaison translationnelle :

$$Q_j = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{AI.7})$$

l'énergie cinétique de l'élément i dans la liaison i est

$$dE_{Ci} = \frac{1}{2} \text{trace} \left(\sum_{j=1}^i \sum_{k=1}^i U_{ij} (r_i^i r_i^{iT} dm) U_{ik}^T \dot{q}_j \dot{q}_k \right) \quad (\text{AI.8})$$

l'énergie cinétique de la liaison i est :

$$E_{Ci} = \frac{1}{2} \text{trace} \left(\sum_{j=1}^i \sum_{k=1}^i U_{ij} J_i U_{ik}^T \dot{q}_j \dot{q}_k \right) \quad (\text{AI.9})$$

L'énergie cinétique des actionneurs est définie par :

$$E_{Ca} = \frac{1}{2} \sum_{i=1}^n I_i \dot{q}_i^2 \quad (\text{AI.10})$$

Où I_i caractérise un moment d'inertie est définie dans le cas d'une rotation et une masse dans le cas d'une translation de l'actionneur i . L'énergie cinétique totale sera :

$$E_C = \sum_{i=1}^n E_{Ci} + E_{Ca} \quad (\text{AI.11})$$

L'énergie potentielle est décrite par [33] :

$$E_p = \sum_{i=1}^n -m_i g^T r_0^i = -\sum_{i=1}^n m_i g^T T_0^i r_i^i \quad (\text{AI.12})$$

$$g^T = [0 \quad 0 \quad -|g| \quad 1]$$

Où g est la gravité.

Enfin, l'énergie de dissipation est donnée par :

$$E_D = \frac{1}{2} \sum_{i=1}^n f_{vi} \dot{q}_i^2 \quad (\text{AI.13})$$

Où f_{vi} le coefficient de frottement visqueux ($i=1, n$). l'expression finale du Lagrangien sera :

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \text{trace}(U_{ij} J_i U_{ik}^T) \dot{q}_j \dot{q}_k + \sum_i m_i g^T T_0^i r_i^i \quad (\text{AI.14})$$

Appliquant la formule d'Euler-Lagrange à la fonction Lagrangienne donnée ci-dessus, on trouve la force ou couple généralisé.

$$T_i = \sum_{j=1}^n \sum_{k=1}^j \text{trace}(U_{jk} J_j U_{ji}^T) \ddot{q}_k + \sum_{j=1}^n \sum_{k=1}^j \sum_{l=1}^j \text{trace}(U_{jkl} J_l U_{ji}^T) \dot{q}_k \dot{q}_l - \sum_{j=1}^n m_j g^T U_{ji} r_j^j + f_{vi} \dot{q}_i \quad (\text{AI.15})$$

avec

$$U_{ijk} = \begin{cases} T_0^{k-1} Q_k T_{k-1}^{j-1} Q_j T_{j-1}^i & k \leq j \leq i \\ T_0^{j-1} Q_j T_{j-1}^{k-1} Q_k T_{k-1}^i & j \leq k \leq i \\ 0 & j < i < k \end{cases} \quad (\text{AI.16})$$

L.2.1 Méthode Directe1[28], [31] :

Cette méthode est basée essentiellement sur l'équation (AI.15) qui est extraite par la méthode d'Euler-Lagrange, sa forme est :

$$T_i = \sum_{j=1}^n M_{ij}(q) \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n N_{ijk}(q) \dot{q}_j \dot{q}_k + G_i(q) + H_i(\dot{q}_i) \quad (\text{AI.17})$$

avec

$$M_{ij}(q) = \sum_{k=\max(i,j)}^n \text{trace}(U_{kj} J_k U_{ki}^T) \quad \begin{matrix} i=1, \dots, n \\ j=1, \dots, n \end{matrix} \quad (\text{AI.18})$$

