REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTRE DE L'ENSEIGNEMENT SUPERIER

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE CONSTANTINE 1

FACULTE DES SCIENCES DE La TECHNOLOGIE

DEPARTEMENT D'ELECTRONIQUE

N° Ordre :

Série :

MEMOIRE

Présenté pour l'obtention du diplôme de magistère

en électronique

Option : Contrôle des systèmes

Par

Boumaza hamza

THEME

Commande prédictive approximante

SOUTENU LE : 21/11/2013

Devant le jury :

Président : Salim Filali, Professeur, Université de Constantine.

Rapporteur : Khaled Belarbi, Professeur, Université de Constantine.

Examinateurs : Brahim Boutamina, Maître de conférences, Université de Constantine.

Mohamed Chemachema, Maître de conférences, Université de Constantine.

Table des matières

Dédicaces

Je dédie ce Mémoire

A ma famille.

Hamza

Remerciements

Mes remerciements vont en premier lieu à ALLAH Tout Puissant qui a illuminé mon chemin de la lueur du savoir et de la science et pour la volonté, la santé et la patience qu'il ma prodiguées durant toutes ces années d'études.

Je tiens aussi à exprimer ma reconnaissance et ma profonde gratitude à Monsieur BELARBI Khaled, professeur à l'université Mentouri Constantine, pour avoir assuré l'encadrement de ce travail. Son aide, sa grande disponibilité ont joué un rôle essentiel dans l'aboutissement de ce travail.

J'exprime mes vifs remerciements à Monsieur FILALI Salim, professeur à l'université Mentouri Constantine, qui m'a fait l'honneur de présider ce jury.

Mes sincères remerciements vont également à Messieurs BOUTAMINA Brahim et CHEMACHEMA Mohamed, Maitres de Conférences à l'université Mentouri Constantine qui ont bien voulu examiner ce mémoire.

Enfin, mes remerciements vont aussi à mes parents pour leur patience, leurs encouragements continus et leur soutien inconditionnel. Qu'il trouve ici toute ma gratitude et mon amour.

Que toutes les personnes que j'ai involontairement oubliées, trouvent ici, en cette heureuse et solennelle circonstance, l'expression de notre profonde gratitude et de notre indéfectible dévouement.

APC Approximate Predective Control

CPE Commande Prédictive Explicite

DMC Dynamic Matrix Control

MPC Model Predective Control

MBPC Model Based Predective Control

PQM Programmation Quadratique Multiparamétrique

PSO Particle Swarm Optimization

PWA Piecewise Affine

Chapitre 1

INTRODUCTION

La commande prédictive (MPC pour Model Predictive Control) est devenue de plus en plus populaire ces dernières années dans le milieu industriel pour sa tolérance envers différents types de systèmes et le respect des contraintes imposées, ainsi que la compatibilité avec le matériel. L'idée principale de la commande prédictive est basée sur l'utilisation d'un modèle du système à commander pour prédire sa sortie sur un certain horizon, l'élaboration d'une séquence optimale de commandes futures satisfaisant les contraintes et minimisant une fonction coût, l'application du premier élément de la séquence optimale sur le système et la répétition de la procédure complète à la prochaine période d'échantillonnage [1]. C'est le principe de l'horizon fuyant. MBPC implique alors la résolution d'un problème d'optimisation de dimension finie à chaque période d'échantillonnage. Historiquement, il est généralement admis que les idées de base de la MPC ont été introduites par Richalet en 1976 [2] suivi un peu plus tard par la Dynamic Matrix Control, DMC de Cutler et Ramaker [3]. Cependant, il est admis que la popularité de la MPC dans sa forme actuelle, est due à Clarke et al [4] avec la commande prédictive généralisée.

L'inconvénient principal de la commande MPC est le temps d'exécution, de ce fait la méthode trouve son application beaucoup plus dans le domaine des procédés chimiques qui sont en général des systèmes lents à grande période d'échantillonnage où l'application des méthodes numériques ne pose pas de problème. Pour les systèmes rapides échantillonnés à haute fréquence tels que les moteurs, robots…, la solution numérique en ligne du problème d'optimisation peut être impraticable. Aussi, il est utile de rechercher des solutions rapides et efficaces. Plusieurs chercheurs se sont alors penchés sur ce sujet pour trouver une solution à cet obstacle. Ainsi, la technique dite commande prédictive explicite (*explicit* MPC) a été introduite au cours de la dernière décennie [5,6]. La technique repose sur le calcul de la loi de commande de la MPC hors ligne et la convertit en loi linéaire par morceau (PWA), ce qui réduit le problème en ligne à une évaluation d'une table de consultation (lookup table) de gains d'une commande linéaire. Malgré le succès qu'elle a eu dans certaines applications il y a toujours des limitations concernant le nombre d'ensemble PWA [6].

Les nouvelles techniques introduites récemment sont basées sur une approximation de la commande explicite. Dans la commande prédictive approximative, MPCA, dans un premier stade on calcule hors ligne la loi de commande explicite de la MPC et ensuite on développe une approximation pour cette loi. Cette approximation est ensuite implantée en ligne. Plusieurs méthodes d'approximation ont été utilisées [7-14].

Dans ce travail, nous considérons une commande prédictive approximative floue où la loi de commande prédictive optimale est approximée par un système d'inférence floue de type Takagi Sugeno. La loi de commande prédictive optimale, obtenue hors ligne par simulation sur un modèle du système, est utilisée pour construire une base de données qui servira pour apprentissage du système flou. L'apprentissage est réalisé par une optimisation qui utilise l'algorithme d'optimisation par essaim de particules, Particle Swarm Optimization, PSO. La motivation de la méthode repose sur la capacité d'approximation universelle des systèmes flous [15].

Ce mémoire est organisé comme suit :

Dans le chapitre 2, nous consacrons la première partie à la définition de quelques concepts de la commande prédictive tels que son principe de fonctionnement et ses éléments. Ensuite, nous parlons du concept de la commande prédictive explicite. Finalement, nous développons le cas de la commande prédictive non linéaire.

Dans le chapitre 3, nous introduisons le concept de la commande prédictive approximative par un système flou de Takagi Sugeno. Ensuite, nous parlons du principe général de la commande prédictive approximative, APC, Approximate Predictive Control. Finalement, nous présentons la stratégie de commande prédictive approximative floue ainsi que la synthèse du contrôleur flou.

Dans le chapitre 4, nous analysons par les simulations les performances du contrôleur flou étudié dans le Chapitre 3. Quatre exemples sont étudiés ; deux linéaires et deux autres non linéaires.

Chapitre 2

COMMANDE PREDICTIVE

2.1. Introduction

2.2. Généralités sur la commande prédictive

2.2.1. Principe générale de la commande prédictive

2.2.2. Eléments de la commande prédictive

2.2.2.1. Modèle de prédiction

2.2.2.2. Fonction coût (critère de performance)

2.2.2.3. Loi de commande

2.2.3. Commande prédictive linéaire stable avec contraintes

2.2.4. Commande prédictive explicite

2.3. Commande prédictive non linéaire

2.1. Introduction :

Dans ce chapitre, nous introduisons quelques concepts de la commande prédictive tels que son principe de fonctionnement et ses éléments. Ensuite, nous parlons du concept de la commande prédictive explicite. Finalement, nous développons le cas de la commande prédictive non linéaire.

2.2. Généralités sur la commande prédictive :

2.2.1. Principe générale de la commande prédictive:

La principale idée de la commande prédictive se résume à « utiliser un modèle pour prédire le comportement du système et choisir la meilleure décision au sens d'un certain coût tout en respectant les contraintes ».

Le principe de la commande prédictive est illustré dans la Figure (2.1), à chaque période d'échantillonnage du contrôleur un calcul des prédictions des variables contrôlées est effectué jusqu'à un horizon de temps N (horizon de prédiction) grâce au modèle interne et la future loi de commande calculée, à appliquer sur les variables manipulées jusqu'à un horizon temporel N_u (horizon de commande), en minimisant un critère de performances. A la période d'échantillonnage suivante, seul le premier élément (en rouge) de la loi de commande calculée est appliqué sur le système. Cette procédure est ensuite répétée: c'est le principe de l'horizon fuyant. Ainsi à chaque période d'échantillonnage, un problème d'optimisation doit être résolu en temps réel. Dans le cas linéaire ce problème est exprimé sous forme d'un programme quadratique qui admet donc un seul minimum global tandis que dans le cas non linéaire, c'est un programme non linéaire avec contraintes non convexe admettant plusieurs minima locaux.

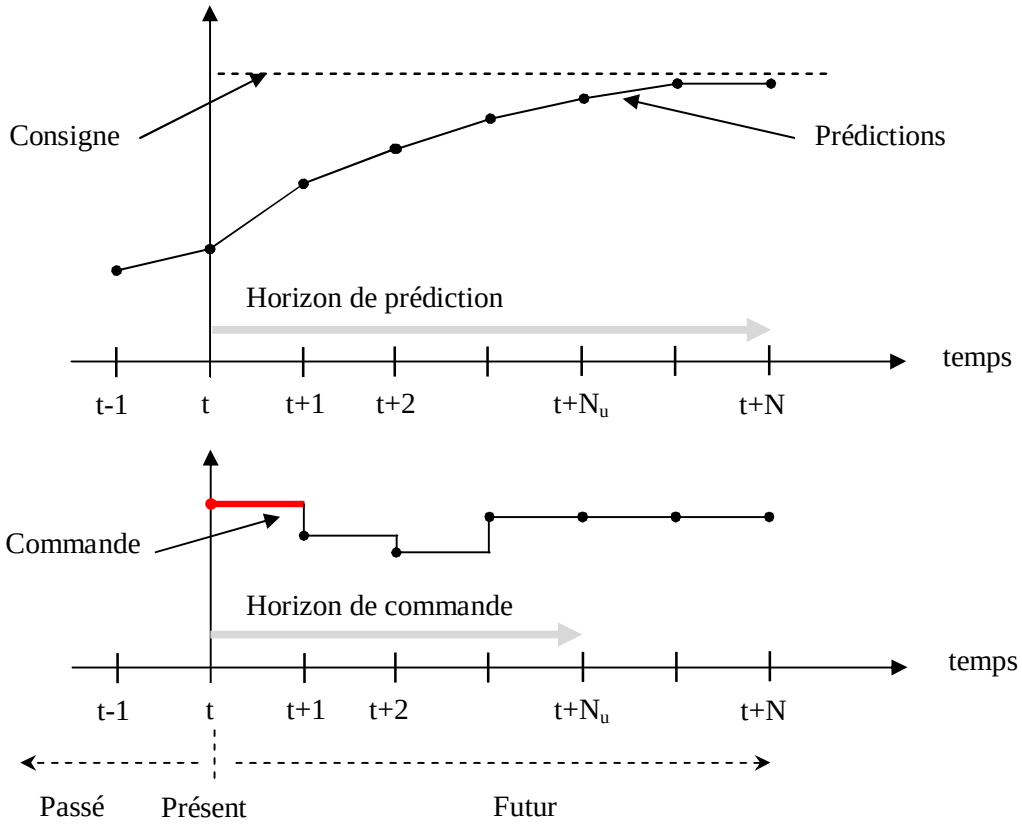


Figure (2.1) Principe de la commande prédictive.

2.2.2. Eléments de la commande prédictive :

Les éléments de base de la commande prédictive (voir Figure (2.2)) sont :

1. Un modèle pour réaliser les prédictions.
2. Une fonction coût à minimiser plus les contraintes.
3. Un algorithme d'optimisation (pour calculer la commande future).

Différentes options peuvent être considérées pour chaque élément, ce qui donne une variété d'algorithmes de commande prédictive.

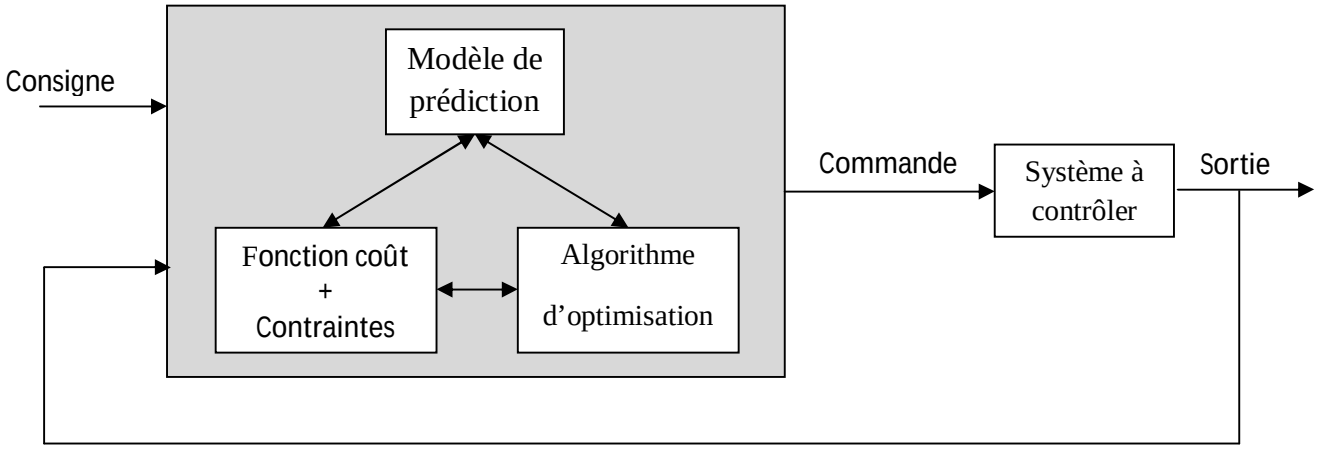


Figure (2.2) Stratégie de commande prédictive.

2.2.2.1. Modèle de prédiction

Le modèle de prédiction se compose de deux parties, la première décrit la relation entrées sorties et l'autre les perturbations et les erreurs de modélisation. Le modèle doit être discret car la commande prédictive est une commande numérique.

Selon le modèle, il existe plusieurs formes de commande prédictive :

- Commande prédictive linéaire à base de modèle d'état, Fonction de transfert … etc.
- Commande prédictive non linéaire à base de modèle d'état non linéaire … etc.

Prédictions à base de Modèle d'état :

Modèle d'état discret :

$$\begin{cases} \mathbf{x}(t+1) = A\,\mathbf{x}(t) + B\,\mathbf{u}(t) \\ \mathbf{y}(t) = C\,\mathbf{x}(t) \end{cases} \tag{2.1}$$

Où $\mathbf{x} \in \mathbb{R}^n$ est le vecteur d'états appartenant à un espace d'état $M \subset \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ le vecteur d'entrées et $\mathbf{y} \in \mathbb{R}^l$ le vecteur de sorties du système.

On calcule les *n* prédictions en développant les équations d'état :

$$\mathbf{x}(t+1) = A\,\mathbf{x}(t) + B\,\mathbf{u}(t) = A\,\mathbf{x}(t) + B\,\mathbf{u}(t\text{-}1) + B\,\Delta\mathbf{u}(t)$$

$$\mathbf{x}(t+2) = A\,\mathbf{x}(t+1) + B\,\mathbf{u}(t+1)$$

$$= A^2\,\mathbf{x}(t) + (AB + B)\,\mathbf{u}(t\text{-}1) + (AB + B)\,\Delta\mathbf{u}(t) + B\,\Delta\mathbf{u}(t+1)$$

$$\vdots$$

$$\mathbf{x}(t+p) = A^p\,\mathbf{x}(t) + (A^{p-1}B + \cdots + B)\,\mathbf{u}(t\text{-}1) + (A^{p-1}B + \cdots + B)\,\Delta\mathbf{u}(t) + \ldots + B\,\Delta\mathbf{u}(t+p\text{-}1)$$

D'où le prédicteur optimale sous forme matricielle :

$$\widehat{\mathbf{X}} = \begin{bmatrix} \hat{\mathbf{x}}(t) \\ \hat{\mathbf{x}}(t+1) \\ \vdots \\ \hat{\mathbf{x}}(t+p-1) \end{bmatrix} = \Psi_0 \mathbf{x}(t) + \Gamma_0 \mathbf{u}(t-1) + \Lambda\, \Delta\mathbf{U} \tag{2.2}$$

Où $\Delta\mathbf{U} = \begin{bmatrix} \Delta\mathbf{u}(t) \\ \Delta\mathbf{u}(t+1) \\ \vdots \\ \Delta\mathbf{u}(t+p-1) \end{bmatrix}$, $\Psi_0 = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^p \end{bmatrix}$, $\Gamma_0 = \begin{bmatrix} B \\ AB + B \\ \vdots \\ A^{p-1}B + \cdots + B \end{bmatrix}$,

$$\Lambda = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB + B & B & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{p-1}B + \cdots + B & A^{p-2}B + \cdots + B & \cdots & B \end{bmatrix}$$

Ou encore :

$$\hat{\mathbf{y}} = C\, \hat{\mathbf{x}} = \Psi_{\mathbf{y}} \mathbf{x}(t) + \Gamma_{\mathbf{y}} \mathbf{u}(t-1) + \Lambda_{\mathbf{y}} \Delta\mathbf{U} \tag{2.3}$$

avec $\Psi_{\mathbf{y}} = C\, \Psi_0,\ \Gamma_{\mathbf{y}} = C\, \Gamma_0,\ \Lambda_{\mathbf{y}} = C\, \Lambda$

2.2.2.2. Fonction coût (critère de performance) :

La fonction coût J pénalise les écarts entre les sorties prédites commandées **y**(t+k|t) et la trajectoire de référence **r**(t+k|t). La trajectoire de référence peut dépendre de mesures effectuées à l'instant k, en particulier, son point de départ peut être la mesure de la sortie **y**(t) ; ainsi elle peut être un ensemble de point fixe ou autre trajectoire prédéterminée. On définit la fonction coût suivante :

$$J = \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{r}_k\|_Q^2 + \sum_{k=0}^{N_u - 1} \|\mathbf{u}_k\|_R^2 \tag{2.4}$$

Où N est l'horizon de prédiction et Nu l'horizon de commande, on assume que $N_u \leq N$, et que $\Delta u(t + k|t) = 0$ pour $k \geq N_u$, donc $u(t + k|t) = u(t + Nu - 1|t)$ pour tout $k \geq N_u$.

