

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE de Constantine
Faculté des Sciences de l'ingénieur
Département d'Electronique

Thèse pour l'obtention du :
Diplôme de Doctorat d'Etat en Electronique.
Option : Contrôle

Par
Boussiala Noureddine

Méthodes Numériques pour la Résolution des Problèmes de
Commande Optimale Via les Fonctions à Blocs d'Impulsions
(BPFs)

Jury :

Président : M. L. Riabi Prof à l'Université de Constantine

Rapporteur : A. Chaabi Prof à l'Université de Constantine

Examineurs : S. Yahmedi Prof à l'Université d'Annaba

F. Krim Prof à l'Université de Sétif

S. Filali Prof à l'Université de Constantine

Décembre 2008

Remerciements

Je tiens tout d'abord à remercier Monsieur Chaabi Abdelhafid, professeur à l'Université de Constantine, qui a assuré l'encadrement de mon travail où Il a été une source de motivation et d'encouragement. Je conserverai un très bon souvenir de nos discussions.

J'adresse également mes sincères remerciements :

A Monsieur Riabi Mohamed Lamine, professeur à l'Université de Constantine, pour avoir accepté d'être président du jury de cette thèse.

A Messieurs : Filali Salim, professeur à l'Université de Constantine, Yahmedi Said, professeur à l'Université d'Annaba, et Krim Fateh, professeur à l'Université de Sétif, pour l'honneur qu'ils m'ont fait en acceptant de participer à ce jury.

A toute ma famille et mes amis qui m'ont donné la motivation pour réussir cette tâche, parfois difficile.

Enfin je voudrais remercier très chaleureusement ma femme qui sans elle je ne serais pas arrivé à réaliser cette thèse.

Dédicace

A mes parents

A oum Wassim

A mes filles et fils

Et a mes frères et sœurs

Resumé

Plusieurs méthodes numériques ont été proposées pour résoudre les problèmes de commande optimale. Ces méthodes sont classées en méthodes directes et méthodes indirectes. Cette thèse est consacrée à résoudre les problèmes de commande optimale en utilisant l'approche directe où ce problème est converti en un problème de programmation mathématique. Les méthodes directes peuvent être employées en utilisant la technique de paramétrisation qui peut être appliquée de trois manières différentes : paramétrisation de la commande, paramétrisation d'état-commande et paramétrisation d'état. Dans cette thèse, la paramétrisation d'état-commande est utilisée.

Cette thèse présente des méthodes numériques pour résoudre les problèmes de commande optimale soumis ou non à des contraintes en employant la paramétrisation d'état-commande via les fonctions à blocs d'impulsion (BPFs), ceci permet d'approximer le système de variables d'état et de commande par des séries de fonctions impulsionnelles à paramètres inconnus.

En plus, dans cette thèse nous décrivons une méthode pour déterminer la commande optimale en boucle fermée des systèmes à coefficients variables ainsi qu'une méthode hybride, basée sur les algorithmes génétiques et l'algorithme de Newton nécessaire à la résolution du système d'équations non linéaires dérivé, pour trouver l'optimum global de certains problèmes de commande optimale.

Les méthodes proposées ont été appliquées sur plusieurs exemples où on a remarqué qu'elles donnent des résultats meilleurs ou comparables aux résultats trouvés par d'autres méthodes. Enfin, nous avons appliquée l'algorithme hybride à un problème pratique de la chimie industrielle. Celui-ci est non linéaire et multivariable. Les résultats obtenus sont très satisfaisants par comparaison avec ceux de la littérature.

Table des matières

Chapitre 1 Introduction.....	1
Chapitre 2 Problème De La Commande Optimale : Rappel	4
2.1 Introduction	4
2.2 Rapport de la commande optimale.....	5
2.3 Programmation dynamique	6
2.4 Conditions Nécessaires d'Optimalité	7
2.4.1 Equations d'Euler-Lagrange.....	7
2.4.2 Principe Du Minimum De Pontryagin	10
2.5 Méthodes Indirectes	10
2.5.1 Méthodes En Boucle Fermée	10
2.5.2 Méthodes De Commande En Boucle Ouverte	12
2.6 Méthodes Directes	12
2.6.1 Méthodes De Discrétisation	12
2.6.2 Méthodes De Paramétrisation	13
2.6 Conclusion.....	16
Chapitre 3 Problème De La Commande Optimale Linéaire	17
3.1 Introduction	17
3.2 Formulation Du Problème	17
3.3 Paramétrisation Via Les BPFs.....	18
3.3.1 Propriétés Opérationnelles Des BPFs	18
3.3.2 Etude Du Problème De La Commande Optimale En Boucle Fermée ...	21
3.3.3 Etude De La Commande Optimale En Boucle Ouverte Sans Contraintes	35
3.5 Etude De La Commande Optimale Linéaire Avec Contraintes	41
Chapitre 4 Problème De La Commande Optimale Non Linéaire.....	51
4.1 Commande Optimale Sans Contraintes.....	51
4.1.1 Introduction	51
4.1.2 Formulation Du Problème	51
4.1.3 Paramétrisation Du Problème.....	52
<i>B- Convergence Du Problème D'approximation</i>	57
4.2.1 Introduction	72
4.2.2 Formulation Du Problème	73
4.2.3 Paramétrisation du problème.....	73