$$G_i(q) = -\sum_{j=1}^n m_j g^T U_{ji} r_j^j \quad (\text{AI.19})$$

$$N_{ijk}(q) = \sum_{l=\max(i,j,k)}^n \text{trace}(U_{ijk} J_l U_{li}^T) \quad (\text{AI.20})$$

$$H_i(\dot{q}_i) = f_{vi} \dot{q}_i \quad (\text{AI.21})$$

on peut réécrire l'équation dynamique sous forme matricielle qui sera :

$$T(t) = M(q)\ddot{q} + N(q, \dot{q}) + G(q) + H(\dot{q}) \quad (\text{AI.22})$$

avec $q \in R^n$, $\dot{q} \in R^n$, $\ddot{q} \in R^n$ représentent respectivement les positions, les vitesses et les accélérations articulaires et :

$M(q) \in R^{n \times n}$: Matrice symétrique définie positive des accélérations inertielles dont l'élément M_{ij} de cette matrice est l'inertie de la $i^{\text{ème}}$ articulation sur la $j^{\text{ème}}$ articulation et vice versa ;

$N(q, \dot{q}) \in R^n$: vecteur de forces et /ou couples dus aux accélérations de coriolis et centrifuge ;

$G(q) \in R^n$: vecteur de forces et /ou couples dus aux forces de gravitation ;

$H(q) \in R^n$: représente les frottements visqueux ;

$T(t) \in R^n$: vecteur de forces et /ou couples moteurs ;

1.2.2 Méthode Directe 2 [33] :

La méthode directe citée auparavant est efficace mais elle est longue lors de la détermination du vecteur due aux accélérations de coriolis et centrifuge. Pour cela on a trouvé une issue pour le calcul rapide de ce dernier. Le modèle dynamique est calculé en 3 étapes.

- 1) Calcul des éléments de la matrice d'inertie par la méthode directe 1.
- 2) Calcul de deux matrices centrifuges et coriolis qui sont multipliées par les vecteurs. La somme des vecteurs obtenue donne le vecteur voulu N. l'extrait de ces dernières matrices se fait par la dérivation de la matrice d'inertie (*principe de la conservation d'énergie*).
- 3) Enfin, le calcul des vecteurs gravités et visqueux, par la méthode directe se basant sur les équations (AI.19) et (AI.21) respectivement. La forme appropriée du modèle final sera :

$$T(t) = M(q)\ddot{q} + D(q)[\dot{q} \quad \dot{q}] + C(q)[\dot{q}^2] + G(q) + H(\dot{q}) \quad (\text{AI.23})$$

Où

$D(q) \in R^{n \times \frac{n(n-1)}{2}}$: matrice couple de coriolis ;

$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \\ \dot{q} \dot{q} \end{bmatrix} \in R^{\frac{n(n-1)}{2}}$: vecteur de produit des vitesses généralisées ;

$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \\ \dot{q}^2 \end{bmatrix} \in R^n$: vecteur carré des vitesse généralisées ;

Le calcul de la matrice de coriolis D se base essentiellement sur la formule suivante : [1]

$$D_{ij} = 2\beta^{i,j}$$

D'où le symbole de Christoffel défini par :

$$\beta^{i,k} = \frac{1}{2} \left[\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i} \right] \quad (\text{AI.24})$$

puisque la matrice d'inertie est définie positive, on aura les propriétés suivantes :

$$\frac{\partial M_{ij}}{\partial q_k} = \frac{\partial M_{ji}}{\partial q_k} \quad \forall i, j, k \quad (\text{AI.25})$$

et

$$\frac{\partial M_{ij}}{\partial q_k} = 0 \quad \text{pour } i \geq k, \quad j \geq k \quad (\text{AI.26})$$

Les éléments de la matrice centrifuge sont définis par l'équation suivante : [1]

$$C_{ij} = \beta^{i,j} \quad (\text{AI.27})$$

I.3 MODELE DYNAMIQUE GENERAL [9] :