La forme de l'équation (2.4) implique que l'erreur entre la sortie et la référence est pénalisée à chaque instant situé dans l'horizon de prédiction (1<=k<=N). C'est le cas le plus fréquent dans la commande prédictive. Pour le cas de la régulation à l'origine la référence r est nulle.

2.2.2.3. Loi de commande :

Pour obtenir la loi de commande, on remplace le prédicteur (2.2) dans la fonction coût :

$$\begin{aligned} J &= \left\|\widehat{\mathbf{X}}\right\|_Q^2 + \|\Delta\mathbf{U}\|_R^2 \\ &= \|\Psi_0 \mathbf{x}(t) + \Gamma_0 \mathbf{u}(t-1) + \Lambda\, \Delta\mathbf{U}\|_Q^2 + \|\Delta\mathbf{U}\|_R^2 \end{aligned} \tag{2.5}$$

En prenant le gradient de J par rapport au commande $\Delta\mathbf{U}$ **:**

$$\nabla_{\Delta \mathbf{U}} J = 0 \tag{2.6}$$

On aura :

$$\Delta\mathbf{U} = \frac{\mathbf{1}}{\mathbf{2}}\mathrm{H}^{-\mathbf{1}}\mathrm{G} \tag{2.8}$$

Où $\mathrm{H} = \Lambda^{\mathrm{t}}\mathrm{Q}\Lambda + \mathrm{R},\ \mathrm{G} = -2\Lambda^{\mathrm{t}}\mathrm{Q}(\Psi_0\mathbf{x}(t) + \Gamma_0\mathbf{u}(t-1))$

On applique seulement la première composante de $\Delta\mathbf{U}$ au système :

$$\mathbf{u}(t) = \mathbf{u}(t-1) + \Delta\mathbf{u}(\mathrm{t}) \tag{2.9}$$

2.2.3. Commande prédictive linéaire stable avec contraintes :

Dans la commande prédictive linéaire avec contraintes, on considère le problème de régulation à l'origine du système dont le modèle est donné sous forme d'équations d'état :

$$\begin{aligned} \mathbf{x}(t+1) &= A\,\mathbf{x}(\mathrm{t}) + B\,\mathbf{u}(\mathrm{t}) \\ \mathbf{y}(\mathrm{t}) &= C\,\mathbf{x}(\mathrm{t}) \end{aligned} \tag{2.10}$$

Où $\mathbf{x} \in \mathbb{R}^n$ est le vecteur d'états, $\mathbf{u} \in \mathbb{R}^m$ le vecteur d'entrées , $\mathbf{y} \in \mathbb{R}^l$ le vecteur de sorties du système, $A \in \mathbb{R}^{n*n}$, $B \in \mathbb{R}^{n*m}$ et (A, B) est une paire gouvernable.

Etant donné le vecteur d'état $\mathbf{x}(\mathrm{t})$ *obtenu à l'instant t* par mesure ou observation, la commande $\mathbf{u}(\mathrm{t})$ à appliquer au système à l'instant t+1 est obtenue en solvant le problème d'optimisation statique suivant pour $U = [\,\mathbf{u}(t)^T, \dots, \mathbf{u}(t+M-1)^T]$

$$\min_{\mathbf{u}}\{J(U, \mathbf{x}(t)) = \mathbf{x}_{t+N/t}^T P \mathbf{x}_{t+N/t} + \sum_{k=0}^{N-1} \mathbf{x}_{t+k/t}^T Q \mathbf{x}_{t+k/t} + \mathbf{u}_{t+k}^T R \mathbf{u}_{t+k}\} \tag{2.11}$$

Avec les contraintes:

$$\mathbf{y}_{min} \le \mathbf{y}(t+k) \le \mathbf{y}_{max}, \quad k = 1, \dots, N$$

$$\mathbf{x}_{min} \le \mathbf{x}(t+k) \le \mathbf{x}_{max}, \quad k = 1, \dots, N$$

$$\mathbf{u}_{min} \le \mathbf{u}(t+k) \le \mathbf{u}_{max}, \quad k = 1, \dots M-1$$

Avec N est l'horizon de prédiction sur l'état, M est l'horizon de commande, les matrices symétriques $Q \ge 0$ (semi définie positive) et $R > 0$ (définie positive), sont la matrice de pondération sur l'état et la matrice de pondération d'entrée, respectivement et P est une matrice de pondération sur l'état final. Pour garantir la stabilité du système commandé la matrice P est choisie comme la solution de l'équation algébrique de Riccati :

$$PA + A^t P - PBR^{-1}B^t P + Q = 0 \tag{2.12}$$

En effet, il a été démontré [16,17] qu'il est possible de transformer le problème LQRinfini en un problème à horizon fini si le coût est considéré comme la somme de deux composantes :

$$\sum_{k=0}^{\infty}(\mathbf{x}_k^t Q\mathbf{x}_k + \mathbf{u}_k^t R\mathbf{u}_k) = \sum_{k=0}^{N-1}(\mathbf{x}_k^t Q\mathbf{x}_k + \mathbf{u}_k^t R\mathbf{u}_k) + \sum_{k=N}^{\infty}(\mathbf{x}_k^t Q\mathbf{x}_k + \mathbf{u}_k^t R\mathbf{u}_k) \quad (2.13)$$

Où $N < \infty$ est l'horizon de prédiction choisi. Il est à noter qu'après un certain temps les contraintes sont satisfaites de façon naturelle (dans l'horizon de prédiction), les entrées considérées à l'optimisation sont uniquement celles de la première composante du coût, ce qui nous mène à introduire la notion du coût final.

$$\sum_{k=0}^{\infty}(\mathbf{x}_k^t Q\mathbf{x}_k + \mathbf{u}_k^t R\mathbf{u}_k) = \sum_{k=0}^{N-1}(\mathbf{x}_k^t Q\mathbf{x}_k + \mathbf{u}_k^t R\mathbf{u}_k) + \mathbf{x}_N^t P\mathbf{x}_N \quad (2.14)$$

Où $\mathbf{x}_N^t P\mathbf{x}_N$ *est la fonction coût final.*

2.2.4. Commande prédictive explicite:

La commande prédictive explicite (CPE) est une implémentation de la commande prédictive basée sur la programmation quadratique multiparamétrique (PQM). Cette dernière est une technique d'optimisation qui permet de déterminer la solution optimale d'un problème d'optimisation comme étant une fonction explicite de certains paramètres variables. Par conséquent, la PQM ne nécessite pas la résolution d'un nouveau problème d'optimisation quand les paramètres changent, car la solution optimale peut être facilement mise à jour en utilisant la fonction pré-calculée.

Dans le contexte de la commande prédictive, la PQM peut être utilisée pour obtenir la commande optimale comme étant une fonction explicite des mesures, considérées comme les paramètres du problème d'optimisation. De cette manière, le calcul en ligne est réduit à une évaluation d'une série de fonction au lieu de l'optimisation en temps réel.

La solution explicite optimale est un ensemble de fonctions des variables d'état du système par morceaux, où le domaine d'intérêt est un sous ensemble de l'espace d'état qui est divisé en un nombre fini de régions critiques. Pour chaque région critique, une loi de commande par retour d'état donne la valeur de l'entrée (commande). L'ensemble de ces lois de commande (fonctions par morceaux) forme la solution optimale explicite.

Cependant, l'implémentation en ligne de la solution explicite nécessite la sauvegarde de l'ensemble des fonctions par morceaux et, pour chaque mesure, de trouver la région critique qui contient le vecteur d'état actuel et d'évaluer la loi de commande par retour d'état correspondante.

Le Tableau 2.1 donne un aperçu des avantages et des inconvénients de la commande prédictive explicite.

Avantages	Inconvénients
- solution optimale explicite calculée hors ligne, ce qui donne une mise en œuvre en matériel simple comme une puce (peut être reproduite à bon marché pour la production de masse). - La forme explicite de la solution permet son utilisation sur des systèmes rapides, puisque l'évaluation de la fonction est généralement une opération très rapide (comparé à résoudre un problème d'optimisation). - une compréhension intuitive précise et profonde des comportements de contrôle, ce qui permet l'analyse des performances telles que les vérifications de sécurité.	- Obtenir la solution optimale explicite revient à résoudre (hors ligne) un problème d'optimisation paramétrique, qui est en général une tâche difficile. - l'effort de calcul hors ligne se développe rapidement quand la taille du problème augmente (long horizon de prédiction, grand nombre de contraintes, système multi variables) - la complexité de l'optimisation grandit, la complexité de la solution explicite également se développe généralement en termes de nombre de lois de commande formant la fonction par morceaux.

Tableau 2.1 Avantages et Inconvénients de la commande prédictive explicite.

2.3. Commande prédictive non linéaire:

Considérons le modèle non linaire continu suivant:

$$\begin{cases} \dot{\boldsymbol{x}} = f(\mathbf{x}) + g(\mathbf{x})\,\mathbf{u} \\ \mathbf{y} = h(\mathbf{x}) \end{cases} \tag{2.15}$$

Où $\mathbf{x} \in \mathbb{R}^n$ est le vecteur d'états appartenant à un espace d'état $M \subset \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ le vecteur d'entrées et $\mathbf{y} \in \mathbb{R}^l$ le vecteur de sorties du système. On suppose qu'il existe un point d'équilibre $\mathbf{x_0} \in M$ avec $f(\mathbf{x_0}) = 0$. De plus, on suppose que le système est contrôlable, observable, et que tous les états sont mesurables. Les perturbations et les incertitudes du modèle ne sont pas tenues en compte.

Afin de simplifier, le système (2.13) peut être formulé sous la forme linaire à dépendance d'état :

$$\begin{cases} \dot{\mathbf{x}} = A(\mathbf{x})\,\mathbf{x} + B(\mathbf{x})\,\mathbf{u} \\ \mathbf{y} = C(\mathbf{x})\,\mathbf{x} \end{cases} \tag{2.16}$$

Où $f(\mathbf{x}) = A(\mathbf{x})\,\mathbf{x}$, $g(\mathbf{x}) = B(\mathbf{x})$ *et* $h(\mathbf{x}) = C(\mathbf{x})\,\mathbf{x}$. Dans ce qui suit on note l'évolution dans le temps du vecteur d'état et de la loi de commande, respectivement, par $\mathbf{x_t}$ et $\mathbf{u_t}$, en ce qui concerne les prédictions le vecteur $\mathbf{x}_k$ représente le vecteur d'état et $\mathbf{u}_k$ la future commande.

La discrétisation est assurée par la technique d'Euler avec un pas δt :

$$\begin{cases} \mathbf{x_{t+1}} = \mathbf{x_t} + \delta t \times (A(\mathbf{x_t})\,\mathbf{x_t} + B(\mathbf{x_t})\,\mathbf{u_t}) \\ \mathbf{y_t} = C(\mathbf{x_t})\,\mathbf{x_t} \end{cases} \tag{2.17.a}$$

Ou par la méthode de Range Kutta :

$$\begin{cases} \mathbf{x}_{\mathbf{t},\frac{1}{2}} = \mathbf{x_t} + \frac{\delta t}{2} \times F(\mathbf{x_t}) \\ \breve{\mathbf{x}}_{\mathbf{t},\frac{1}{2}} = \mathbf{x_t} + \frac{\delta t}{2} \times F\left(\mathbf{x}_{\mathbf{t},\frac{1}{2}}\right) \\ \mathbf{x}_{\mathbf{t},1} = \mathbf{x_t} + \delta t \times F\left(\breve{\mathbf{x}}_{\mathbf{t},\frac{1}{2}}\right) \\ \mathbf{x_{t+1}} = \mathbf{x_t} + \frac{\delta t}{6} \times \left(F(\mathbf{x_t}) + \mathbf{2}\left(F\left(\mathbf{x}_{\mathbf{t},\frac{1}{2}}\right) + F\left(\breve{\mathbf{x}}_{\mathbf{t},\frac{1}{2}}\right)\right) + F(\mathbf{x}_{\mathbf{t},1})\right) \\ F(\mathbf{x}) = A(\mathbf{x})\mathbf{x} + B(\mathbf{x})\mathbf{u} \\ \mathbf{y_t} = C(\mathbf{x_t})\,\mathbf{x_t} \end{cases} \tag{2.17.b}$$

Considérant le problème de régulation du système discret (2.15.a) ou (2.15.b) à l'origine. L'objectif est de chercher la loi de commande optimale en minimisant le critère à horizon fini suivant:

$$J = \sum_{k=0}^{N-1}(\mathbf{x}_k^t Q(\mathbf{x}_k)\mathbf{x}_k + \mathbf{u}_k^t R(\mathbf{x}_k)\mathbf{u}_k) + \mathbf{x}_N^t P(\mathbf{x}_k)\mathbf{x}_N \tag{2.18}$$

Où $\mathbf{x}_k$ est le vecteur d'état prédit à l'instant t+k en appliquant les k premier élément du vecteur d'entrée $\mathbf{u}_k$ ($\mathbf{u}_0 \cdots \mathbf{u}_N$) au système (2.15.a) ou (2.15.b), en prenant $\mathbf{x}_0 = \mathbf{x}_t$. Avec les matrices symétriques $Q(\mathbf{x}_k) \geq 0$ (semi définie positive) et $R(\mathbf{x}_k) > 0$ (définie positive), sont la matrice de pondération d'état et la matrice de pondération d'entrée, respectivement.

Pour alléger l'écriture on met :

$$\begin{cases} A(\mathbf{x}) = A_{\mathbf{x}} \\ B(\mathbf{x}) = B_{\mathbf{x}} \\ Q(\mathbf{x}) = Q_{\mathbf{x}} \\ R(\mathbf{x}) = R_{\mathbf{x}} \end{cases}$$

Où $\mathbf{x}_N^t P(\mathbf{x})\mathbf{x}_N$ *est la fonction coût final* et $P_{\mathbf{x}} = P(\mathbf{x})$ est la solution de l'équation algébrique de Riccati obtenue par :

$$P_x A_x + A_x^t P_x - P_x B_x R_x^{-1} B_x^t P_x + Q_x = 0 \tag{2.19}$$

Le problème d'optimisation MPC a horizon fini s'écrie comme suit :

$$\min_{\mathrm{u}}\{\mathbf{x}_N^t P_{\mathbf{x}} \mathbf{x}_N + \sum_{k=0}^{N-1}(\mathbf{x}_k^t Q_{\mathbf{x}} \mathbf{x}_k + \mathbf{u}_k^t R_{\mathbf{x}} \mathbf{u}_k)\} \tag{2.20.a}$$

s. à Système (2.a) ou (2.b) avec $\mathbf{x}_0 = \mathbf{x}_t$ (2.20.b)

$$\mathbf{u}_k = -\mathrm{K}(\mathbf{x})\mathbf{x}_k \,, N_u \leq k \leq N-1 \tag{2.20.c}$$

$$\mathbf{x}_k \in E_x \in \mathbb{R}^n,\ 0 \leq k \leq N-1 \tag{2.20.d}$$

$$\mathbf{u}_k \in E_u \in \mathbb{R}^m, 0 \leq k \leq N-1 \tag{2.20.e}$$

$$\mathbf{y}_k \in E_y \in \mathbb{R}^l, 1 \leq k \leq N \tag{2.20.f}$$

Où N est l'horizon de prédiction, N_u est l'horizon de commande, $P_{\mathbf{x}} = P_{\mathbf{x}}^t \geq 0$ est la solution de l'équation de Riccati, $Q_{\mathrm{x}} = Q_{\mathbf{x}}^t \geq 0$ et $R_{\mathbf{x}} = R_{\mathrm{x}}^t \geq 0$ sont des matrices de pondérations, $\mathrm{K}(\mathbf{x})$ matrice gain de la commande, avec les ensembles $E_x, E_u\ et\ E_y$ qui définissent respectivement les contraintes sur les états, les entrées et les sorties.

Chapitre 3

COMMANDE PREDICTIVE APPROXIMANTE PAR APPRENTISSAGE FLOU

3.1. Introduction

3.3. Commande prédictive approximante par apprentissage flou

3.3.1. Principe général de la commande prédictive approximante

3.3.2. Stratégie de la commande prédictive approximante par apprentissage flou

3.3.2. 1. Solution MPC

3.3.2. 2. Synthèse du contrôleur flou

3.1. Introduction

Dans ce chapitre, nous introduisons le concept de la commande prédictive approximante par apprentissage flou. Ensuite, nous parlons du principe général de la commande prédictive approximante. Finalement, nous présentons la stratégie de commande approximante ainsi que la synthèse du contrôleur flou.