Chapitre 5 Problème de la Commande Optimale Non Linéaire Via les Algorithmes Génétiques et les BPFs.....	84
5.1 Introduction	84
5.2 Principe des algorithmes génétiques simples	85
5.2.1 Terminologie	86
5.2.2 Description de la Méthode	86
5.3. Commande Optimale Basé sur les BPFs et les AG.....	90
5.3.1 Position du Problème.....	90
5.3.2 Paramétrisation du Problème via les BPFs	91
5.3.3. Solution du Problème via les AG.....	92
5.4. Exemple d'Application.....	93
5.5 Conclusion	104
 Chapitre 6 Conclusions et Perspectives.....	 105
6.1 Conclusions	105
6.2 Perspectives	106

Chapitre 1

Introduction

Le problème de la commande optimale est de trouver une fonction commande $u^*(t)$ qui minimise une fonctionnelle coût (indice de performance) qui satisfait les équations d'état du système et les contraintes. Il a d'importantes applications de types économiques, environnementaux, de l'engineering, etc.

Généralement, les problèmes de la commande optimale sont résolus de façon numérique, par conséquent les méthodes de résolution ont nettement évoluées ces dernières années. La plupart des anciennes méthodes étaient basées sur l'obtention d'une solution qui satisfait soit les équations d'Euler-Lagrange, qui sont des conditions nécessaires d'optimalité, soit l'équation de Hamilton Jacobi Bellman, qui est une condition suffisante d'optimalité. Ces méthodes sont appelées les méthodes indirectes.

L'inconvénient principal des méthodes indirectes, entres autres, est la résolution fastidieuse de l'équation de Hamilton-Jacobi-Bellman. Ce qui a amené plusieurs chercheurs à utiliser des méthodes directes pour résoudre le problème de la commande optimale. Par l'utilisation de ces méthodes, le problème de la commande optimale est approximé par un problème de programmation non linéaire à dimension finie. Ces méthodes peuvent être appliquées soit en utilisant la technique de discrétisation soit la technique de paramétrisation. La dernière technique est utilisée dans cette thèse où les variables paramétrisées sont les variables d'état et de contrôle via les fonctions à blocs d'impulsion (BPFs). En utilisant les BPFs on a pu éviter les inconvénients rencontrés dans l'application d'autres polynômes orthogonaux pour résoudre les problèmes de la commande optimale.

Un autre avantage de l'utilisation des BPFs est les propriétés intéressantes dont sont dotées ces fonctions qui permettent de traiter le problème de la commande optimale dans son aspect le plus général, contrairement aux autres méthodes les plus récentes qui nécessitent, soit une approximation de la fonction

coût et des équations d'état [88], soit une application de la méthode à des problèmes particuliers de faibles dimensions, soit l'application de la méthode au détriment du temps de calcul.

Le but de cette thèse est, donc, de résoudre le problème de la commande optimale en le rendant directement, par l'utilisation des BPFs, à un problème de programmation non linéaire ou tout simplement à un problème de résolution numérique d'un ensemble de systèmes d'équations algébriques. Cette technique est appliquée de la même manière à des problèmes linéaires ou non linéaires, monodimensionnels ou multidimensionnels, avec ou sans contraintes. En outre, pour éviter le problème de la convergence vers l'optimum local, on a utilisé une nouvelle méthode basée sur les algorithmes hybrides, ce qui a permis de surmonter le problème du temps d'exécution relatif à l'application des algorithmes génétiques seuls tout en tirant profit pour l'obtention du minimum global.

La contribution de cette thèse est de :

- Présenter une nouvelle méthode pour résoudre le problème de la commande optimale basée sur l'utilisation des BPFs comme outil de paramétrisation. Ceci permet de convertir le problème dynamique à un problème statique.
- Décrire les méthodes numériques qui permettent de résoudre le problème de la commande optimale via les BPFs.
- Présenter une nouvelle méthode hybride de résolution basée sur les algorithmes génétiques, l'algorithme de Newton et les BPFs permettant de trouver l'optimum global.

La suite des chapitres sont organisés comme suit :

Le chapitre 2 donne une revue de quelques méthodes numériques pour résoudre le problème de la commande optimale. Dans ce chapitre, nous classifions les méthodes de calcul en méthodes directes et indirectes.

Le chapitre 3 présente une méthode numérique pour résoudre le problème de la commande optimale. Dans ce chapitre le concept de paramétrisation via les BPFs est introduit. Ce chapitre décrit aussi la méthode d'approximation du problème de commande optimale linéaire aussi bien en boucle fermée qu'en boucle ouverte. Enfin ce chapitre montre l'application de notre méthode à des problèmes standard et compare nos résultats avec ceux obtenus précédemment.

Le chapitre 4 généralise la méthode du chapitre 3 pour résoudre le problème de la commande optimale non linéaire. Ainsi ce problème est rendu grâce aux BPFs à un problème d'optimisation non linéaire avec ou sans contraintes. Une propriété élégante relative à l'approximation des fonctions composées via les BPFs est mentionnée. Les conditions de convergences sont établies. Enfin, Des exemples de simulation sont donnés comme application de cette méthode.

Le chapitre 5 décrit une méthode hybride basée sur les algorithmes génétiques associés à l'algorithme de Newton, celle ci a pour objectif de converger vers l'optimum global tout en réduisant le temps de calcul. Ainsi la notion des algorithmes génétiques est rappelée. La méthode est appliquée à la commande d'un système non linéaire multivariable comme application de cette méthode.