Le modèle dynamique générale d'un robot à n degrés de liberté peut être représenté par un système d'équations différentielles non linéaires d'ordre deux à n entrées formant le vecteur de force ou couples généralisés T , et n sorties qui forment le vecteur de position q . les équations de ce système à n liaisons, décrites dans l'espace des coordonnées articulaires, sont données sous forme matricielle comme suit : [9]

$$M(q)\ddot{q} + B(q, \dot{q})\dot{q} + K(q, \dot{q})q + G(q) + H(\dot{q}) = T_p(t) + T(t) \quad (\text{AI.28})$$

Où

$T_p(t)$: le vecteur de force ou couples de perturbations externes ;

$T(t)$: le vecteur de force ou couples ou moteurs ;

Les éléments M , B , K , G et H sont généralement des fonctions très compliquées et non linéaires par rapport aux coordonnées généralisées du manipulateur. Le précédent modèle du robot est complexe mais il vérifie certaines propriétés fondamentales qui peuvent être exploitées pour l'analyse du comportement du système et le calcul de la commande.

Propriété 1 :

La matrice $M(q)$ est symétrique définie positive, par conséquent tous les éléments diagonaux de cette matrice sont positifs. $\{ M_{ii}(q) > 0 \quad i=1, \dots, n \}$.

Propriété 2 :

Les matrices M , B , K et les vecteurs G , H sont uniformément bornés.

Propriété 3 :

L'entrée de commande est indépendante pour chaque articulation du manipulateur.

Propriété 4 :

Le vecteur des frottement visqueux et secs H est caractérisé par les n éléments $\left\{ H_i(\dot{q}_i) \quad i=1, \dots, n \right\}$ tel que :

$$H_i(\dot{q}_i) = f_{vi} \dot{q}_i + f_{si} \operatorname{sgn}(\dot{q}_i) \quad (\text{AI.29})$$

avec : f_{vi} et f_{si} sont respectivement les coefficients de frottement visqueux et secs de la $i^{\text{ème}}$ articulation.

Les propriétés (1-4) découlent de la nature physique du robot manipulateur. La propriété 3 est due au fait que les flexibilités des articulation et des structures n'ont pas été prises en compte. Dans ce cas de figure chaque degré de liberté est piloté par un actionneur (moteur à courant continu).

I.4 MODELE DYNAMIQUE DU ROBOT PUMA 560 [1], [28] :

Le calcul du modèle dynamique du robot PUMA 560 est disponible dans plusieurs références citées auparavant, nous avons épargné au lecteur tous les calculs intermédiaires, il pourra toutefois se référer [26] pour plus de détails. Ce manipulateur réalise trois mouvements rotationnels, le premier suivant l'axe horizontal, le second et le troisième suivant deux axes verticaux. Le modèle dynamique est le suivant :

$$T = (M(q) + \Delta M(q))\ddot{q} + N(q, \dot{q}) + G(q) + T_{m_0} \quad (\text{AI.30})$$

avec

$$M(q) = \begin{bmatrix} I_1 + I_2 c_{23}^2 + I_3 c_2^2 + I_4 c_2 c_{23} & I_5 s_{23} + I_6 s_2 & I_5 s_{23} \\ I_5 s_{23} + I_6 s_2 & I_7 + I_4 c_3 & I_8 + 0.5 I_4 c_3 \\ I_5 s_{23} & I_8 + 0.5 I_4 c_3 & I_{18} \end{bmatrix} \quad (\text{AI.31})$$