Malgré le succès de la commande prédictive et la commande prédictive explicite, l'application de ces méthodes pour les systèmes rapides (temps d'échantillonnage inférieur à 1ms) reste toujours difficile.
Les nouveaux algorithmes introduits récemment dans le cadre de la commande prédictive approximante et ceux de la technique *explicit* MPC partagent le même principe de calcul. Dans un premier temps, hors ligne, on calcule la loi de commande de la MPC et une approximation de cette loi de commande puis, dans un deuxième temps, on synthétise un contrôleur en se basant sur cette approximation pour l'appliquer en ligne.

3.3. Commande prédictive approximante par apprentissage flou

La commande prédictive approximante par apprentissage flou est une technique utilisée pour réduire le temps d'exécution qui est le premier obstacle d'appliquer la commande prédictive aux systèmes rapides.

D'abord la solution MPC est calculée hors ligne en cherchant la solution optimale quelque soit le temps d'exécution, puis cette solution est approximée par un système flou. Le but est de garantir les mêmes performances que la commande optimale ainsi qu'améliorer le temps d'exécution en ligne.

3.3.2. Stratégie de la commande prédictive approximante par apprentissage flou

La commande prédictive approximante passe par deux étapes essentielles :

1- La recherche de la solution MPC hors ligne,

2- La synthèse du contrôleur APC flou.

3.3.2. 1. Solution MPC par PSO:

La technique méthaeuristique d'optimisation « Optimisation par Essaim Particulaire » ou PSO (Particle Swarm Optimization) [18] est adoptée dans ce travail pour la recherche d'une solution au problème 2.18.

Optimisation par essaim particulaire :

L'optimisation par essaim particulaire est une technique d'optimisation qui s'inspire du comportement social de groupes d'oiseaux ou de poissons en quête de nourriture. La technique a été initialement conçue et développée par Eberhart et Kennedy en 1995 [15].
Le système est initialisé par une population de solutions, appelées particules, avec une position et une vitesse aléatoire pour chaque particule dans l'espace de solutions du problème. Chaque itération fait bouger chacune des particules en fonction de sa vitesse actuelle vers sa meilleure solution ***pbest*** et la meilleure solution obtenue dans son voisinage ***gbest***.

Le pseudo code de la technique est comme suit:

Pour toute particule
    Initialisation particule
Fin
Faire
    Pour toute particule
        Calculer la valeur du coût
            Si le coût est mieux que le meilleur coût enregistré
                Sauvegarder la valeur actuelle comme meilleur coût ***pbest***
    Fin
    Choisir la particule avec la meilleure valeur du coût comme étant ***gbest***
    Pour toute particule
        Mise à jour de la vitesse $v_{k+1}^i = wv_k^i + c_1 r_1 (p_k^i - x_k^i) + c_2 r_2 (p_k^g - x_k^i)$
        Mise à jour de la position $x_{k+1}^i = x_k^i + v_{k+1}^i$
    Fin
Tant que le maximum d'itérations n'est pas atteint

Où: - x_k^i : Position de la particule.

- v_k^i : Vitesse de la particule.

- p_k^i : La meilleure position locale.

- p_k^g : La meilleure position globale.

- w : inertie

- c_1, c_2 : Facteurs de corrections

- r_1, r_2 : variables aléatoires entre 0 et 1.

Les paramètres optimaux suggérés par Clerc [19] correspondent à ω = 0.7298 et $c1 = c2$ = 1.49618.

L'organigramme (3.1) illustre le principe général des différentes étapes de l'implémentation de la commande prédictive en utilisant l'optimisation par essaim particulaire.
L'essaim est constitué par un ensemble de particules où chaque particule représente le signal de commande futur. Il est initialisé par une combinaison de valeurs aléatoires, à chaque période d'échantillonnage la fonction coût est initialisée à une grande valeur et le calcul de la loi de commande futur est effectué à travers la minimisation de cette fonction. Au prochain coup d'horloge, seul le premier élément de la loi de commande est appliqué sur le système.

3.3.2. 2. Synthèse du contrôleur flou :

Système flou de type Takagi Sugeno :

Les systèmes d'inférence floue peuvent être considérés comme des systèmes logiques qui utilisent des règles linguistiques pour établir des relations entre leurs variables d'entrée et de sortie. Aujourd'hui, leurs applications sont très nombreuses outre la commande, ils sont largement utilisés pour la commande, la modélisation, le diagnostic et la reconnaissance de formes.

La structure de base d'un système d'inférence flou comprend :

- Interface de fuzzification,
- Base de règles,
- Mécanisme d'inférence floue,
- Interface de défuzzification.

Comme pour tout système flou les systèmes de type Takagi Sugeno passent par une phase de fuzzification en attribuant à chaque donnée appartenant à un certain ensemble flou un degré d'appartenance, sauf que les conclusions de la base de règles sont représentées par des fonctions, la fusion de ces dernières donne la sortie du contrôleur.

Fuzzification :

La fuzzifcation consiste à traduire les valeurs numériques en valeurs floues c.à.d. en degrés d'appartenance aux variables floues.

Le degré d'appartenance est calculé selon une fonction d'appartenance attribuée à chaque variable floue. Le Tableau 3.1 suivant donne quelques exemples des fonctions utilisées.

La Base de règles :

La base de règles est constituée d'un ensemble de phrases du type :

Si *condition*$_1$, *condition*$_2$, … , *condition*$_{Nf}$ **Alors** *conséquence* (3.1)

Cette base peut être construite à partir de la connaissance d'un expert ou par optimisation.

Le mécanisme d'inférence :

Permet d'interpréter dans un calculateur la base de règles.

L'interprétation mathématique de la règle (3.1) est réalisée par :

$$T - norm: [0,1] \rightarrow [0,1]$$

$$\mu_{règle} = T - norm\left(\mu_{condition_1}, \mu_{condition_2}, \cdots, \mu_{condition_{Nf}}\right) \quad (3.2)$$

Où N_f est le nombre total des conditions, $\mu_{règle}$ est le degré d'appartenance de la règle, $\mu_{condition\_i}$ est le degré d'appartenance de la *condition*$_i$ *(i=1,2, …, N*$_f$*).*

La fusion est l'interprétation de l'ensemble des règles. Dans le cas des systèmes flous de type Takagi Sugeno la fusion consiste à déterminer la valeur de sortie par :

$$sortie = \sum_{i=1}^{p} \mu_{règle_i} * conséquence_i \Big/ \sum_{i=1}^{p} \mu_{règle_i} \quad (3.3)$$

Où $\mu_{règle_i}$ est le degré d'appartenance de la *règle*$_i$ (*i=1…p), conséquence*$_i$ sont des fonctions dans les systèmes de type de Takagi Sugeno.

Le schéma ci après (Figure 3.1) donne un aperçu général du principe de fonctionnement du système flou de type Takagi Sugeno.

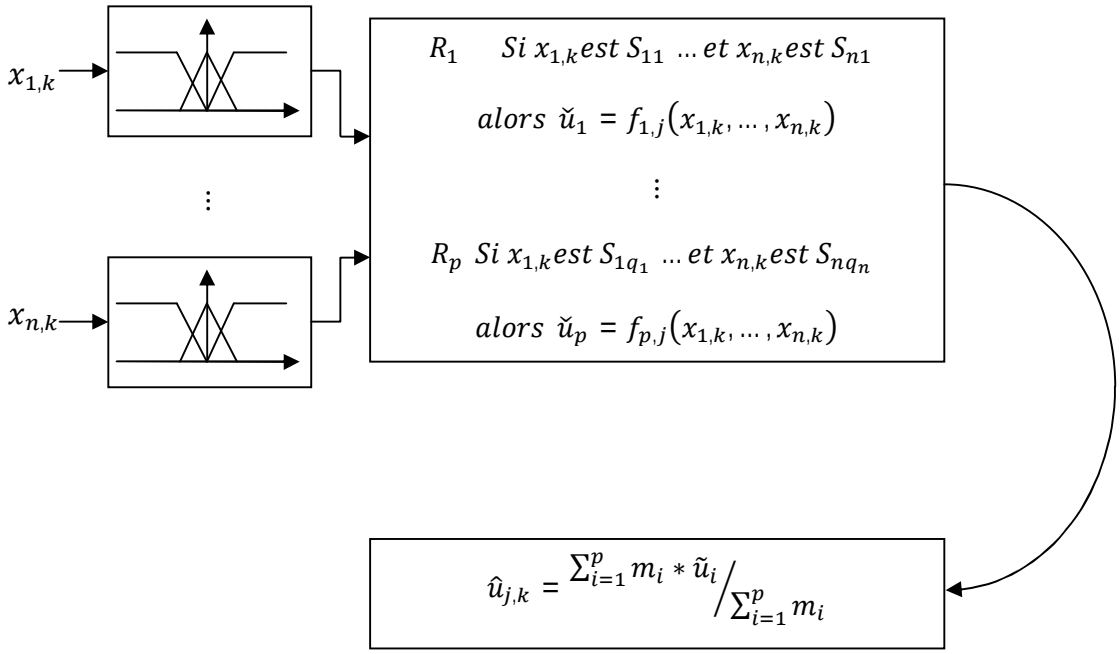


Figure 3.1 Schéma de principe d'un système flou de Takagi Sugeno

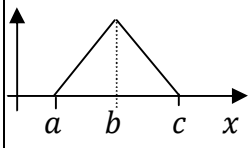
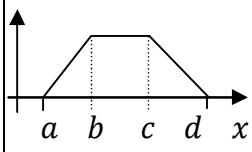
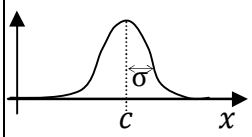
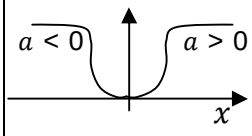
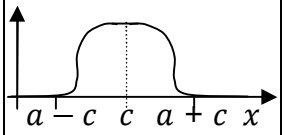
Fonction	Forme	Modèle mathématique
Triangulaire		$F(x;a,b,c) = max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$
Trapézoïdale		$F(x;a,b,c) = max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-b}\right), 0\right)$
Gaussienne		$F(x;\sigma,c) = exp\left(-\left(\frac{x-c}{\sigma}\right)^2\right)$
Sigmoïde		$F(x;a,c) = \frac{1}{1 + exp(-a(x-c))}$
En cloche		$F(x;a,b,c) = \frac{1}{1 + \left\lvert\frac{x-a}{a}\right\rvert^{2b}}$

Tableau 3.1 Formes et Modèles des fonctions d'appartenance les plus utilisées

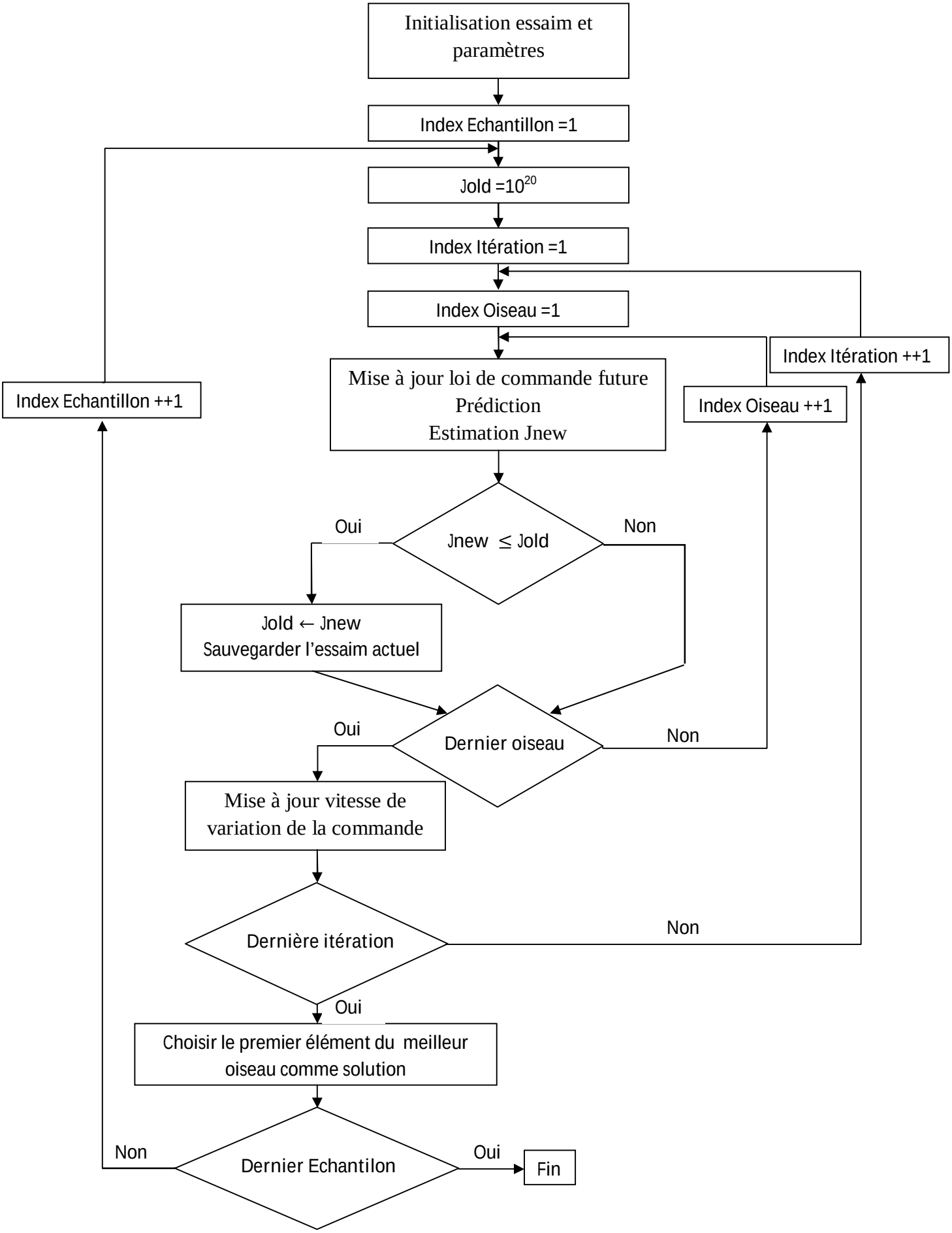


Figure 3.2. Organigramme (3.1) Calcul de la solution MPC en utilisant l'optimisation par Essaim particulaire

Algorithme PSO pour APC

Le calcul des coefficients du système flou de Takagi Sugeno présenté dans le paragraphe précédent est effectué par l'optimisation du problème suivant :

$$\min_{a_{ij}} \left\{ \sum_{k=1}^{N_e} \left(u_{j,k}^* - \hat{u}_{j,k} \right)^2 \right\} \tag{3.4.a}$$

$$\text{s. à } \hat{u}_{j,k} = \sum_{i=1}^{p} m_i * \tilde{u}_i \Big/ \sum_{i=1}^{p} m_i \,, (j = 1 \dots m) \tag{3.4.b}$$

$$\tilde{u}_i = a_{0j} + \sum_{h=1}^{n} a_{hj} \ \mathbf{x}_{hk}, (i = 1 \dots p) \tag{3.4.c}$$

$$\hat{u}_{j,k} \in E_u \in \mathbb{R}^m \tag{3.4.d}$$

Où $u_{j,k}^*$ est la commande MPC calculée hors ligne, $\hat{u}_{j,k}$ est la commande MPC estimée par le système flou de Takagi Sugeno, N_e nombre total d'échantillons, avec *j* indice de l'entrée et *i* indice de la règle.

Le but est de minimiser l'erreur quadratique entre la loi de commande MPC calculée hors ligne et la loi de commande estimée à travers le système flou de Takagi Sugeno.

Calcul des paramètres flous (fonctions f_i) avec PSO :

Pour le calcul des coefficients du systèmes flou Takagi Sugeno la paire $(\mathbf{x}_k^*, \mathbf{u}_k^*)$ de la solution réelle de la MPC est chargée avec les degrés d'appartenance de $\mathbf{x}_k^*$, où l'étape de calcule des coefficients du système flou (étape 3 de l'algorithme suivant) est illustrée par l'organigramme (3.2).

L'essaim est constitué par un ensemble d'oiseaux où chaque oiseau représente les coefficients des fonctions $f_{i,j}(x_{1,k}, \dots, x_{n,k})$ $(i = 1 \dots p)$, il est initialisé par une combinaison de valeurs aléatoires puis à chaque coup d'horloge le calcul des coefficients de Takagi sugeno est effectué à travers la minimisation de fonction coût.

1- charger les pairs $(\mathbf{x}_k^*, \mathbf{u}_k^*)$ avec $k = 1 \dots n$, solution de la MPC

2- calculer les degrés d'appartenance m_i $(i = 1 \dots p)$ pour $\mathbf{x}_k^*$ $(k = 1 \dots n)$,

3- utiliser PSO pour estimer $f_{i,j}$ $(i = 1 \dots p, j = 1 \dots m)$.