Le chapitre 6 contient les conclusions de cette thèse et des recommandations pour les travaux futurs.

Chapitre 2

Problème De La Commande Optimale : Rappel

2.1 Introduction

Le problème de la commande optimale a été traité dans plusieurs manuels et articles [1-10]. L'objectif de la commande optimale est de déterminer une commande optimale $u^*(t)$ ou $u^*(x, t)$ qui force le système à satisfaire des contraintes physiques tout en minimisant ou maximisant un indice de performance.

Le problème fondamental de la commande optimale consiste en les trois éléments suivants :

1. Un modèle mathématique du système à contrôler qui peut être décrit par un ensemble d'équations d'état de la forme :

$$\dot{x}(t) = f(x(t), u(t), t) \quad (2.1)$$

où $x(t) \in R^n$ est le vecteur d'état, $u(t) \in R^m$ est le vecteur de commande, f est continûment différentiable par rapport à toutes ses arguments.

2. Un ensemble de conditions aux limites sur les variables d'état qui donne la valeur des états du système à l'instant initial.

$$x(t_0) = x_0 \quad (2.2)$$

3. Un indice de performance qui doit être minimisé (ou maximisé). L'indice de performance décrit quelques spécifications désirées exprimées

mathématiquement sous forme scalaire. L'indice de performance qui nous intéresse peut être formulé comme suit :

$$J = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t) dt \quad (2.3)$$

où φ et L sont des fonctions scalaires, continûment différentiables par rapport à toutes leurs variables. Ces fonctions sont sélectionnées compte tenu des objectifs de performances.

2.2 Rapport de la commande optimale

Le problème de la commande optimale sans contraintes peut être énoncé comme suit : Etant donné f , x_0 , t_0 , t_f , φ , et L , il s'agit de trouver $u^*(t)$ ou $u^*(x, t)$ qui minimise l'indice de performance :

$$J = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t) dt \quad (2.4)$$

sachant que :

$$\dot{x}(t) = f(x(t), u(t), t) \quad , \quad x(t_0) = x_0 \quad (2.5)$$

Ce problème peut être résolu par l'une des méthodes suivantes :

- La méthode de la programmation dynamique de Bellman qui est basée sur le principe d'optimalité (L'équation de Hamilton-Jacobi-Bellman (HJB)).
- La méthode variationnelle et le principe du minimum de Pontryagin (équation d'Euler-Lagrange).
- Les méthodes directes utilisant la discrétisation ou la paramétrisation.

Ces méthodes seront discutées brièvement dans les sections suivantes.

En général, il n'est pas possible de résoudre analytiquement le problème (2.4), (2.5). Cependant une solution analytique existe pour un cas particulier de ce problème qui s'énonce comme suit : Trouver la commande optimale qui minimise la fonctionnelle :

$$J = x^T(t_f)Sx(t_f) + \int_{t_0}^{t_f} (x^T Qx + u^T Ru) dt \quad (2.6)$$

sachant que :

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(t_0) = x_0 \quad (2.7)$$

où S et Q sont des matrices semi définies positives et R est une matrice définie positive. Pour ce problème, la solution peut être exprimée en boucle fermée par :

$$u^*(x,t) = -R^{-1}B^T(t)P(t)x(t) \quad (2.8)$$

où $p(t)$ est la solution de l'équation de Riccati.

2.3 Programmation dynamique

L'utilisation du principe d'optimalité pour dériver une équation permettant de résoudre le problème de la commande optimale était proposée en premier lieu par Bellman [11]. L'application de ce principe au problème continu de la commande optimale a conduit à l'invention de la fameuse équation de HJB. Pour le problème de la commande optimale (2.4), (2.5), l'équation de HJB est donnée par :

$$\frac{\partial J^*(x(t),t)}{\partial t} = -\min_{u(t)} \left\{ L(x(t),u(t),t) + \frac{\partial J^*(x(t),t)}{\partial t} f(x(t),u(t),t) \right\} \quad (2.9)$$

et la condition finale est :

$$J^*(x(t_f),t_f) = \varphi(x(t_f),t_f) \quad (2.10)$$

La solution du problème (2.9)-(2.10) permet de trouver la commande optimale u^* en fonction de $x(t)$ et t .

La dérivation de l'équation de HJB peut être trouver dans plusieurs manuels standard de la commande optimale, voir par exemple [2, 4-6]. Cette équation est une condition suffisante d'optimalité, elle est satisfaite pour tout couple $(x(t), t)$. Un avantage de l'utilisation de l'approche HJB est de résoudre le problème de la commande optimale pour obtenir la loi de commande en boucle fermée.

En général, l'équation de HJB ne possède pas une solution analytique. Cependant, pour le problème de la commande optimale quadratique, (2.6)-(2.7), l'équation de HJB se réduit à l'équation différentielle de Riccati:

$$-\dot{P}(t) = A^T(t)P(t) + P(t)A(t) + Q - P(t)B(t)R^{-1}B^T(t)P(t) \quad (2.11)$$

$$P(t_f) = S \quad (2.12)$$

Ce résultat peut être obtenu si la valeur $J^* = x^T P(t)x$ est substituée dans l'équation de HJB. Citons qu'une nouvelle notion de solution appelée la solution de viscosité a été introduite récemment par [13].