$$N(q, \dot{q}) = \begin{bmatrix} -(2(I_3 s_2 c_2 + I_2 c_{23} s_{23}) + I_4 (c_2 s_{23} + s_2 c_{23})) \dot{q}_1 \dot{q}_2 - (2I_2 s_{23} c_{23} + I_4 c_2 s_{23}) \dot{q}_1 \dot{q}_2 \\ + (I_6 c_2 + I_5 c_{23}) \dot{q}_2^2 + 2I_5 c_{23} \dot{q}_2 \dot{q}_3 + I_5 c_{23} \dot{q}_3^2 \\ ((I_3 s_2 c_2 + I_2 c_{23} s_{23}) + 0.5 I_4 (c_2 s_{23} + s_2 c_{23})) \dot{q}_1^2 - I_4 s_3 \dot{q}_2 \dot{q}_3 - 0.5 I_4 s_3 \dot{q}_3^2 \\ (I_2 s_{23} c_{23} + 0.5 I_4 c_2 c_{23}) \dot{q}_1^2 + 0.5 I_4 s_3 \dot{q}_2^2 \end{bmatrix} \quad (\text{AI.32})$$

$$G(q) = \begin{bmatrix} 0 \\ -(m_3 l_2 + 0.5 m_2 l_2) g c_2 - 0.5 m_3 l_3 g c_{23} \\ -0.5 m_3 l_3 g c_{23} \end{bmatrix} \quad (\text{AI.33})$$

où

$$\begin{aligned} c_i &= \cos(q_i) & c_{ij} &= \cos(q_{ij}) \\ s_i &= \sin(q_i) & s_{ij} &= \sin(q_{ij}) \end{aligned}$$

et

$$\begin{aligned} I_1 &= I_{yy1} + I_{xx2} + m_2 d_2 (d_2 + e) + m_3 d_2^2 + I_{xx3} \\ I_2 &= I_{yy3} - I_{xx3} \\ I_3 &= I_{yy2} - I_{xx2} + m_3 l_2^2 \\ I_4 &= m_3 l_2 l_3 \\ I_5 &= 0.5 m_3 l_3 d_2 \\ I_6 &= 0.5 m_2 l_2 (d_2 + e) + m_3 d_2 l_2 \\ I_7 &= I_{zz2} + I_{zz3} + m_3 l_2^2 \\ I_8 &= I_{zz3} \end{aligned} \quad (\text{AI.34})$$

La matrice ΔM représente les effets de l'effecteur car ce dernier possède une masse non négligeable. Pour le de cette matrice on se contente de la méthode donnée par[1] :

$$\begin{aligned} \Delta M_{11} &= I_{xxt} + m_t d_{22} + I_{M1} + (I_{yyt} - I_{xxt} + m_t l_{32}) c_{23}^2 + m_t l_2^2 c_2^2 + 2 m_t l_2 l_3 c_2 c_{23} \\ \Delta M_{12} &= m_t l_3 d_2 s_{23} + m_t d_2 l_2 s_2 \\ \Delta M_{13} &= m_t l_3 d_2 s_{23} \\ \Delta M_{22} &= I_{zzt} + m_t (l_2^2 + l_3^2) + I_{M2} + 2 m_t l_2 l_3 c_3 \\ \Delta M_{23} &= I_{zzt} + m_t l_3^2 + m_t l_2 l_3 c_3 \\ \Delta M_{33} &= I_{zzt} + m_t l_{32} + I_{M3} \end{aligned} \quad (\text{AI.35})$$

I_{mi} : moment d'inertie des différents moteurs.

$I_{xxt}, I_{yyt}, I_{zzt}$: moments d'inertie totale par rapport aux principaux axes de l'effecteur.

Le vecteur des couples additifs T_{m0} représente l'effet de la charge, il est calculé par la matrice jacobienne, cette dernière est la dérivée du vecteur position de l'effecteur.