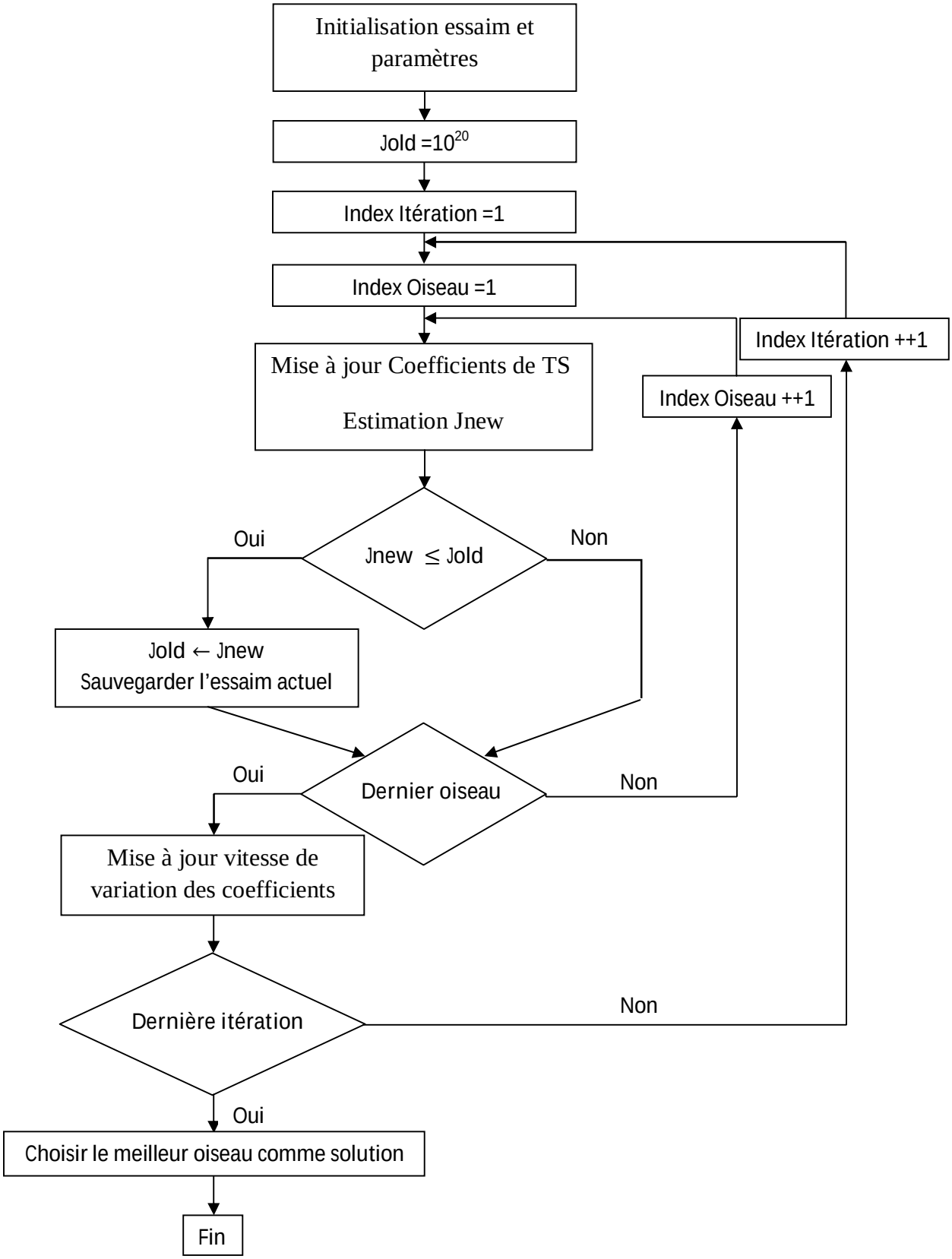


Figure 3.3 Organigramme (3.2) de Calcul des coefficients de Takagi Sugeno en utilisant l'optimisation par Essaim particulaire

Algorithme du contrôleur APC :

L'algorithme suivant donne les étapes du contrôleur APC à apprentissage flou, les fonctions $f_{i,j}$ $(i = 1 \dots p, j = 1 \dots m)$ calculées ultérieurement sont chargées dans le contrôleur puis à chaque coup d'horloge les états sont mesurés et leurs degrés d'appartenance sont calculés. Les étapes 3 et 4 sont consacrées au calcul de la commande à appliquer.

1- charger les $f_{i,j}$ $(i = 1 \dots p, j = 1 \dots m)$, k =1,

2- calculer les degrés d'appartenance m_i $(i = 1 \dots p)$ pour $\mathbf{x}_k$, j =1,

3- calculer $\breve{u}_i = f_{i,j}(x_{1,k}, \dots, x_{n,k})$ $(i = 1 \dots p)$,

4- calculer $\hat{u}_{j,k} = \sum_{i=1}^{p} m_i * \tilde{u}_i \Big/ \sum_{i=1}^{p} m_i$,

5- $j = j + 1$, si $j \leq m$ retour à 3

6- $k = k + 1$, retour à 2.

Chapitre 4

SIMULATIONS

4.1. Introduction

4.2. Cas des systèmes Linéaires

4.2.1. Exemple 1 : Système mono-variable.

4.2.2. Exemple 2 : Système multi-variable.

4.3. Cas des systèmes Non Linéaire

4.3.1. Exemple 1 : Système mono-variable: Pendule inversé.

4.3.2. Exemple 2 : Système multi-variable.

4.1. Introduction

Dans ce chapitre, nous analysons par les simulations les performances du contrôleur flou étudié dans le Chapitre 3. Nous considérons le cas linéaire et non linéaire, pour chaque cas un exemple SISO et un exemple MIMO sont analysés.

4.2. Cas des systèmes Linéaires

4.2.1. Exemple 1 : Système mono-variable

On considère le problème de régulation à l'origine du système discret mono-variable sous amorti :

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0.9 & -0.0861 \\ 0.1772 & 0.9909 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0.0609 \\ 0.0609 \end{bmatrix} u_k$$

$$\mathbf{y}_k = [1 \quad 0]\, \mathbf{x}_k$$

Soumis aux contraintes sur la commande suivantes :

$$-2 < u_k < 2$$

En minimisant le critère (2.6) avec :

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 0.1, N_p = 3, N_c = 2 \text{ et } \mathbf{x}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Paramètres PSO :

Paramètres / Méthode	inertie	Facteurs de corrections	Taille d'essaim	Nombre d'itérations
MPC	1 .4	2	30	100
APC	0.8	2	30	1000

Les figures 4.1 et 4.2 donnent les résultats de l'optimisation de la commande MPC par PSO et de la commande approximée pour le cas nominal, les figures 4.3 et 4.4 représentent les résultats de validation du contrôleur APC pour des conditions initiales autres que celles ayant servies à l'optimisation où on remarque que le contrôleur approximant arrive à réguler le système. Le contrôleur flou a donné une bonne approximation de la solution calculée par MPC, avec une moyenne du temps d'exécution de $1{,}23e^{-4}$s pour APC contre une moyenne de 0,1167s pour MPC, et une erreur en régime permanent de l'ordre de 10^{-5} pour APC et 10^{-6} pour MPC.

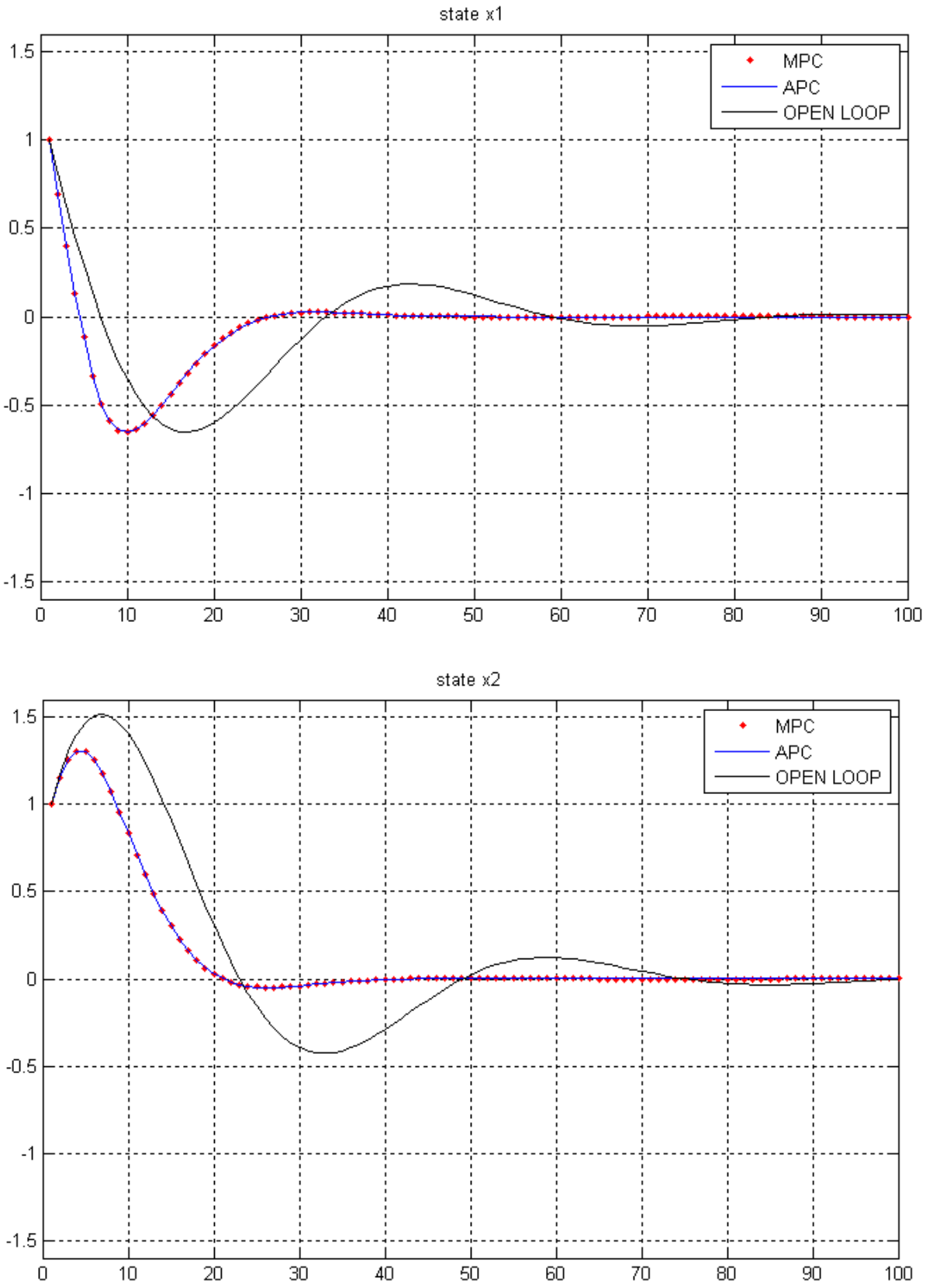


Figure 4.1 : Régulation par MPC et APC : les états du système linéaire mono variable : cas nominal..

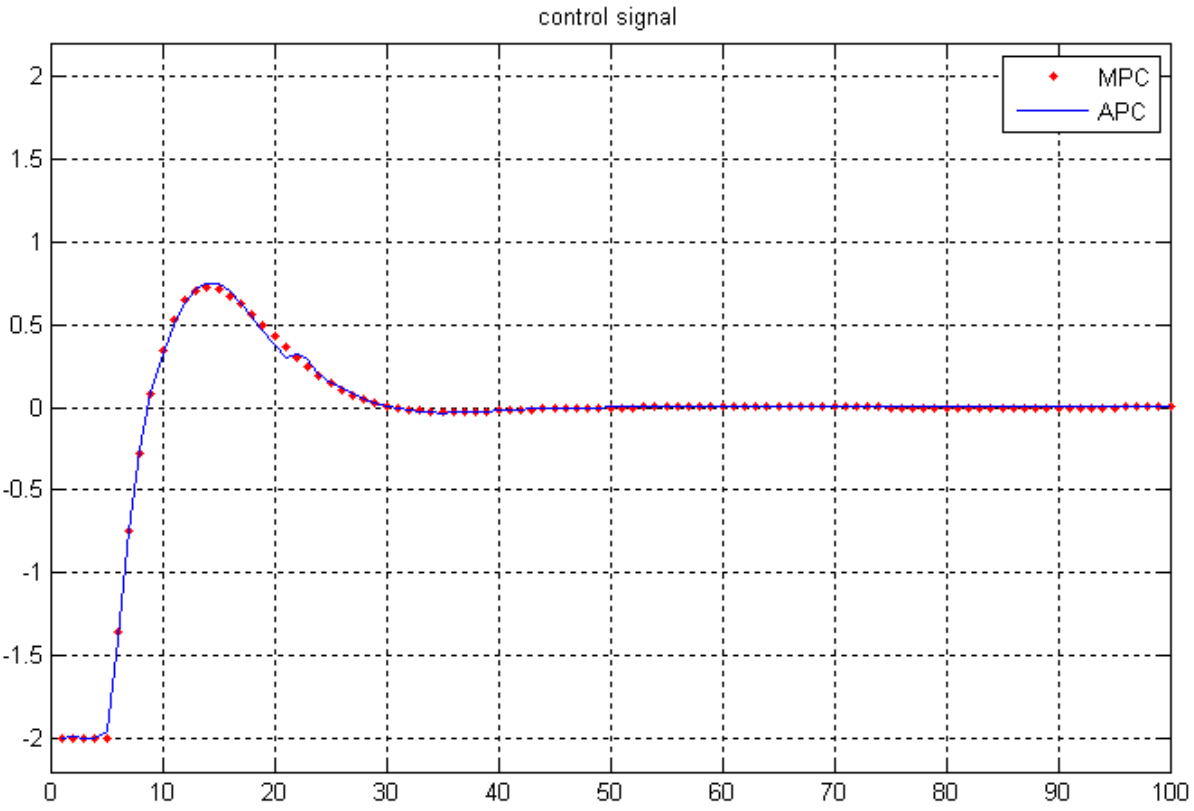


Figure 4.2 : Régulation MPC et APC : Commande du système linéaire mono variable :cas nominal.

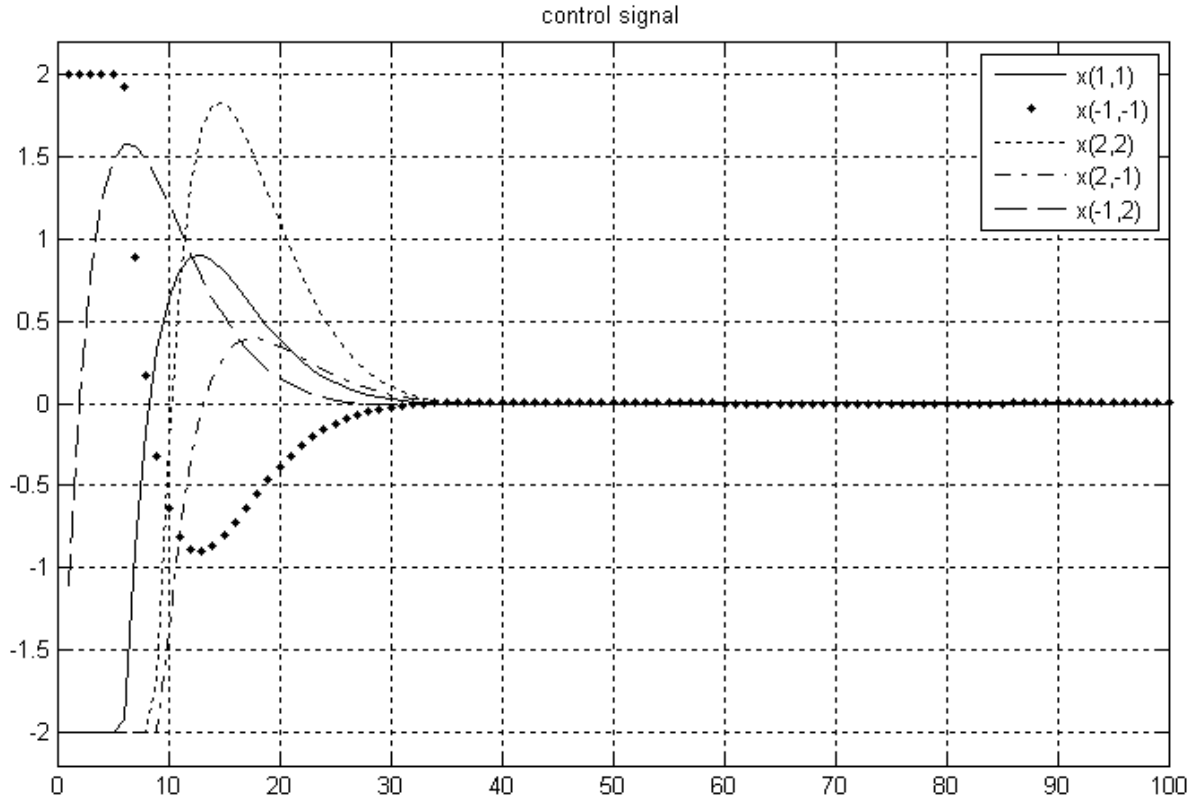


Figure 4.3 : Régulation MPC et APC : la commande du système linéaire mono variable : Validation