2.4 Conditions Nécessaires d'Optimalité

2.4.1 Equations d'Euler-Lagrange

Les conditions nécessaires d'optimalité peuvent être déduites à partir des méthodes de calculs variationnels qui sont basées sur le fait qu'à chaque point stationnaire la variation de la fonction coût disparaît pour une variation arbitraire de la commande [3].

Pour résoudre le problème (2.4)-(2.5), nous utiliserons les multiplicateurs de Lagrange, $\lambda(t) \in R^n$. Par conséquent l'indice de performance est donné par :

$$J_A = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} \left[L(x(t), u(t), t) + \lambda^T(t)(f(x(t), u(t), t) - \dot{x}) \right] dt \quad (2.13)$$

Introduisons le Hamiltonien H défini par :

$$H(x(t), u(t), \lambda(t), t) = L(x(t), u(t), t) + \lambda^T(t)f(x(t), u(t), t) \quad (2.14)$$

Nous pouvons, ainsi, écrire (2.13) sous la forme :

$$J_A = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} H(x(t), u(t), \lambda(t), t) dt - \int_{t_0}^{t_f} \lambda^T \dot{x}(t) dt \quad (2.15)$$

L'intégration du dernier terme à droite par partie conduit à :

$$\int_{t_0}^{t_f} \dot{\lambda}^T x(t) dt = \lambda^T(t_f)x(t_f) - \lambda^T(t_0)x(t_0) - \int_{t_0}^{t_f} \lambda^T(t)\dot{x}(t) dt \quad (2.16)$$

Cependant (2.15) devient :

$$J_A = \varphi(x(t_f), t_f) - \lambda^T(t_f)x(t_f) + \lambda^T(t_0)x(t_0) + \int_{t_0}^{t_f} \left[H(x(t), u(t), \lambda(t), t) + \dot{\lambda}^T(t)x(t) \right] dt \quad (2.17)$$

Le problème original, (2.4)-(2.5), a été converti à un problème de minimisation, (2.17), sans contraintes.

Pour achever la stationnarité, l'effet de la variation de la commande sur la fonction coût doit être nul pour $t_0 \leq t \leq t_f$. En considérant t_0 et t_f comme étant fixes, alors la première variation de la commande due à la variation de J_A est :

$$\delta J_A = \left[\left(\frac{\partial \varphi}{\partial x} - \lambda^T \right) \delta x \right]_{t=t_f} + \lambda^T \delta x_0 \Big|_{t=t_0} + \int_{t_0}^{t_f} \left[\frac{\partial H}{\partial u} \delta u + \left(\frac{\partial H}{\partial x} + \dot{\lambda}^T \right) \delta x \right] dt \quad (2.18)$$

Puisque $\lambda(t)$ est arbitraire, nous pouvons la choisir telle que :

$$\dot{\lambda}^T(t) = -\frac{\partial H}{\partial x} = -\lambda^T \frac{\partial f}{\partial x} - \frac{\partial L}{\partial x} \quad (2.19)$$

avec la condition finale :

$$\lambda^T(t_f) = \frac{\partial \varphi}{\partial x} \Big|_{t=t_f} \quad (2.20)$$

Comme la condition initiale $x(t_0)$ est fixe, ceci implique que $\delta x(t_0)$ disparaît, et l'équation (2.18) se réduit à :

$$\delta J_A = \int_{t_0}^{t_f} \left[\frac{\partial H}{\partial u} \delta u \right] dt \quad (2.21)$$

Pour un minimum local, il est nécessaire que δJ_A disparaît pour δu quelconque, donc il est nécessaire que :

$$\frac{\partial H}{\partial u} = \left(\frac{\partial f}{\partial u} \right)^T \lambda + \left(\frac{\partial L}{\partial u} \right)^T = 0 \quad (2.22)$$

pour tout $t_0 \leq t \leq t_f$

Les équations (2.5), (2.19), (2.20) et (2.22) sont des conditions nécessaires qui doivent être satisfaites par les solutions optimales du problème, quand l'instant final est fixe. Ces équations sont connues sous l'appellation Equations d'Euler-Lagrange.

En résumé, pour obtenir la commande optimale $u^*(t)$ qui minimise l'indice de performance (2.4) sous la condition (2.5) les équations suivantes doivent être résolues :

$$\dot{x}(t) = f(x(t), u(t), t) \quad (2.23)$$

$$x(t_0) = x_0 \quad (2.24)$$

$$\dot{\lambda}(t) = - \left(\frac{\partial f}{\partial x} \right)^T \lambda - \left(\frac{\partial L}{\partial x} \right)^T \quad (2.25)$$

$$\lambda(t_f) = \left(\frac{\partial \varphi}{\partial x} \right)^T \quad (2.26)$$

$$\frac{\partial H}{\partial u} = \left(\frac{\partial f}{\partial u} \right)^T \lambda + \left(\frac{\partial L}{\partial u} \right)^T = 0 \quad (2.27)$$

Remarques :

1. Dans le cas où le temps final n'est pas fixe, une autre condition nécessaire doit être donnée. Cette condition est dérivée en variant la fonction coût par rapport au temps. La condition nécessaire obtenue est :

$$\left(\frac{\partial \varphi}{\partial t} + H \right)_{t=t_f} = 0 \quad (2.28)$$

2. Dans le cas où $x(t_f)$ est fixe, ie :

$$x(t_f) = x_f \quad (2.29)$$

donc, seul la condition (2.26) doit être remplacée par (2.29).