$$j_j(q) = \frac{\partial p}{\partial q_j} \quad (\text{AI.36})$$

en particulier pour le robot PUMA 560, on a :

$$j(q) = [j_1(q) \quad j_2(q) \quad j_3(q)] \quad (\text{AI.37})$$

et

$$p = [c_1(l_2c + l_3c_{23}) - d_2s_1 \quad s_1(l_2c_2 + l_3c_{23}) + d_2c_1 \quad -(l_2s_2 + l_3s_{23})]^T \quad (\text{AI.38})$$

d'où la matrice jacobienne :

$$j(q) = \begin{bmatrix} -s_1(l_2s_2 + l_3c_{23}) - d_2c_1 & -c_1(l_2s_2 + l_3s_{23}) & -c_1l_3s_{23} \\ c_1(l_2c_2 + l_3c_{23}) - d_2s_1 & -s_1(l_2s_2 + l_3s_{23}) & -s_1l_3s_{23} \\ 0 & -(l_2c_2 + l_3c_{23}) & -l_3c_{23} \end{bmatrix} \quad (\text{AI.39})$$

le couple du à la porté de la charge sera :

$$T_{m0} = m_0 J^T(q) \left[J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} + g \right] \quad (\text{AI.40})$$

avec $g = [0 \quad 0 \quad 9.81]^T$ et $\dot{J}(q, \dot{q})$ dérivée de la matrice jacobienne obtenue à partir de la différentiation par rapport au temps. Les paramètres réels de ce robot sont [26] :

masses des différentes liaisons :

$$m_2 = 17.40 \text{ kg} \quad m_3 = 5.04 \text{ kg} \quad m_4 = 0.82 \text{ kg}$$

$$m_5 = 0.35 \text{ kg} \quad m_6 = 0.09 \text{ kg} \quad m_t = m_4 + m_5 + m_6 = 1.26 \text{ kg}$$

paramètres géométriques :

$$d_2 = 149.09 \text{ mm} \quad l_2 = 431.8 \text{ mm} \quad l_3 = 433.07 \text{ mm}$$

paramètres d'inertie :

N ^o de liaison	$I_{xxi} [10^3 \text{ kg m}^2]$	$I_{yyi} [10^3 \text{ kg m}^2]$	$I_{zxi} [10^3 \text{ kg m}^2]$	$I_{Mi} [\text{kg m}^2]$
1	-	350	-	1.14
2	130	524	539	4.71
3	192	15.4	212	0.83
4	1.30	1.80	1.80	-
5	0.30	0.30	0.40	-
6	0.04	0.15	0.15	-
4+5+6	1.64	2.25	2.35	-

Tableau (A.1) paramètres d'inertie pour le robot PUMA 560

ANNEXE II

CALCUL DE LA NORME H_2

Gramien de commandabilité et d'observabilité :

Le calcul de la norme H_2 d'une matrice de transfert peut se faire de deux manières différentes. L'une des deux méthodes nécessite l'introduction des gramiens de commandabilité et d'observabilité d'un système donné. Pour cela, nous commencerons par introduire ces deux notions avant de donner tout détail sur le calcul de la norme H_2 d'une matrice.

Nous considérons un système $G(s)$ strictement stable dont une représentation d'état est donnée par $[A, B, C, D]$.

Le gramien de commandabilité, noté P , est défini par :

$$P = \int_0^{\infty} e^{At} B B^T e^{At} dt \quad (\text{AII.1})$$

et le gramien d'observabilité, noté Q , est donné par :

$$Q = \int_0^{\infty} e^{At} C C^T e^{At} dt \quad (\text{AII.2})$$

P et Q sont deux matrices symétriques semi-définies positives et solutions des équations de Lyapunov :

$$AP + PA^T + BB^T = 0 \quad (\text{AII.3})$$

$$A^T Q + QA + C^T C = 0 \quad (\text{AII.4})$$

Calcul de la norme H_2 :

Soient P et Q les gramiens de commandabilité et d'observabilité du système $G(s)$ stable. Il est aisé de montrer en appliquant le théorème de Parseval [29], le résultat connu suivant :

$$\|G(s)\|_2 = \sqrt{\text{trace}(CPC^T)} = \sqrt{\text{trace}(B^T QB)} \quad (\text{AII.5})$$

La norme H_2 s'obtient donc par résolution d'un système d'équations linéaires suivi d'un calcul d'une trace. Elle est obtenue directement et sans itérations.