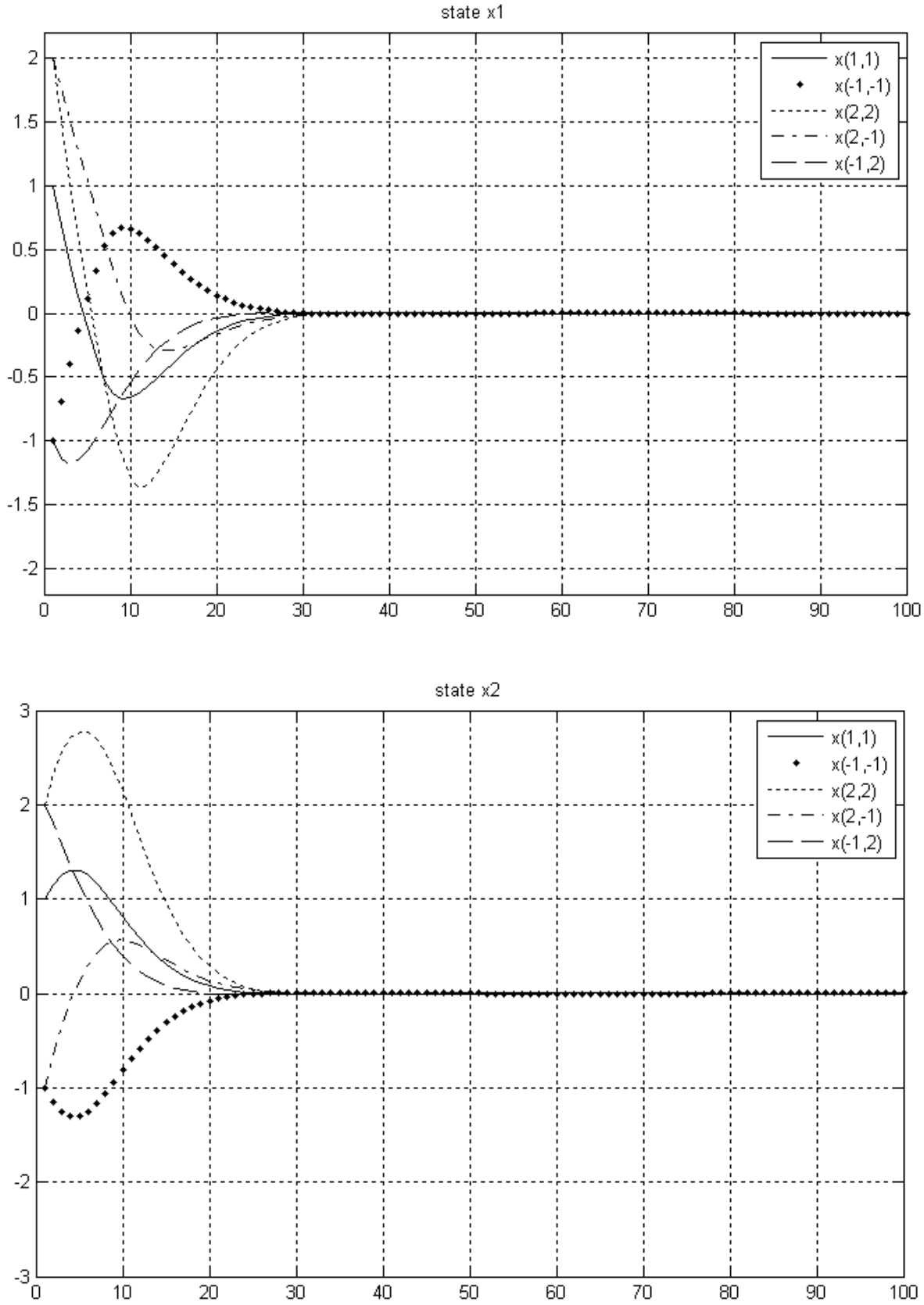


Figure 4.4 : Régulation MPC et APC états du système linéaire mono variable :validation

4.2.2. Système multi-variable:

On considère le problème de régulation à l'origine du système discret multi-variable instable [14] :

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1.2 & 1 \\ 0 & 1.1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{u}_k$$

$$\mathbf{y}_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}_k$$

Soumis aux contraintes suivantes :

Sur les commandes : $-0.5 < u_{k,1} < 0.5\,,\ -0.6 < u_{k,2} < 0.6$

Sur les sorties : $-2 < y_{k,1} < 2\,,\qquad -2 < y_{k,2} < 2$

En minimisant le critère (2.6) avec :

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, N_p = 5, N_c = 4 \text{ et } \mathbf{x}_0 = \begin{pmatrix} -1.3 \\ -1 \end{pmatrix}$$

Paramètres PSO :

Paramètres / Méthode	inertie	Facteurs de corrections	Taille d'essaim	Nombre d'itérations
MPC	1 .4	2	30	100
APC	0.8	2	30	10000

Les figures 4.5, 4.6 et 4.9 donnent les résultats de l'optimisation de la commande MPC par PSO et de la commande approximée obtenue après optimisation par PSO pour le cas nominal. Les figures 4.7 et 4.8 représentent les résultats de validation du contrôleur APC pour des conditions initiales autres que celles ayant servies à l'optimisation où on remarque que le contrôleur approximant arrive à réguler le système. Le contrôleur flou a donné une bonne approximation de la solution calculée par MPC, avec une moyenne du temps d'exécution de $1.2359e^{-4}$s pour APC contre une moyenne de 0.3064s pour MPC et une erreur en régime permanent de l'ordre de 10^{-5} pour APC et 10^{-6} pour MPC.

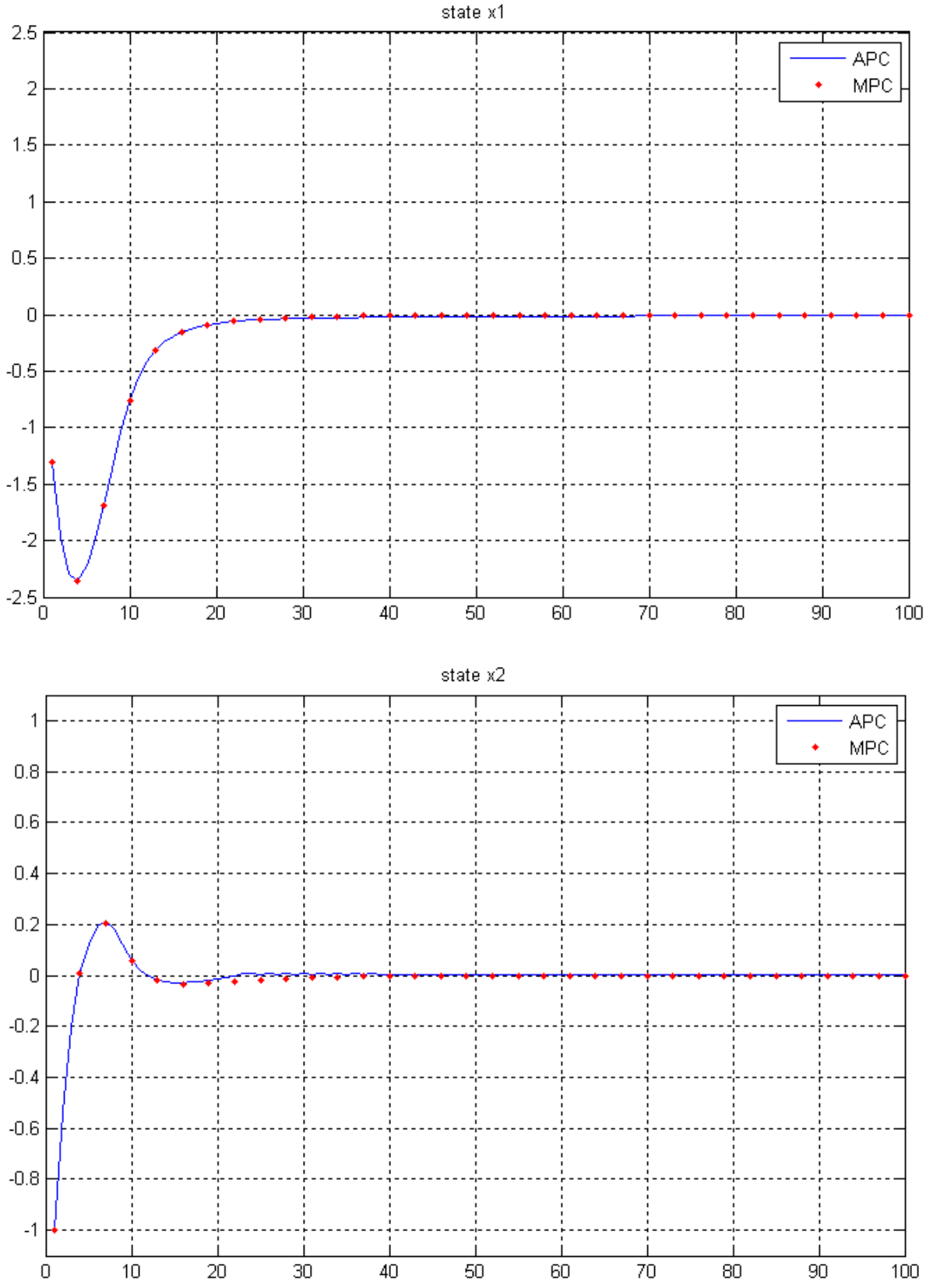


Figure 4.5 : Régulation par MPC et APC
pour les états du système linéaire multi variables : cas nominal

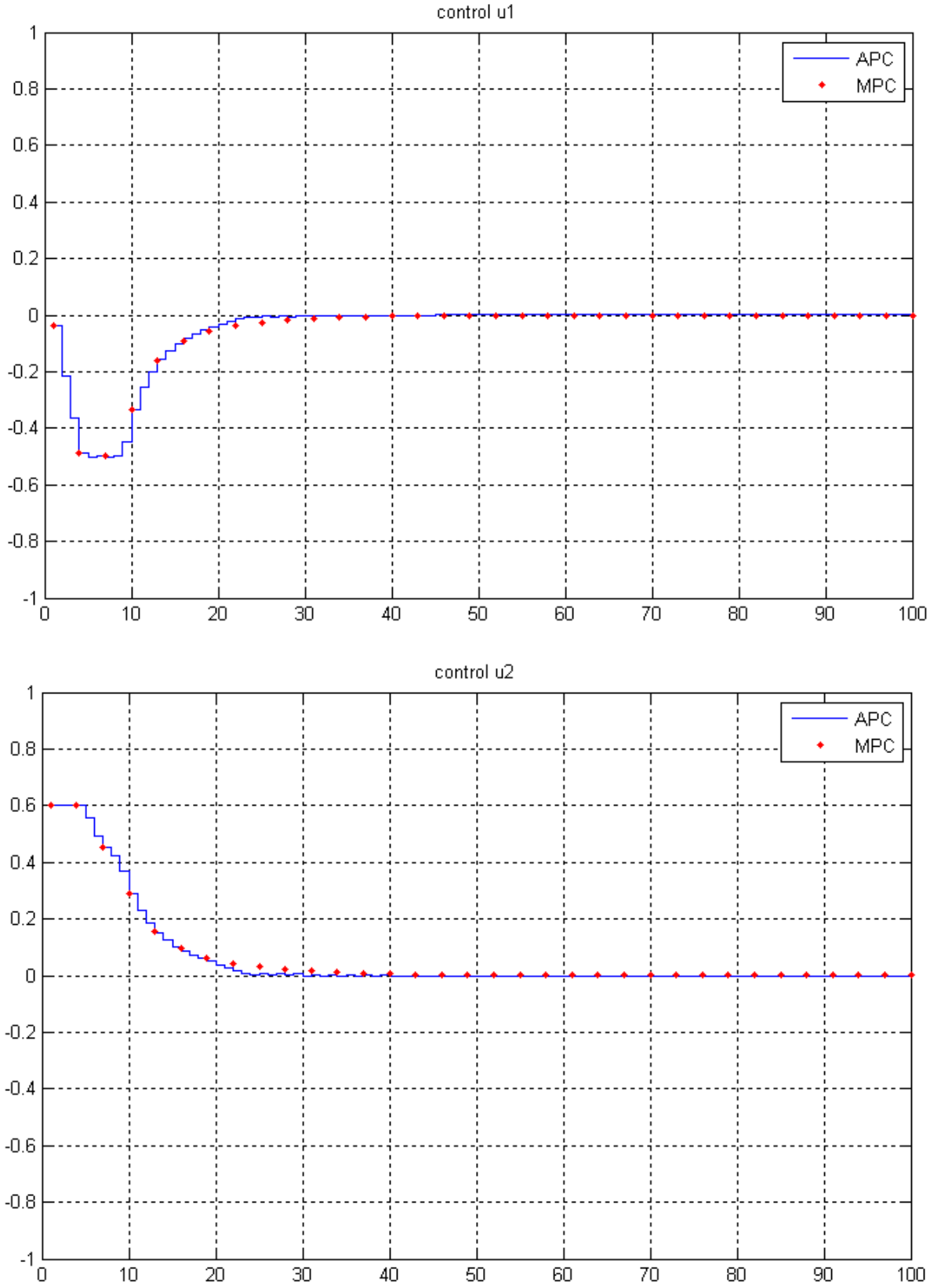


Figure 4.6 : Régulation par MPC et APC : commandes du système linéaire multi variables :cas nominal

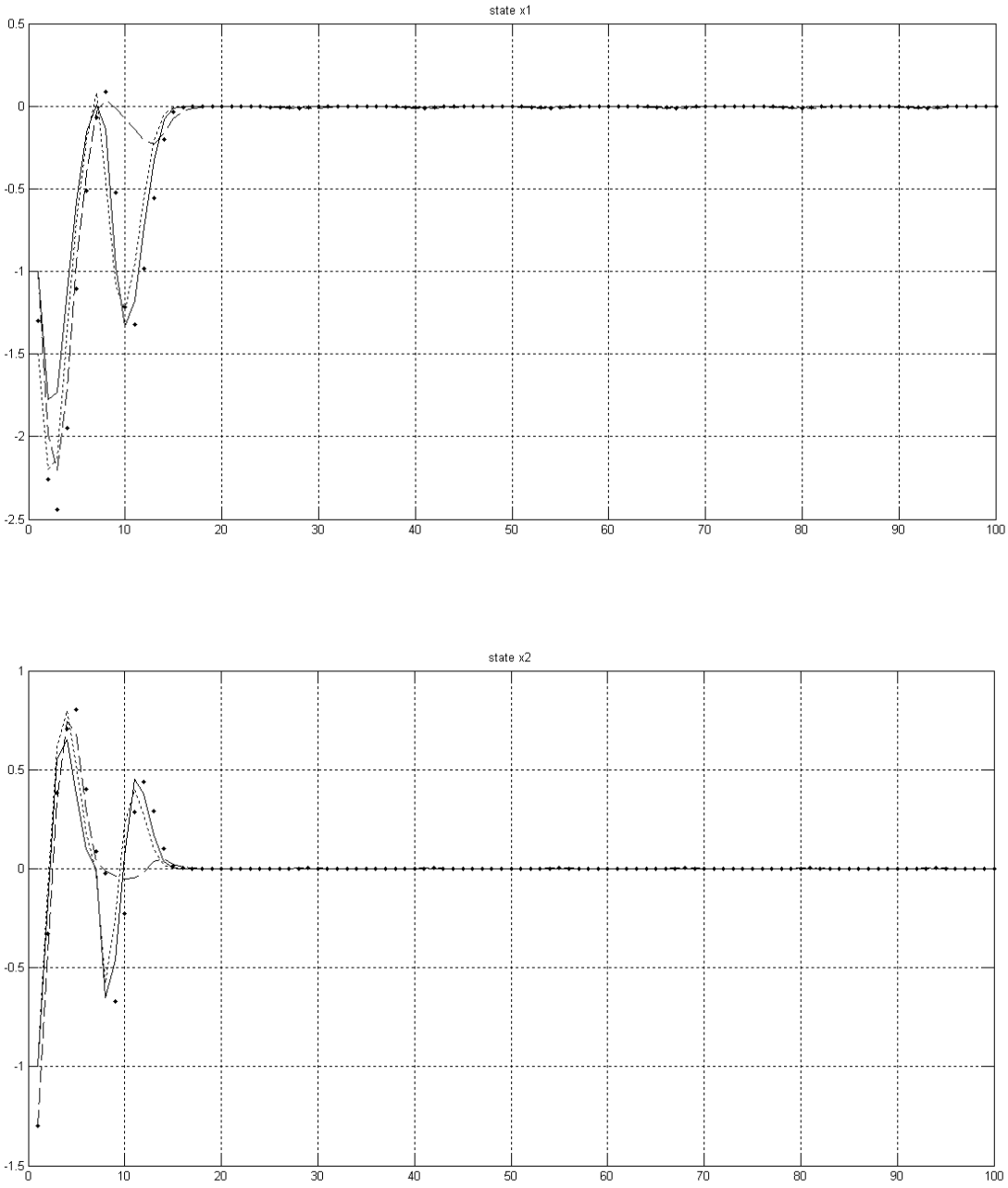


Figure 4.7 : Régulation par MPC et APC états du système linéaire multi variable: validation.