2.4.2 Principe Du Minimum De Pontryagin

Dans les problèmes réels, les variables de commande sont, d'habitude, bornées, donc le Hamiltonien ne peut pas être différenciée par rapport à la commande. Dans ce cas les conditions nécessaires sont dérivées par le principe du minimum de Pontryagin [12] qui s'énonce ainsi :

Supposons que $u^*(t)$ est la commande optimale et $x^*(t)$ est la trajectoire optimale correspondante, donc il doit exister un vecteur $\lambda^*(t)$ telles que :

$$\dot{\lambda} = -\frac{\partial H}{\partial x} \quad (2.30)$$

$$\lambda(t_f) = \left(\frac{\partial \varphi}{\partial x} \right)^T \quad (2.31)$$

$$H(x^*, u^*, \lambda^*, t) \leq H(x^*, u, \lambda^*, t) \quad (2.32)$$

pour tout $t \in [t_0 \quad t_f]$ et pour tout $u(t) \in U \subset R^m$. Donc la commande optimale doit minimiser le Hamiltonien défini par (2.14).

L'inégalité (2.32) est très utilisée pour l'obtention de la commande optimale si la commande est bornée. Il est à noter que le principe du minimum est une généralisation de l'approche du calcul variationnel. Il faut aussi signaler que l'approche variationnel et le principe du minimum mènent à un problème connu comme « A non linear two-point boundary problem :NLTPBP » qui est difficile à résoudre de façon analytique. Les méthodes numériques proposées pour résoudre ces problèmes sont appelées méthodes indirectes par opposition aux méthodes directes basées sur la transformation du problème de la commande optimale à un problème de programmation non linéaire.

2.5 Méthodes Indirectes

Ces méthodes sont divisées en deux catégories : les méthodes en boucle fermée et les méthodes en boucle ouverte :

2.5.1 Méthodes En Boucle Fermée

Parmi les méthodes proposées pour obtenir la commande optimale en boucle fermée citons les approches suivantes:

- La première approche pour obtenir une approximation de la commande optimale en boucle fermée est basée sur l'utilisation du développement en séries de puissance pour résoudre soit l'équation de HJB soit le NLTPBP. Cette approche est utilisée par Lukes [14] pour trouver une solution approximative de l'équation de HJB relatif à la commande optimale à horizon infini. La solution de l'équation de HJB est réduite à la résolution d'un système d'équations algébriques. Utilisant la même idée, Willemstein [15] a développé la méthode de Lukes pour résoudre le problème de la commande optimale à horizon fini. Le travail de Lukes a été appliqué par Garrard et Jordan à la commande de l'avion F8 [16]. La même technique est utilisée par Nishikawa et al. [17] pour obtenir la solution optimale approximatif d'un indice de performance quadratique à horizon fini pour le système perturbé suivant :

$$\dot{x} = A(t)x + \mathcal{E}f(x, t) + B(t)u \quad (2.33)$$

De la même manière, Yoshida et al. [18] ont pu résoudre le problème de la commande optimale à indice de performance quadratique à horizon fini et infini pour le système décrit par :

$$\dot{x} = f(x) + Bu \quad (2.34)$$

- La seconde approche pour obtenir la commande optimale en boucle fermée est basée sur la linéarisation des conditions nécessaires d'optimalité autour de la solution optimale ou via le développement de l'indice de performance jusqu'à l'ordre 2 et les contraintes jusqu'à l'ordre 1 autour de la solution optimale [3, 19-22].

- La troisième approche [23-24] est basée sur l'écriture des équations d'état sous forme linéaire :

$$\dot{x} = f(x, u, t) = A(x, u, t)x + B(x, u, t)u \quad (2.35)$$

et, donc, de résoudre le problème de la commande optimale en résolvant l'équation de Riccati.

$$\dot{P}(x, u, t) = P(x, u, t)A(x, u, t) + A^T(x, u, t)P(x, u, t) - P(x, u, t)B(x, u, t)R^{-1}B^T(x, u, t)P(x, u, t) + Q \quad (2.36)$$

et la commande optimale est donnée par :

$$u^*(x, t) = -R^{-1}B^T(x, u, t)P(x, u, t)x(t) \quad (2.37)$$

- La quatrième approche est basée sur la solution du problème inverse de la commande optimale [25].
- D'autres approches peuvent être trouvées dans [29-31].

2.5.2 Méthodes De Commande En Boucle Ouverte

Ces méthodes sont basées sur la résolution du NTPBVP. Parmi ces méthodes, on peut citer entre autres : les méthodes du gradient, les méthodes de quasi linéarisation, les méthodes de la fonction de pénalité etc. [3-5]. Ce sont des méthodes standard pour résoudre les problèmes de la commande optimale.

2.6 Méthodes Directes

Ces méthodes offrent quelques avantages quand elles s'appliquent aux problèmes de la commande optimale. Le premier avantage est que le problème de la commande optimale dynamique peut être converti à un problème d'optimisation statique qui est facile à résoudre par rapport au problème original. Le deuxième avantage est qu'il y a des algorithmes bien développés pour résoudre les problèmes de programmation non linéaire. Le troisième avantage est la possibilité de traiter facilement différents types de contraintes. Par conséquent plusieurs auteurs ont utilisé les méthodes directes pour résoudre le problème de la commande optimale [32-34, 37, 39-41, 43, 48, 49, 52, 54, 55, 70, 75-78].