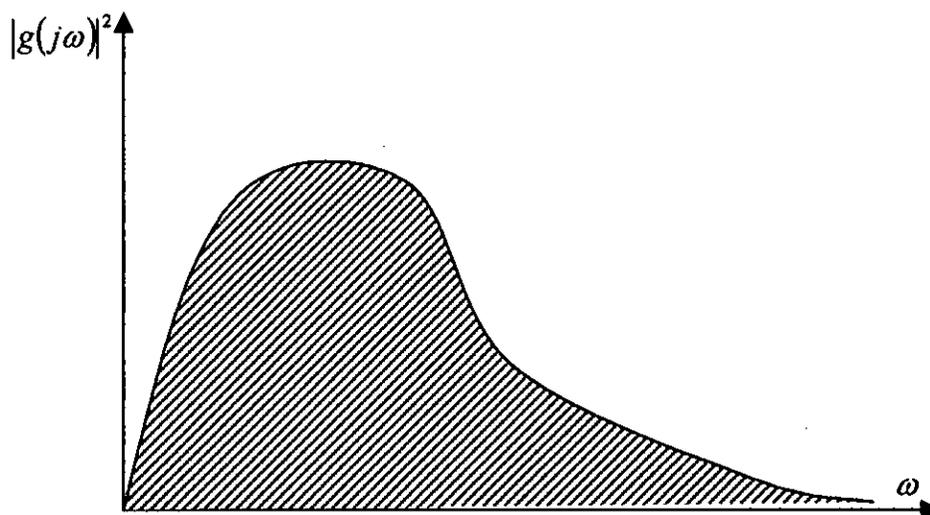
Une autre manière de calculer cette norme et de passer par la définition initiale de cette dernière qui est

$$\|G(j\omega)\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{+\infty} \text{trace}(G^*(j\omega)G(j\omega)) d\omega \right)^{\frac{1}{2}} \quad (\text{AII.6})$$

De cette formule, on déduit aisément l'interprétation physique de la norme H_2 , qui n'est rien d'autre que l'énergie de transmission du signal de sortie lorsque l'entrée est une impulsion de Dirac.

D'un point de vue stochastique, c'est la covariance du signal de sortie lorsque l'entrée est un bruit blanc gaussien normalisé.

Pour un système monovariante, la norme H_2 est la racine carrée de surface présentée ci-dessous :



Figure(AII.1) la norme H_2 pour un système monovariante

Minimiser la norme H_2 c'est minimiser l'énergie de transfert entre les signaux perturbateur et l'erreur à la sortie.

BIBLIOGRAPHIE

ARTICLES :