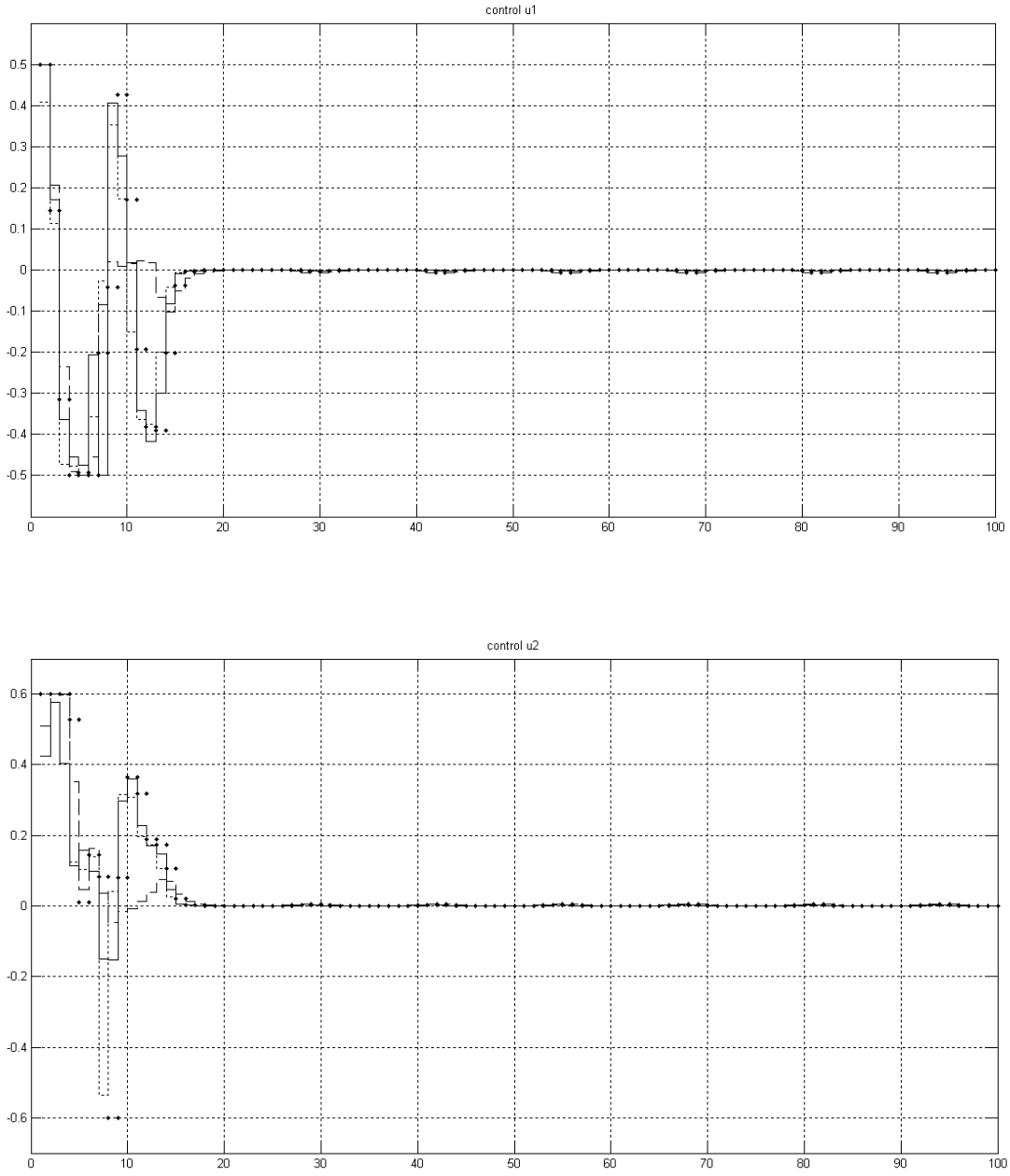


Figure 4.8 : Régulation par MPC et APC commandes du système linéaire multi variable : validation

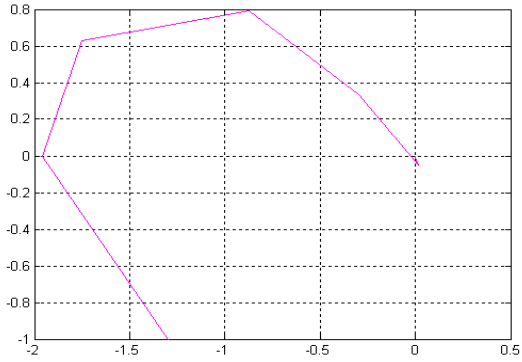


Figure 4.9 : Représentation des résultats de simulations de la régulation par APC dans l'espace d'état

4.3. Cas des systèmes Non Linéaire:

4.3.2. Système mono-variable: Pendule inversé

Considérons le modèle non linéaire du pendule inversé donné par [21] :

$$\dot{x}(t) = \begin{bmatrix} v(t) \\ w(t) \\ \dfrac{a_1 w_1\big(x(t),u(t)\big) + w_2\big(x(t)\big)\cos\theta(t)}{d\big(x(t)\big)} \\ \dfrac{w_1\big(x(t),u(t)\big)\cos\theta(t) + a_2 w_2\big(x(t)\big)}{d\big(x(t)\big)} \end{bmatrix}$$

Où le vecteur d'état $x^t(t) = [p(t), \theta(t), v(t), w(t)]$avec $p(t)$ est la position du chariot, $\theta(t)$ est l'angle du pendule, $v(t)$ est la vitesse du chariot, $w(t)$ est la vitesse angulaire du pendule.

Les autres termes sont donnés par :

$$w_1\big(x(t),u(t)\big) = k_1 u(t) - \big(w(t)\big)^2 \sin\theta(t) + k_2 v(t)$$

$$w_2\big(x(t)\big) = g\sin\theta(t) - k_3 w(t)$$

$$d\big(x(t)\big) = b - cos^2\big(\theta(t)\big)$$

Et les constantes

$$a_1 = \frac{J_p}{ml}, \qquad a_2 = \frac{1}{l}, \qquad b = \frac{J_p}{ml^2}, \qquad k_1 = \frac{c_1}{ml}, \qquad k_2 = \frac{f_c - c_2}{ml}, \qquad k_3 = \frac{f_p}{ml}$$

Avec J_p le moment d'inertie du pendule en relation avec l'axe de rotation, m masse équivalente du chariot et du pendule, l distance entre l'axe de rotation et le centre du système, f_c frottement du chariot, f_p frottement de rotation, c_1 gain commande de rapport signal PWM et c_2 le gain de commande de rapport de vitesse du chariot.

Les valeurs numériques sont [22]:

$$m = 0.29[kg], \quad l = 0.4[m], \quad J_p = 1.3\,10^{-3}\left[\frac{Nm}{rad/s}\right], \quad f_c = 0.3\left[\frac{N}{rad/s}\right],$$

$$c_1 = 9.72\left[\frac{N}{V}\right], \quad c_2 = 46.64[kg]$$

Soumis aux contraintes sur la commande suivantes :

Sur la commande : $-0.5 < u_k < 0.5$

Sur les états (position): $-0.7 < p_k < 0.7$

En minimisant le critère (2.6) avec :

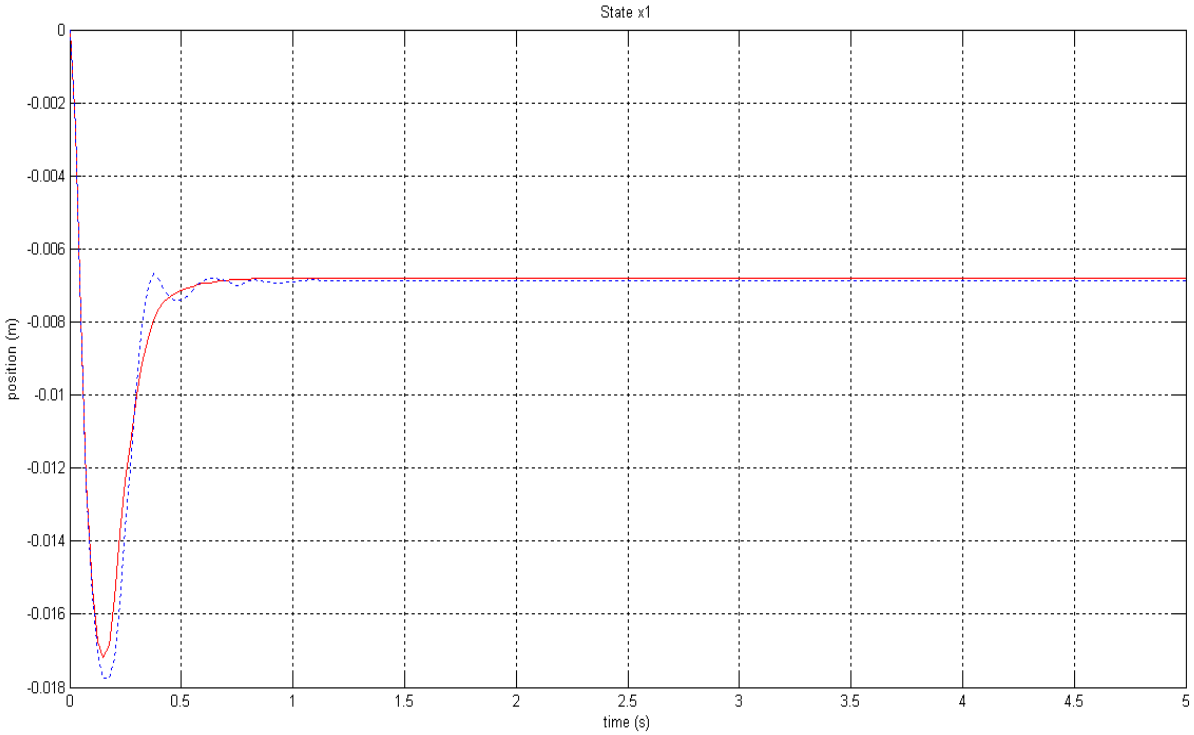
$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 10 & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 10^{-4} \end{bmatrix}, R = 0.1, N_p = 60, N_c = 3 \text{ et } \mathbf{x}_0 = \begin{pmatrix} 0 \\ \pi/4 \\ 0 \\ 0 \end{pmatrix}$$

Paramètres PSO :

Paramètres / Méthode	Taille d'essaim	Nombre d'itérations
MPC	50	50
APC	50	10000

Les figures 4.10, 4.11 et 4.12 donnent les résultats de l'optimisation de la commande MPC par PSO et de la commande approximée obtenue après optimisation par PSO pour le cas nominal. Les figures 4.13 et 4.14 représentent les résultats de validation du contrôleur APC pour des conditions initiales autres que celles ayant servies à l'optimisation où on remarque que le contrôleur approximant arrive à réguler le système.

Le contrôleur flou a donné une bonne approximation de la solution calculée par MPC, avec une moyenne du temps d'exécution de 1.6 ms pour APC contre une moyenne de 1.8576 s pour MPC, et une erreur en régime permanent de l'ordre de 10^{-4} pour APC et 10^{-6} pour MPC.



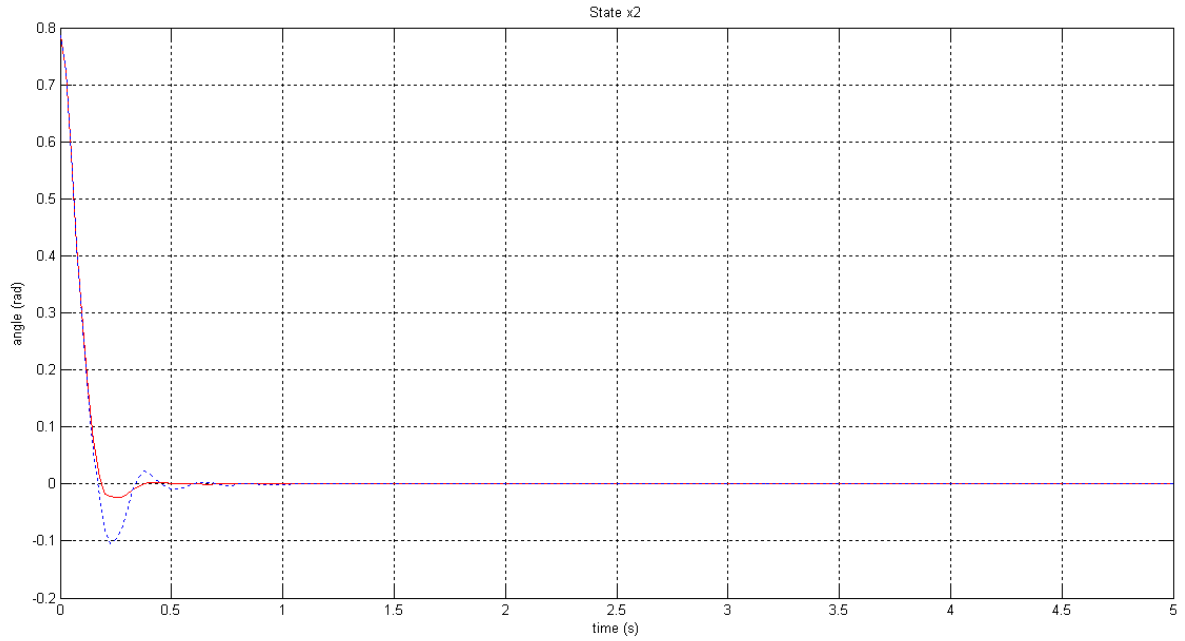
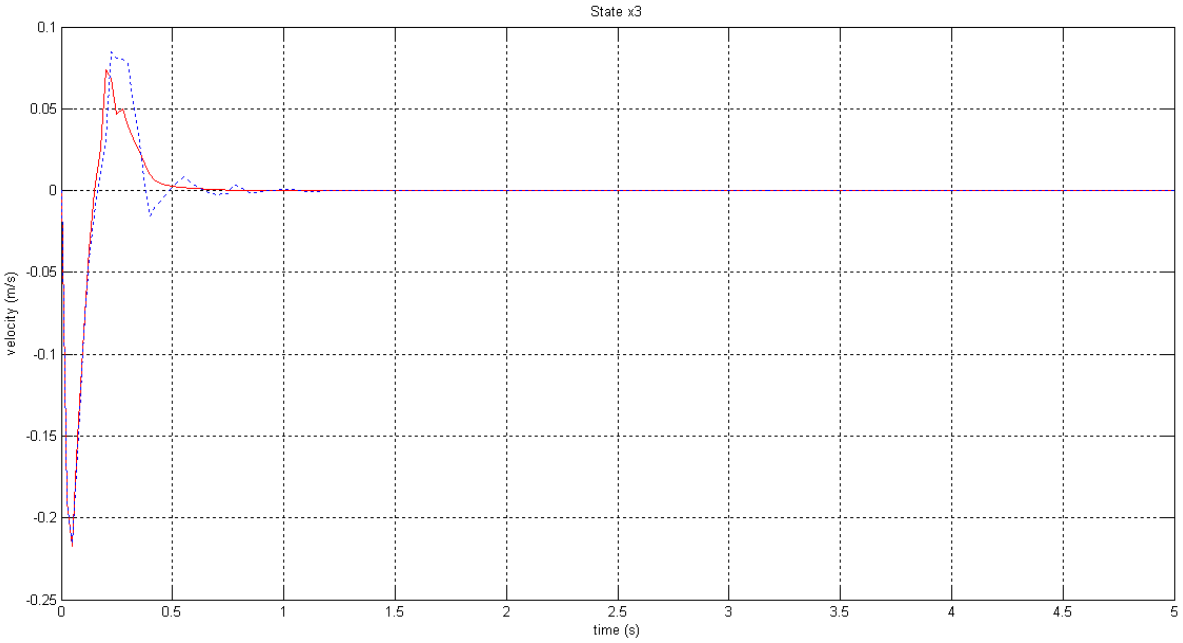


Figure 4.10 : Régulation par MPC et APC états $x_1$et x_2 du pendule inversé : cas nominal



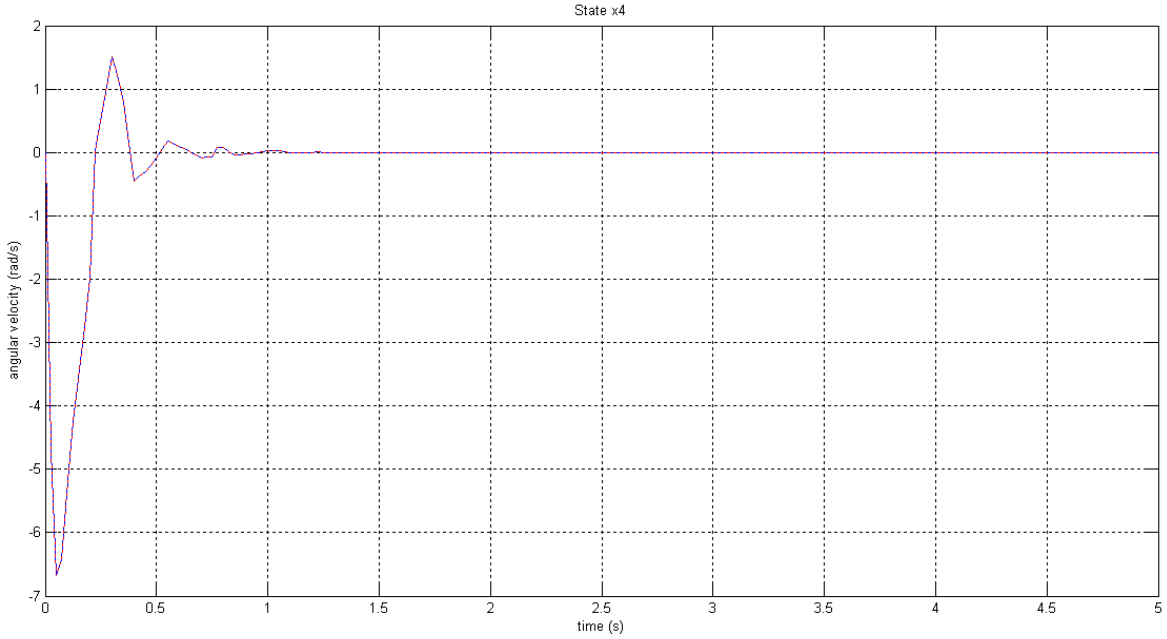


Figure 4.11 : La régulation par MPC et APC états x_3 et x_4 du pendule inversé : cas nominal

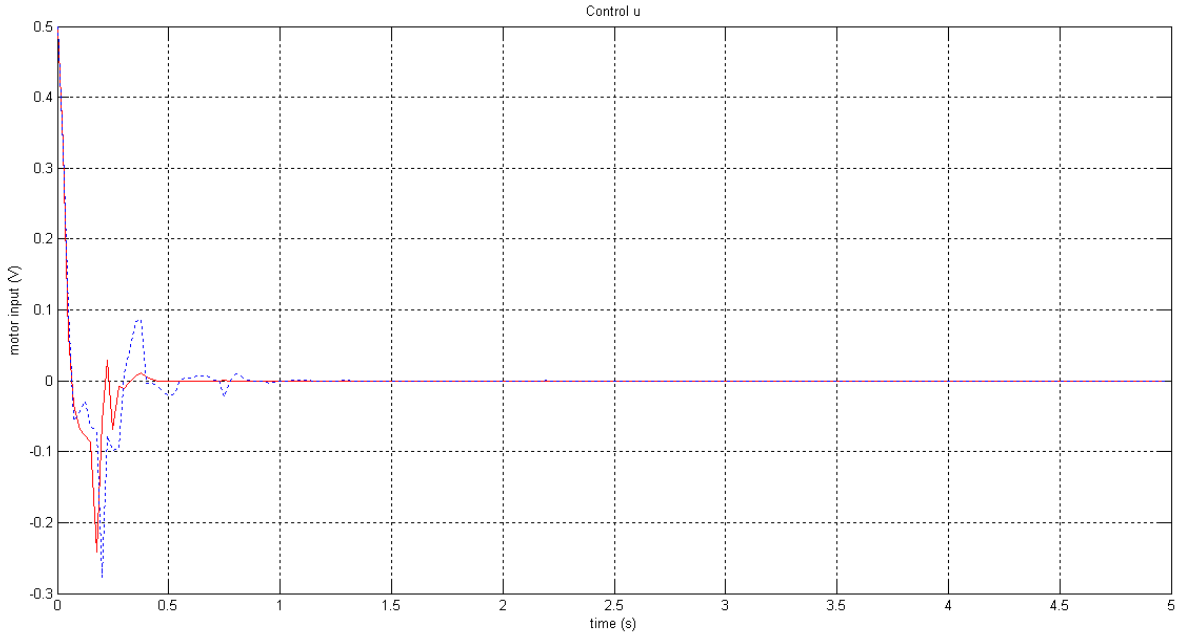


Figure 4.12 : Régulation par MPC et APC le signal de commande : cas nominal.