Le problème de commande optimale peut être converti à un problème de programmation mathématiques soit en utilisant la technique de discrétisation, soit en se servant de la technique de paramétrisation. Notre thèse est basée sur la deuxième technique.

2.6.1 Méthodes De Discrétisation

Toutes les techniques de discrétisation divisent l'intervalle de temps en n segments

$$t_0 < t_1 < t_2 < \dots < t_n = t_f$$

où les points t_i sont appelés des nœuds.

Une approche qui applique cette méthode discrétise les variables d'état et les variables de contrôle ; nous avons, donc, la séquence des valeurs inconnues des variables d'état et de contrôle suivante :

$$z = (x_0, x_1, \dots, x_n, u_0, u_1, \dots, u_{n-1})$$

et le système d'équations d'état est remplacé par un ensemble d'équations algébriques qui peuvent être considérées comme des contraintes d'égalité. Ce problème peut, ainsi être résolu en utilisant les techniques de la programmation non linéaire. L'inconvénient majeur de cette approche est la dimension élevée du vecteur z .

Une autre approche possible est de discrétiser la variable de contrôle seule.

$$z = (u_0, u_1, \dots, u_{n-1})$$

et d'intégrer ensuite le système d'équations d'état pour trouver les variables d'état en fonction des variables de contrôle [35, 75, 78].

2.6.2 Méthodes de Paramétrisation

La technique de paramétrisation peut être appliquée de trois manières différentes :

a-Paramétrisation de la Commande :

Cette technique est basée sur l'approximation des variables de la commande en choisissant une structure appropriée de la forme :

$$u_k(t) = \sum_{i=0}^N C_i^{u_k} \varphi_i(t) \quad k=1,2,\dots,m \quad (2.36)$$

où les $C_i^{u_k}$ sont les paramètres inconnus et les φ_i dénotent un ensemble approprié de fonctions formant une base de l'espace de commande à dimension finie. Les variables d'état sont obtenues en fonction des paramètres inconnus en intégrant les équations d'état du système, et en substituant les variables approximatives de la commande et les variables d'état correspondantes dans l'indice de performance, le problème de la commande optimale est ainsi converti à un problème de programmation non linéaire à paramètres statiques qu'il est facile à résoudre que le problème original.

Plusieurs fonctions ont été utilisées pour l'approximation des variables de la commande [33], parmi lesquelles on peut citer : les fonctions constantes par morceaux, les fonctions linéaires par morceaux, les fonctions splines d'ordre donné.

L'application de cette technique nécessite l'intégration des équations d'état qui est un processus coûteux en terme de temps de calcul [73].

b. *Paramétrisation d'Etat-Commande* [39-42, 48, 52, 54, 70, 71]

Cette approche est basée sur l'approximation, en même temps, des variables d'état et des variables de commande par une séquence de fonctions connues avec des paramètres inconnus :

$$u_k(t) = \sum_{i=0}^N C_i^{u_k} \varphi_i(t) \quad k=1,2,\dots,m \quad (2.37)$$

$$x_j(t) = \sum_{i=0}^N C_i^{x_j} \varphi_i(t) \quad j=1,2,\dots,n \quad (2.38)$$

où les $C_i^{u_k}, C_i^{x_j}$ sont des paramètres inconnus et les φ_i sont un ensemble de fonctions appropriées. En utilisant cette méthode, le problème de la commande optimale est converti à un problème de programmation mathématique non linéaire.

L'inconvénient principal de cette approche est le nombre important de paramètres que l'on a à déterminer. Cependant, en utilisant cette approche nous aboutissons à un problème de programmation non linéaire à dimension très élevées en terme du nombre de paramètres inconnus et du nombre de contraintes d'égalité.

Dans cette thèse, nous avons opté pour cette technique mais comme les fonctions φ_i dans notre cas sont des fonctions à blocs d'impulsion (BPFs), nous avons pu surmonter les inconvénients sus-cités.

c- *Paramétrisation d'Etat*

L'idée de cette technique est de faire l'approximation seulement de la variable d'état du système [43, 49, 77, 79, 88] :

$$x_j(t) = \sum_{i=1}^N C_i^{x_j} \varphi_i(t) \quad j=1,2,\dots,n \quad (2.39)$$

les $C_i^{x_j}$ sont les paramètres inconnus et les φ_i sont définies comme précédemment. Dans cette technique les variables de commande sont obtenues à partir des équations d'état. Cette technique a quelques avantages :

1. On n'a pas besoin d'intégrer les équations d'état.
2. Le nombre de paramètres inconnus est faible par rapport aux autres techniques.
3. Les paramètres peuvent être manipulés directement.

Cependant, cette technique souffre de quelques inconvénients :

1. Il est difficile de traiter les systèmes non linéaires car il n'est pas toujours facile de trouver les variables de commande comme fonction des variables d'état.
2. Il n'existe pas une voie systématique pour appliquer cette technique aux problèmes de la commande optimale à aspect général quand le nombre de variable d'état et de commande n'est pas égal.

Cette technique a été appliquée à des cas spéciaux, par exemple au problème de la commande optimale linéaire avec le nombre de variables d'état égal au nombre de variables de commande ou à des systèmes non linéaires mono entrée pouvant être exprimés sous la forme canonique contrôlable. Récemment M.Jaddu [88] a utilisée cette technique via les polynômes de Chebyshev associée à la méthode de quasilinearisation [45] pour résoudre le problème de la commande optimale dans son aspect général, cependant sa méthode souffre de la taille du problème qui demeure exorbitant, et surtout de la violation des contraintes.