- [1] AMSTRONG. B, O. KHATIB, J. BURDIK *The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm* Proceeding IEEE pp510-519 1986.
- [2] BABUSKA. R *Fuzzy Systems Modeling and Identification* Departement of Electrical Engineering Control Laboratory 1997.
- [3] FRANCIS. A. B *A Course in H_∞ Theory* Lectures notes in Control and Information Sciences, Vol 88 Springer Verlag , New York 1987.
- [4] FRANCIS. A. B, J. C. DOYLE *Linear Control Theory With an H_∞ Optimality Criterion* SIAM J. Control and Optimization, Vol 25 1987.
- [5] FRANCIS. A. B, G. ZAMES, J. W. HELTON, *H_∞ Optimal Feedback Controllers for Linear Multivariable Systems* IEEE Trans. Automat. Cont. Vol 10, pp 888-900 1984.
- [6] FRANCIS. A. B, G. ZAMES *on H_∞ Optimal Sensitivity Theory for SISO Feedback Systems* IEEE Trans. Automat. Cont. Vol 1, pp 9-16 1984.
- [7] DOYLE. J. C, K. GLOVER *State Space Solutions to Standard H_2 and H_∞ Problems* IEEE Trans. Automat. Control, Vol 34, pp 831-846, 1989.
- [8] JANG. J. S. J, C. T. SUN *Neuro Fuzzy Modeling and Control* Poceeding IEEE Vol83, pp 378-406 1995
- [9] KAIS. Y, Y. P. CHEN *A New Contrller Design for Manipulators Using The Theory of Variable Structure Systems* IEEE Trans. Automt. Cont. Vol 33, 1988.
- [10] KIMURA. H *Robust Stabilizability for A Class of Transfert Functions* IEEE Trans. Automat. Cont. Vol29, pp 788-793 1984.
- [11] LEE. C. C. *Fuzzy Logic in Control Systems : Fuzzy Logic Controller* IEEE Trans. Syst. Cybern. Vol 20, pp 404-435 1990.
- [12] LIN. C. T, C. S. G. LEE *Neural Networks Based Fuzzy Logic Control And Decision Systems* IEEE Trans. Computer. Vol 40, pp 1320-1336 1991.
- [13] MENDEL. M *Fuzzy Systems for Engineering : A Tutorial* Proceeding IEEE Vol 83, pp 347-377 1995.
- [14] Mc FARLANE, K. GLOVER *Robust Controller Design Using Normalized Coprime Factors Plant Description* Lectures notes in Control and Information Sciences, Springer Verlag, New York 1990.
- [15] PERES. D *H_2 Control for Dicrete Time Systems Optimality and Robustness* IEEE Trans. Automat. Control, pp 225-228, 1993.
- [16] ROBOUS. J. A *Predictic Control by Local Linearization of Takagi-Sugeno Fuzzy Model* Control Engineering Laboratory Faculty of Information Technology and Systems 1997.

- [17] SUGENO. M, G. T. KANG *Structure Identification of Fuzzy Models* Fuzzy Sets and Systems Vol 88, pp 116-132 1985.
- [18] SUGENO. M *Special Issue on Fuzzy Control* Fuzzy Sets and Systems Vol 28, pp 149-223 1988.
- [19] TANG. Y, M. TOMIZUKA *Decentralized Robust Control of Mechanical Systems* IEEE Trans. Automat. Control., Vol 45, pp 771-775, 2000.
- [20] WANG. L. X, J. M. MENDEL *Back Propagation Fuzzy Systems as Non-Linear Dynamic Systems Identifiers* Proceeding IEEE int. Conf. On fuzzy systems pp 1409-1418, San Diego C. A. 1992.
- [21] WANG. L. X, J. M. MENDEL *Generating Fuzzy Rules From Numerical Data With Applications* IEEE Syst. Man. Cybern. Vol 22, pp 1414-1427. 1992.
- [22] WONG. K. W, C. S. LIENG, P. F. SAN, L. W. CHAN *On Line Training and Pruning for Recursive Least Square Algorithms* Electronic Letters Vol 32, pp 2152-2153 1996.
- [23] YIH. L, W. WEI YEN *Robust Adaptive Fuzzy Neural Controllers for Uncertain Nonlinear Systems* IEEE Trans. Automat. Control., Vol 15, pp 805-817, 1999

THESES :

- [24] GUENFAF. L *Etude de Différentes Commandes Adaptatives Application à un Robot Manipulateur* Thèse de Magister Ecole Nationale Polytechnique 1995.
- [25] LABIOD. H *Commande Adaptative par les Systèmes Flous Application aux Robot Manipulateurs* Thèse de Magistère Département de Génie Electrique Ecole Nationale Polytechnique 1998.
- [26] MADANNI. *Commande Décentralisée à Structure Variable Application Robotique* Projet de Fin d'Etude Ecole Nationale Polytechnique 1998
- [27] MESSAS. O *Commande H_{∞} Décentralisée Application au Robot PUMA 560* Projet de Fin d'Etude Ecole Nationale Polytechnique 1999
- [28] NEDJARI. M. S, H. BOUKARI *Commande Adaptative Décentralisée Application en Robotique* Projet de Fin d'Etude Ecole Nationale Polytechnique 1996.
- [29] SAMBLANCAT. C *Commande Robuste Multivariable Application à l'Hélicoptère* Thèse de Doctorat Ecole Nationale Supérieure de l'Aéronautique et de l'Espace 1991.