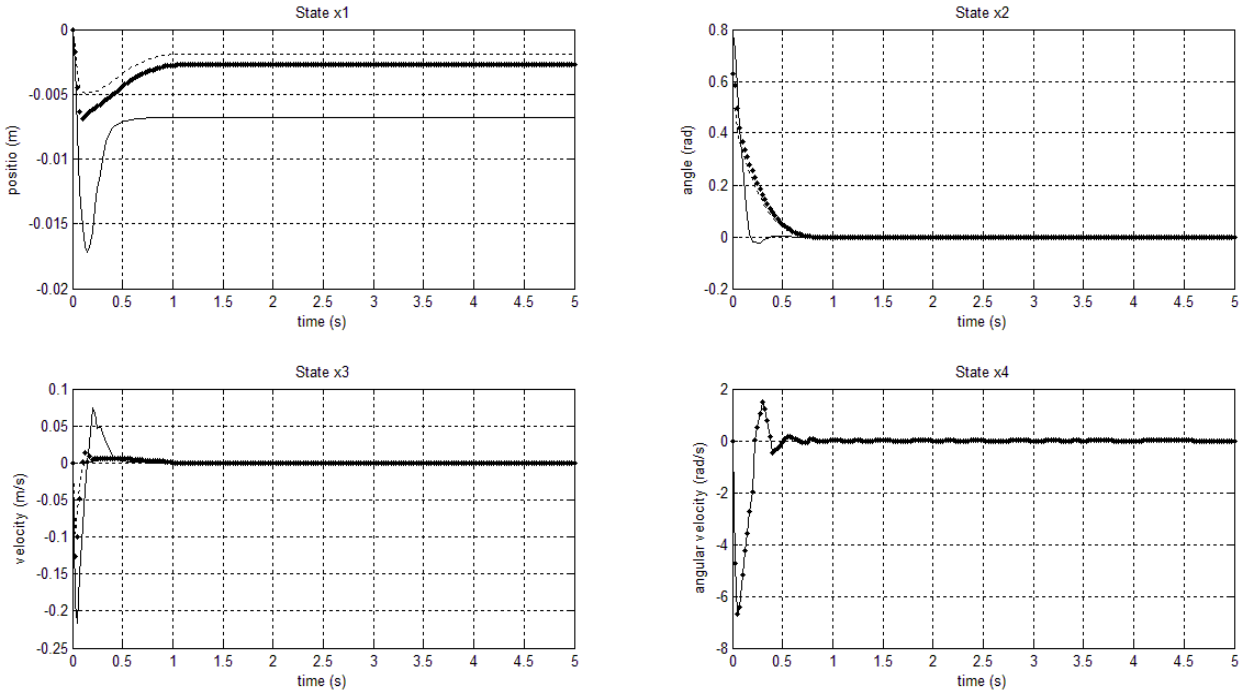


Figure 4.13 : Régulation par MPC et APC : Validation états du pendule inversé pour plusieurs points de départs.

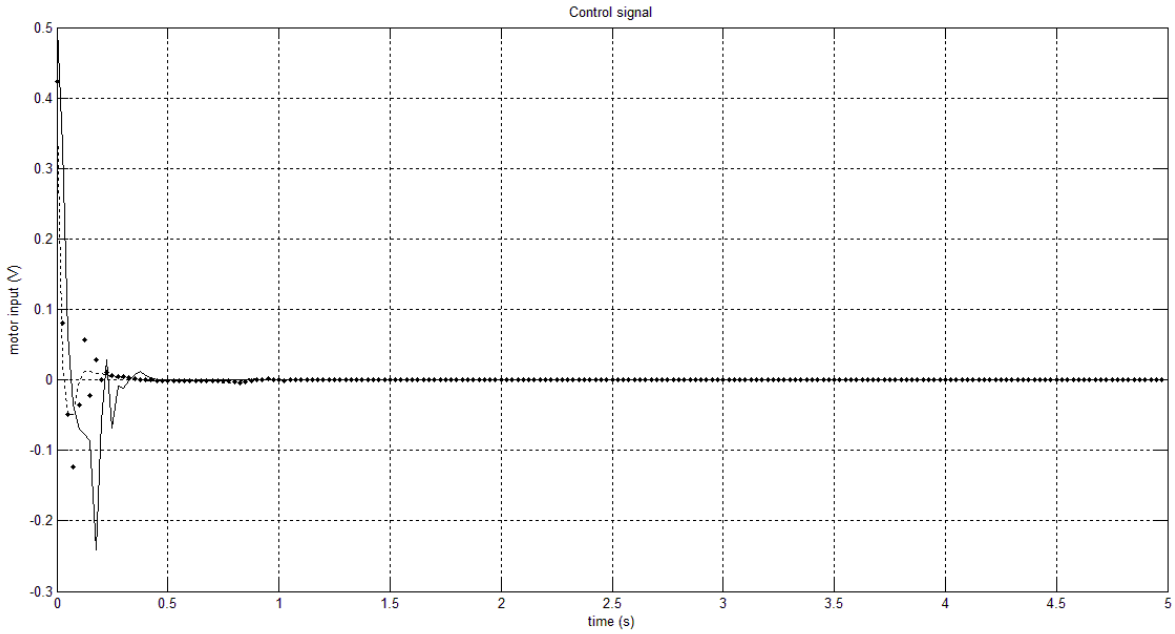


Figure 4.14 : Régulation par MPC et APC : Validation commande du pendule inversé pour plusieurs points de départs.

4.3.2. Système multi-variable

On considère le problème de régulation à l'origine du système [20] continu multi-variables :

$$\dot{\mathbf{x}} = \begin{bmatrix} 1 - x_1^2 & 1 \\ 1 & x_1^2 - 1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}$$

Soumis aux contraintes sur les commandes suivantes :

$$-3 < u_1 < 3\,, \qquad -2 < u_2 < 2$$

En minimisant le critère (2.6) avec :

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, N_p = 15, N_c = 5 \text{ et } \mathbf{x}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Paramètres PSO :

Paramètres / Méthode	inertie	Facteurs de Corrections	Taille d'essaim	Nombre d'itérations
MPC	1 .4	2	30	100
APC	0.8	2	50	20000

Les figures 4.15 et 4.16 donnent les résultats de l'optimisation de la commande MPC par PSO et de la commande approximée obtenue après optimisation par PSO pour le cas nominal. Les figures 4.17 et 4.18 représentent les résultats de validation du contrôleur APC pour des conditions initiales autres que celles ayant servies à l'optimisation où on remarque que le contrôleur approximant arrive à réguler le système. Le contrôleur flou a donné une bonne approximation de la solution calculée par MPC, avec une moyenne du temps d'exécution de $3.4725e^{-4}$s pour APC contre une moyenne de 0.8576 s pour MPC, et une erreur en régime permanent de l'ordre de 10^{-4} pour APC et 10^{-7} pour MPC.

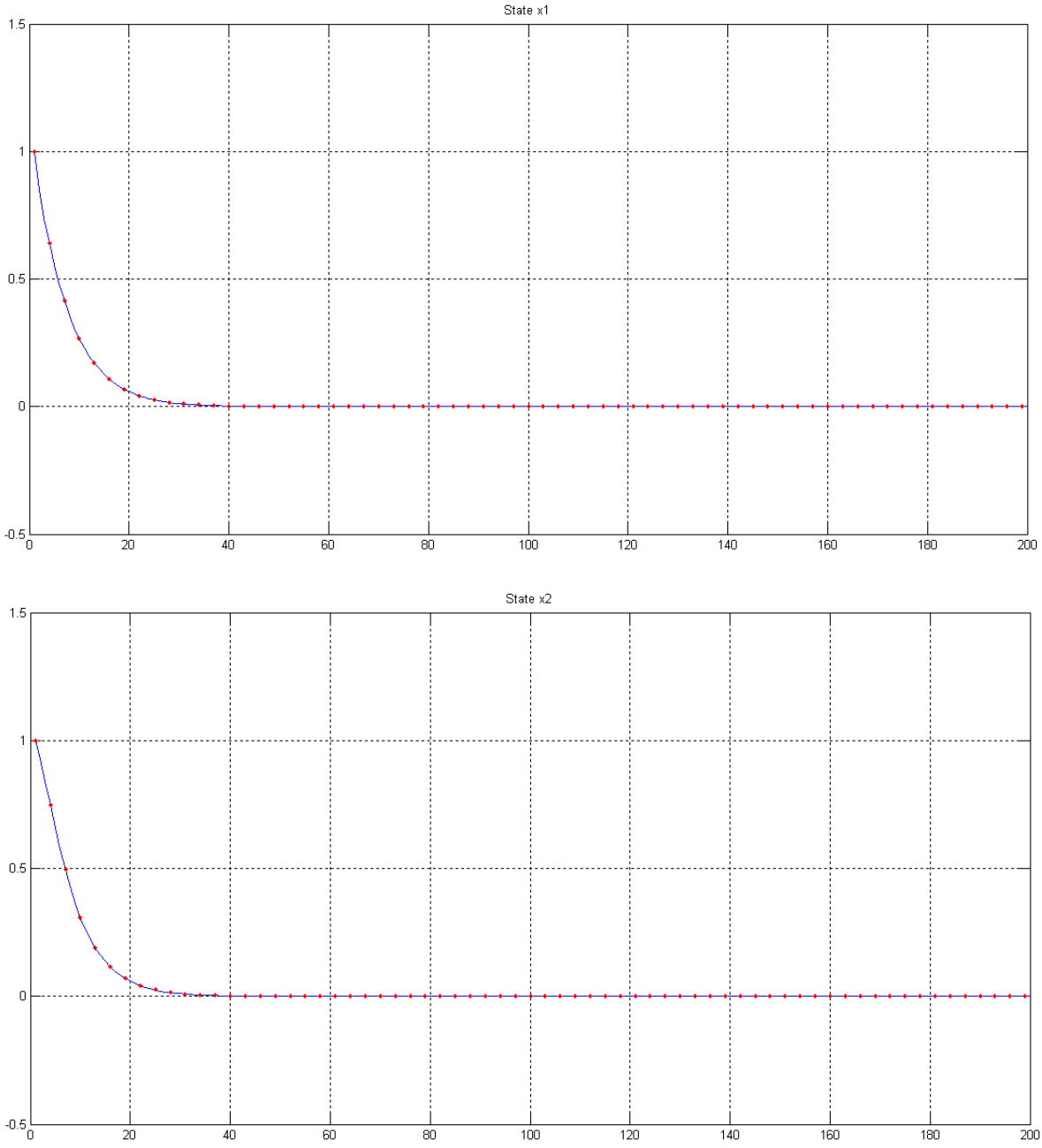
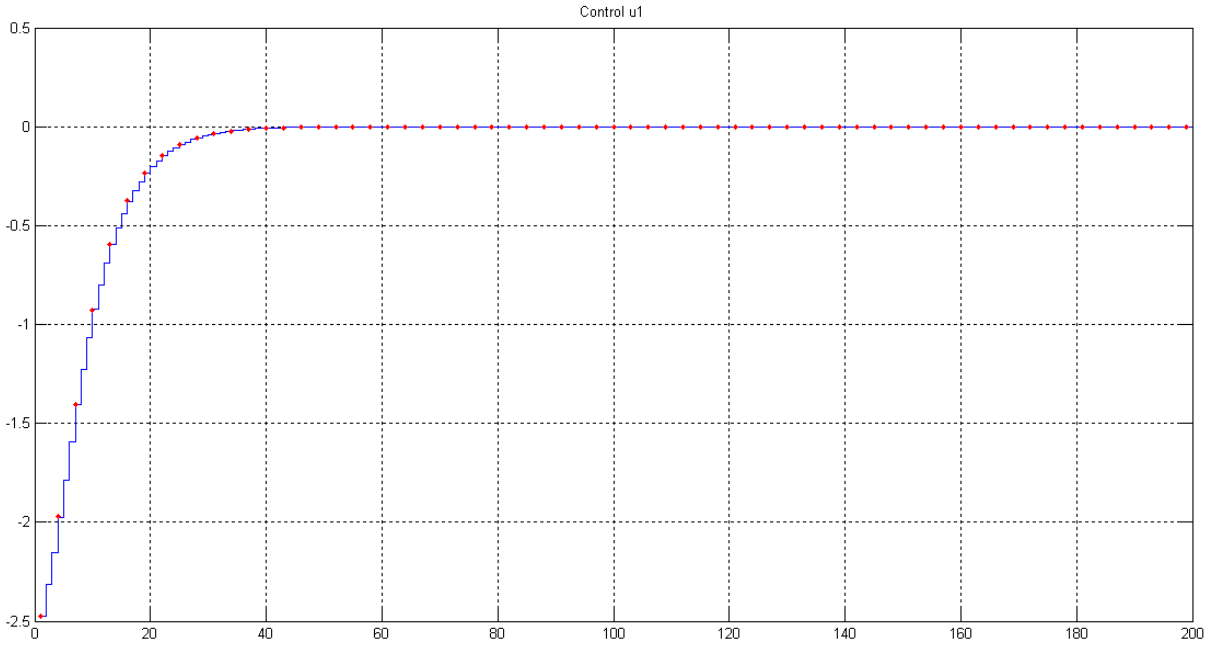


Figure 4.15 : La régulation par MPC et APC les états du système non linéaire multi variables : cas nominal



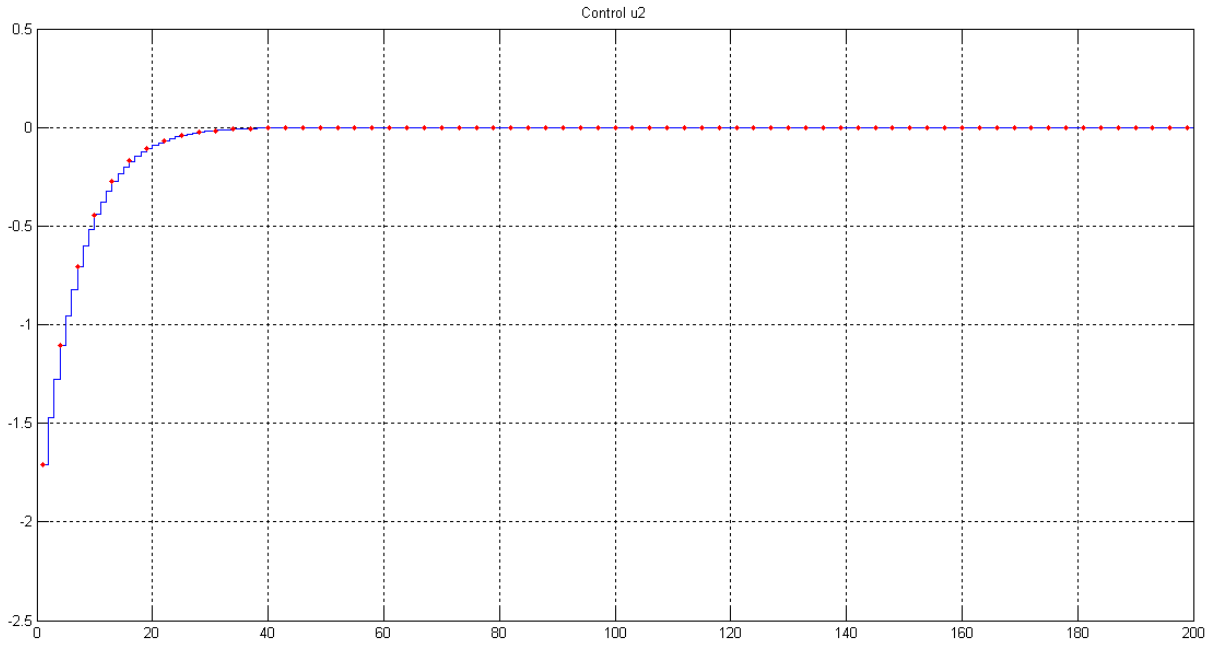


Figure 4.16 : La régulation par MPC et APC
les commandes du système non linéaire multi variables : cas nominal