La technique de paramétrisation peut être employée en utilisant différentes fonctions de base. Les BPFs sont utilisées dans cette thèse pour paramétriser les variables d'état et les variables de contrôle. Les BPFs ont des propriétés exceptionnelles (à voir dans le prochain chapitre) qui permettent d'agir directement sur le problème de la commande optimale quelque soit sa nature.

L'utilisation des fonctions à blocs d'impulsion pour résoudre le problème de la commande optimale n'est pas nouvelle. V. Purnachandra et K. Ranganatha [67] ont utilisé les BPFs pour déterminer la commande en boucle fermée d'un problème de commande optimale linéaire, P. Sannuti [89] a utilisé les BPFs

pour l'analyse et la synthèse des systèmes dynamiques linéaires, Shien-Yu Wang [90] a pu résoudre le problème de la commande optimale des systèmes non linéaires à retard en utilisant les BPFs, S. Lapin et N.D. Egupov [91, 92] ont analysé les systèmes linéaires et non linéaires à coefficients variables via les BPFs. A ce stade les BPFs sont utilisés avec succès dans l'étude du problème de la commande optimale sans contraintes. Récemment N. Boussiala et H. Chaabi [93,94] ont utilisé les BPFs pour résoudre le problème de la commande optimale des systèmes non linéaires multidimensionnels soumises à des contraintes. Nous verrons dans cette thèse qu'il est possible de traiter le problème de la commande optimale sujet à divers types de contraintes.

2.6 Conclusion

Dans ce chapitre, les différentes méthodes de commande optimale ont été rappelées brièvement. La méthode adoptée dans le cadre de cette thèse s'inscrit dans la catégorie des méthodes directes avec paramétrisation commande- état via les fonctions à blocs d'impulsion (BPFs) dont les chapitres suivants exposent de façon détaillée l'outil mathématique ainsi que les applications inhérentes.

Chapitre 3

Problème De La Commande Optimale Linéaire

3.1 Introduction

Comme il a été dit au chapitre précédent, le problème de la commande quadratique linéaire est l'un des problèmes dont une solution analytique optimale peut être obtenue [2, 3]. La solution de ce problème peut être obtenue soit en résolvant l'équation de Riccati, soit en résolvant le LTPBVP.

Pour éviter les difficultés associées à l'intégration numérique de ces méthodes, deux approches sont utilisées : La première approche consiste à convertir le problème de commande optimale quadratique en un problème de programmation quadratique [41, 84, 86]. La seconde approche est de le résoudre en convertissant le LTPBVP en un ensemble d'équations algébriques [61, 62, 64, 66, 67, 69, 85]. Dans ce chapitre les deux approches sont utilisées ; la première est utilisée dans les problèmes à boucle ouverte, la seconde est employée pour trouver la loi de commande en boucle fermée du problème de commande optimale quadratique à coefficients variables.

Ce chapitre est abordé comme suit : D'abord, on pose le problème de la commande optimale à critère quadratique, ensuite les outils nécessaires à la paramétrisation via les BPFs sont rappelés, enfin ces outils sont exploités pour reformuler le problème soit sous forme d'un système d'équations algébriques, soit sous forme d'un problème de programmation quadratique où les algorithmes ne manquent pas pour le résoudre.

3.2 Formulation Du Problème

Considérons le système dynamique décrit par les équations d'état suivantes :

$$\dot{x} = Ax + Bu \quad (3.1)$$

Où $x \in R^n$, $u \in R^r$, A et B sont respectivement des matrices à valeurs qui peuvent dépendre du temps de dimension $n \times n$ et $n \times r$. Nous avons assumé que le processus commence à $t=0$ et finira à $t_f > 0$.

La condition initiale pour l'équation d'état (3.1) est :

$$x(t_0) = x_0 \quad (3.2)$$

Où x_0 est un vecteur donné dans R^n

Le problème de la commande optimale est de trouver une commande optimale $u^*(t)$ qui minimise l'indice de performance quadratique :

$$J = \frac{1}{2} \int_0^{t_f} (x^T Q x + u^T R u) dt \quad (3.3)$$

sous les conditions (3.1) et (3.2).

Q est une matrice semi définie positive de dimension $n \times n$ et R est une matrice définie positive de dimension $r \times r$.

Dans ce chapitre, nous proposons une méthode pour résoudre ce problème de commande optimale en le transformant à la résolution d'un ensemble d'équations algébriques. Cette solution est basée sur l'approximation des variables d'état et de commande via les BPFs.