LIVRES :

- [30] BUHLER. H *Conception de Systèmes Automatiques* Presses Polytechnique Romandes 1988.
- [31] BUHLER. H *Réglage par Logique Floue* Presses Polytechnique Romandes 1994.

- [32] DUC. G, A. RACHID *Systèmes de Régulation : Analyse de Robustesse et Synthèse d'une Commande dans le Cadre H_∞* Masson Paris 1994.
- [33] FU. K. S, R. C. GONZALEZ, C. S. G. LEE *Robotic : Control, Sensing, Vision and Intelligence* Mc Graw Hill 1987.
- [34] KOSKO. B *Neural Networks and Fuzzy Systems : A Dynamical Approach* Englewood Cliffs Printice Hall 1992.
- [35] LOWEN. R, M. ROUBENS *Fuzzy Logic State of the Art* Kluwer Academic Publishers 1993.
- [36] PEDRYCZ. W *Fuzzy Control and Fuzzy Systems* Wiley, New York 1993.
- [37] SLOTINE J. J. E, W.LI *Applied Nonlinear Control*. Printice Hall New Jersey 1991.

ملخص:

العمل المقدم في هذه المنكرة هو دراسة للتحكم القوي اللامركزي وتطبيقه على نراع آلي. أولاً قمنا بمقاربة المنظومة الأصلية بثلاث منظومات خطية و غير متداخلة. هذه المنظومات تم تحديدها باستعمال الأنظمة الغامضة لتكاجي سوجينو. ثم استخلصنا متحكمين, الأول في نطاق H_{∞} و الثاني هو مجرد معامل يسمح, في أول الأمر بإعادة التغذية بواسطة متغيرات الحالة بطريقة قياسية, ثم ليتم وضع الأقطاب. في الباب الأخير قمنا بعدة تجارب على هذه المتحكمات لنرى مدى تحملها للاضطرابات الخارجية.

كلمات مفتاح:

المنطق الغامض, القوة, تحكم H_{∞} , تحكم قياسي, تثبيت الأقطاب.

RESUME :

Le travail présenté dans cette thèse est une étude de la commande robuste décentralisée appliquée à la robotique. D'abord nous avons approximé le système réel par trois sous systèmes linéaires et parfaitement découplés. Ces sous systèmes ont été identifiés à l'aide des systèmes flous de Takagi-sugeno. Ensuite, nous avons synthétiser deux contrôleurs. Le premier dans le cadre H_{∞} et le second est un gain permettant, dans un premier temps, un retour d'état optimal (H_2) puis pour effectuer un placement de pôles. Dans le dernier chapitre les contrôleurs ont été soumis à plusieurs tests de robustesse afin d'apprécier leur tolérances face aux perturbations externes.

Mots clefs :

Logique floue, robustesse, commande H_{∞} , commande optimale, placement de pôles.

ABSTRACT :

The work presented in this thesis is a study of decentralized robust control applied to robotics. We first, approached the real system by three perfectly decoupled linear subsystems. These subsystems were identified using Takagi-sugeno fuzzy models. We then, synthesised two controllers. The first one, is an output controller in an H_{∞} frame. The second, is a state feedback. To obtain, we first minimise the H_2 norm then, we use the pole placement theory. In the last chapter, we submit all these controllers to different robustness tests for showing their tolerances to external perturbations.

Keywords:

Fuzzy logic, robustness, H_{∞} control, optimal control, pole placement.