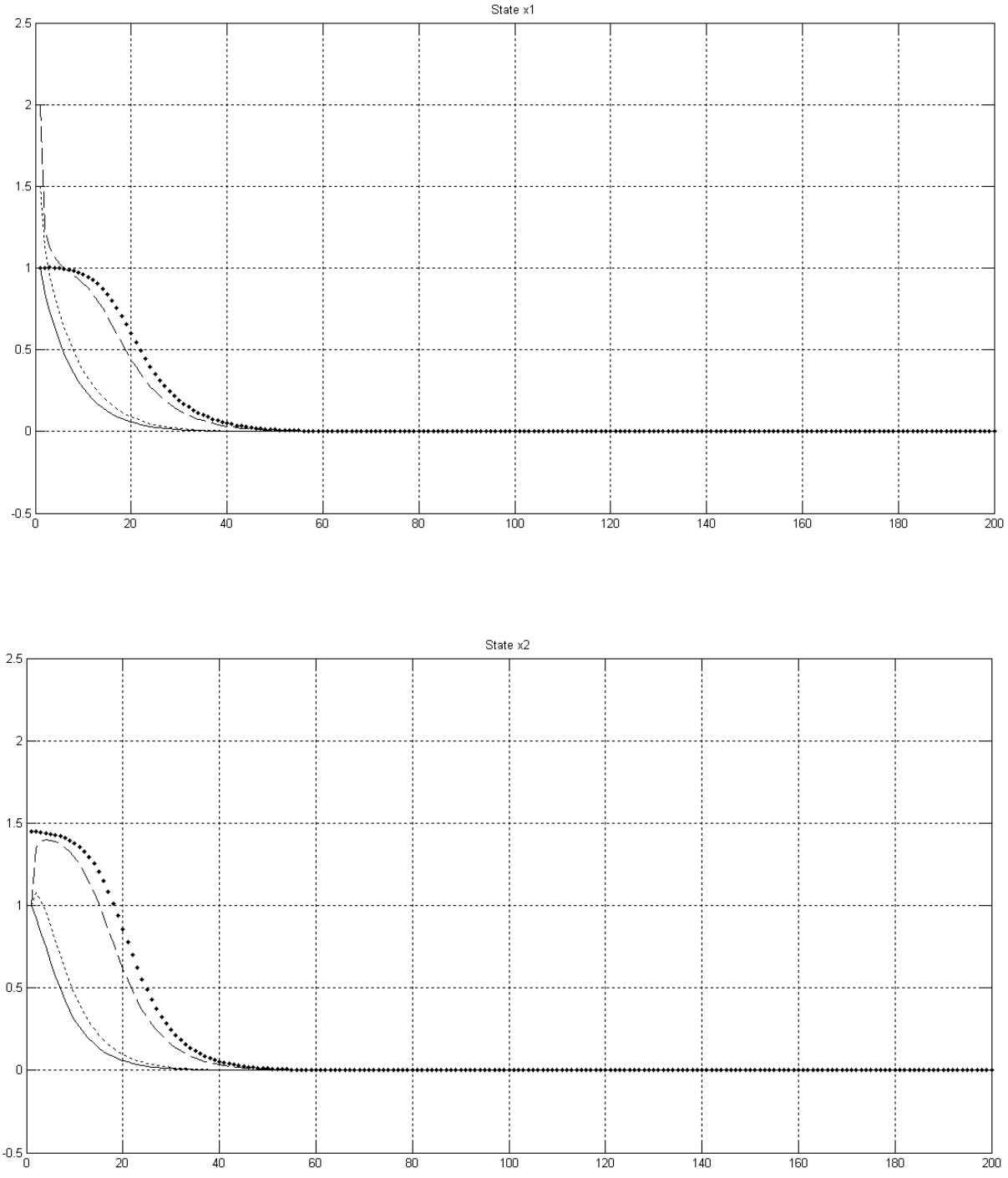
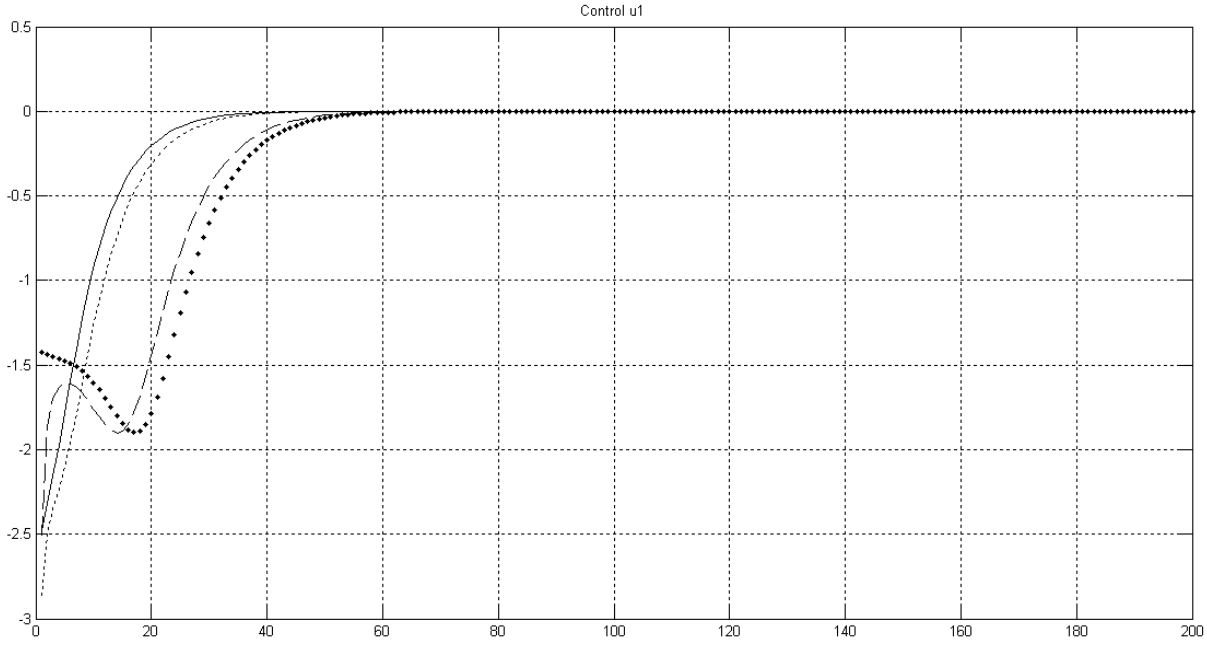


Figure 4.17 : Régulation par MPC et APC les états du système linéaire mono variable : validation



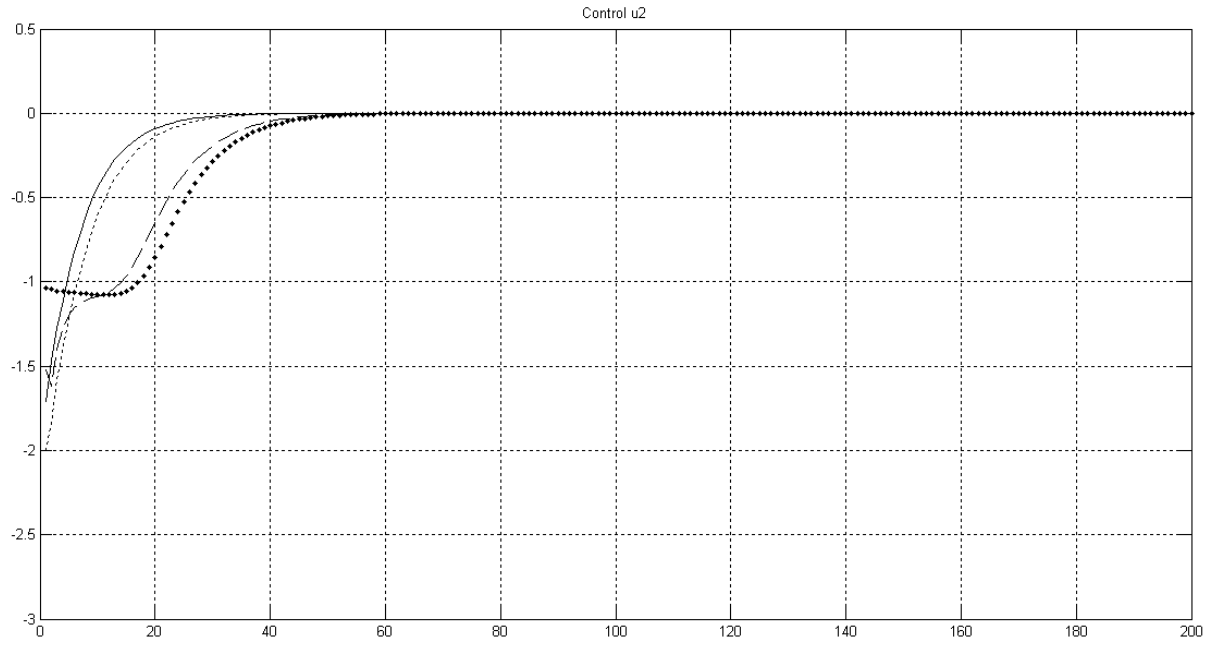


Figure 4.18 : Régulation par MPC et APC les commandes du système linéaire mono variable : validation

Chapitre 5

CONCLUSION

Dans ce mémoire, nous avons proposé l'approximation de la solution de la commande prédictive par les systèmes à logique floue de type Takagi Sugeno. Cette approximation est basée sur le théorème d'approximation universelle. La loi de commande prédictive optimale obtenue hors ligne par simulation sur un modèle du système est utilisée pour construire une base de données qui servira pour l'apprentissage du système flou. L'algorithme de commande prédictive est résolu par l'algorithme PSO. Cet algorithme est facile à mettre en œuvre, a peu de paramètres à régler et converge rapidement dans le cas des problèmes convexes.

L'apprentissage est réalisé par une minimisation de l'erreur entre la sortie du système flou et les données de la solution de la commande prédictive. La minimisation est réalisée, pareillement, par l'algorithme PSO.

L'approche a été testée sur plusieurs systèmes linéaires et non linéaires mono variable et multi variable.

Les résultats prometteurs de nos expériences nous encouragent à envisager la poursuite de cet axe de recherche dans le cadre d'une thèse de doctorat. Cette étude pourra être réalisée du point de vue théorique : étude de la stabilité et des performances et éventuellement une application pratique.

Bibliographie

[1] E.F. Camacho, C. Bordons, “Model predictive control”, Ed. Springer-Verlag, London, 2004.

[2] J. Richalet, A. Rault, J.L.Testud, J. Papon, “Model predictive heuristic control: application to industrial processes”, *Automatica*, Vol. 14, N°5, pp. 413-428, 1978.

[3] C.R. Cutler and B.L. Ramaker, “Dynamic matrix control – A computer control algorithm”, Automatic Control Conference, San Francisco, 1980.

[4]D.W. Clark, C.Mohtadi, P.S. Tuffs, “Generalized Predictive Control: Part I: The Basic Algorithm, Part II: Extensions and Interpretation”, Automatica, Vol. 23, N°2, pp.137-160, 1987.

[5] A.Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos, “The Explicit Linear Quadratic Regulator for Constrained Systems”, Automatica, Vol. 38, pp. 3-20, 2002.

[6] Grancharova, A. and Johansen, T.A. *Survey of Explicit Approaches to Constrained Optimal Control*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005

[7]A. Bemporad and C. Filippi, “Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming,” *J. Optim. Theory Applicat.*, vol. 117, no. 1, pp. 9–38, Apr. 2003.

[8] T. A. Johansen and A. Grancharova, “Approximate explicit constrained linear model predictive control via orthogonal search tree,” *IEEE Trans. Autom. Control*, vol. 58, no. 5, pp. 810–815, 2003.

[9] T.A. Johansen, “Approximate explicit receding horizon control of constrained nonlinear systems,” *Automatica*, vol. 40, pp. 293–300, 2004.

[10] D. Muñoz de la Peña, A. Bemporad, and C. Filippi, “Robust explicit MPC based on approximate multi-parametric convex programming,” *IEEE Trans. Autom. Control*, vol. 51, no. 8, pp. 1399–1403, 2006.

[11] M. Canale, L. Fagiano, and M. Milanese, “Set membership approximation theory for fast implementation of model predictive control,” *Automatica*, vol. 45, no. 1, pp. 45–54, 2009.

[12] Jones, C. N. and Morari, M. “Approximate explicit MPC using bilevel optimization”, in *Proceedings of the European Control Conference*, pp. 2396–2401, 2009.

[13] F. Scibilia, S. Olaru and M. Hovd “Approximate Explicit Linear MPC via Delaunay Tessellation”. The 10th European Control Conference, Budapest, Hungary, August 23-26, 2009.

[14] A. Bemporad, Alberto Oliveri, Tomaso Poggi, and Marco Storace,”Ultra-Fast Stabilizing Model Predictive Control via Canonical Piecewise Affine Approximations”, IEEE Trans. Automatic Control, vol. 56, no. 12, pp. 2883, 2011.

[15] B. Kosko, “Fuzzy systems as universal approximators”, IEEE Transactions on Computers, 43, pp.1329-1333, 1994.

[16] Rawlings, J. B. and Muske, K. R. (1993). Stability of constrained receding horizon control. IEEE Transactions on Automatic Control, 38(10):723–757.

[17] Chmielewski, D. and Manousiouthakis, V. (1996). On constrained infinite-horizon linear quadratic optimal control. Systems Control Lett., 29(3):121–129.

[18] Eberhart R.C.and J. Kennedy, “ A new optimizer using particle swarm theory,” Proceedings of the sixth inter. symposium on micro machine and human science, Nagoya, Japan, pp.39–43, 1995.

[19] M. Clerc and J. Kennedy, “ The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space”, IEEE Trans. on evolutionary computation, vol. 6, no. 1, 2002.

[20] Haessig D.A., and B. Friedland, “State dependent differential Riccati equation for nonlinear estimation and control”, 15th Triennial IFAC World Congress, Barcelona, Spain, 2002.

[21] Adam Mills, Adrian Wills, Brett Ninness “Nonlinear Model Predictive Control of an Inverted Pendulum”.

[22] Freddy Mudry, “Modélisation et régulation d’un pendule inversé”, Fascicule de travaux pratiques, Institut d’Automatisation industrielle de l’Ecole d’Ingénieurs du Canton de Vaud, Suisse, Octobre 2003.

ملخص :

التحكم التنبؤي يعتبر من أفضل التقنيات المستعملة في المجال الصناعي و هذا لتأقلمه مع مختلف أنواع الأنظمة وكذلك مع القيود المفروضة عليها. و برغم وجود هذه الأفضلية تبقى هنالك محدودية تطبيق هذه التقنية على الأنظمة السريعة. نقترح في هذه الأطروحة مقاربة تعتمد على التحكم التنبؤي المقرب عن طريق التعلم الضبابي. فيما يستعمل أحسن قانون تحكم متحصل عليه , من خلال محاكاة أجريت باستعمال نموذج رياضي للنظام المدروس , كقاعدة للبيانات لتدريب النظام الضبابي. و لتوضيح مدى أهمية الفكرة المقترحة و للتحقق من صحتها أجريت محاكاة في هذا الصدد.

الكلمات المفتاحية: التحكم التنبؤي, التحكم التنبؤي المقرب, التدريب النظام الضبابي

Abstract:

Predictive control is by far the best known method in industry for its tolerance towards different types of systems and the constraints imposed. Nevertheless, it is these limitations upwind speed systems. We propose in this work an approach that is based on the approximate predictive control by fuzzy learning. The optimal predictive control law obtained off line using simulation model is used as data base for training the fuzzy system. Simulations were performed to illustrate the interest of our approach and its validation.

Key words: Model Predictive Control, Explicit MPC, Approximate Predictive Control, Takagi Sugeno Fuzzy learning.

Résumé :

La commande prédictive reste de loin la méthode la plus connue dans le milieu industriel pour sa tolérance envers différents types de systèmes et le respect des contraintes imposées. Néanmoins, elle trouve ces limitations au prés des systèmes rapides.

Nous proposons dans ce travail une approche qui se base sur la commande prédictive approximante par apprentissage flou. La loi de commande prédictive optimale, obtenue hors ligne par simulation sur un modèle du système, est utilisée pour construire une base de données qui servira pour l'apprentissage du système flou. L'apprentissage est réalisé par une optimisation qui utilise l'algorithme d'optimisation par essaim de particules, Particle Optimisation Algorithm, PSO. Des simulations ont été réalisées afin d'illustrer l'intérêt de notre approche et sa validation.

Mots clé : Commande prédictive, Commande prédictive explicite, Commande Prédictive Approximante, Apprentissage flou.