3.3 Paramétrisation Via Les BPFs

Avant de commencer la discussion de la paramétrisation, quelques propriétés importantes des BPFs sont rappelées

3.3.1 Propriétés Opérationnelles Des BPFs

- Les fonctions à blocs d'impulsions de rang m sont définies de la façon suivante :

$$\varphi_i(t) = \begin{cases} 1 & \text{si } t \in [t_0 + (i-1)h, t_0 + ih], \quad i = 1, \dots, m \\ 0 & \text{Sinon} \end{cases} \quad (3.4)$$

Il est aisé de voir que les propriétés suivantes sont toujours vérifiées

$$\varphi_i(t)\varphi_j(t) = \begin{cases} \varphi_i(t) & \text{si } i = j \\ 0 & \text{Sinon} \end{cases} \quad (3.5)$$

et :

$$\int_{t_0}^{t_f} \varphi_i(t)\varphi_j(t)dt = \begin{cases} h & \text{si } i = j \\ 0 & \text{Sinon} \end{cases} \quad (3.6)$$

où :

$$h = \frac{t_f - t_0}{m} \quad \text{et} \quad \varphi_m(t_f) = 1 \quad (3.7)$$

Le système de BPFs est donc orthogonal dans l'intervalle $[t_0 \quad t_f]$. Par conséquent tout élément $x(t) \in L^2_{[t_0 t_f]}$ peut être développé en série de BPFs :

$$x(t) \approx x_m(t) = \sum_{i=1}^m C_i^x \varphi_i(t) \quad (3.8)$$

Il est démontré que, dans $L^2_{[t_0 t_f]}$, $x(t)$ converge vers $x_m(t)$ lorsque m tend vers l'infini.

Si $x(t)$ est donnée sous forme analytique, C_i^x peut être évalué comme étant la valeur moyenne de $x(t)$ dans l'intervalle $[(i-1)h \quad ih]$. Donc :

$$C_i^x = h \int_{(i-1)h}^{ih} x(t)dt \quad (3.9)$$

Dans le cas où $x(t)=1$ dans l'intervalle $[t_0 \quad t_f]$, il est clair que :

$$1 = \sum_{i=1}^m \varphi_i(t) \quad (3.10)$$

- *Propriétés d'Intégration :*

Elles sont d'une importance capitale pour l'intégration des équations d'état et de la matrice de transition d'état. Ces intégrales sont :

$$\int_{t_0}^t \varphi_i(t) dt = \frac{h}{2} \varphi_i(t) + h \sum_{j=i+1}^m \varphi_j(t) \quad i=1,2,\dots,m \quad (3.11)$$

$$\int_{t_f}^t \varphi_i(t) dt = -\frac{h}{2} \varphi_i(t) - h \sum_{j=1}^{i-1} \varphi_j(t) \quad i=1,2,\dots,m \quad (3.12)$$

- Propriété De Multiplication Et De Division

Soient $x(t)$ et $y(t)$ deux éléments de $L^2_{[t_0 t_f]}$, vue la propriété d'orthogonalité des BPFs indiquée par (3.5) on a :

$$x(t)y(t) \approx \left[\sum_{i=1}^m C_i^x \varphi_i(t) \right] \left[\sum_{j=1}^m C_j^y \varphi_j(t) \right] = \left[\sum_{i=1}^m C_i^x C_i^y \varphi_i(t) \right] \quad (3.13)$$

$$\frac{x(t)}{y(t)} = \frac{\sum_{i=1}^m C_i^x \varphi_i(t)}{\sum_{j=1}^m C_j^y \varphi_j(t)} = \sum_{i=1}^m \frac{C_i^x}{C_i^y} \varphi_i(t), \quad C_i^y \neq 0 \quad (3.14)$$

- Relations Matricielles

Supposons que tous les éléments des matrices $A(t)=[a_{ij}(t)]_{n \times p}$ et $B(t)=[b_{ij}(t)]_{p \times r}$ appartiennent à $L^2_{[t_0 t_f]}$. Alors :

$$a_{ij}(t) = \sum_{k=1}^m C_k^{a_{ij}} \varphi_k(t) \quad (3.15)$$

$$b_{ij}(t) = \sum_{k=1}^m C_k^{b_{ij}} \varphi_k(t) \quad (3.16)$$

Par ailleurs on a :

$$[a_{ij}(t)]_{n \times p} = \sum_{k=1}^m A_k \varphi_k(t) \quad (3.17)$$

$$[b_{ij}(t)]_{p \times r} = \sum_{k=1}^m B_k \varphi_k(t) \quad (3.18)$$

où, $A_k=[C_k^{a_{ij}}]_{n \times p}$ et $B_k=[C_k^{b_{ij}}]_{p \times r}$.

1°) Transposée de la matrice A peut être développée en série de BPFs comme suit

$$A^T(t) = \left[\sum_{k=1}^m A_k \varphi_k(t) \right]^T = \sum_{k=1}^m A_k^T \varphi_k(t) \quad (3.19)$$

2°) De même la multiplication de deux matrices A et B peut être rendue grâce aux BPFs à :

$$A(t)B(t) = \left[\sum_{k=1}^m A_k \varphi_k(t) \right] \left[\sum_{k=1}^m B_k \varphi_k(t) \right] = \sum_{k=1}^m A_k B_k \varphi_k(t) \quad (3.20)$$

3°) Soit C une matrice inversible de dimension $n \times n$, l'approximation de l'inverse de cette matrice est de la forme suivante :

$$C^{-1}(t) = \left[\sum_{k=1}^m C_k \varphi_k(t) \right]^{-1} = \sum_{k=1}^m C_k^{-1} \varphi_k \quad (3.21)$$

3.3.2 Etude Du Problème De La Commande Optimale En Boucle Fermée

La loi de commande optimale en boucle fermée peut être obtenue, dans le cas des systèmes linéaires à critère quadratique, en résolvant l'équation de Riccati ou en déterminant la matrice de transition du système Hamiltonien [2, 3]. Cependant il n'est pas possible d'obtenir, analytiquement, la solution exacte des problèmes de commande optimale non linéaire ou à coefficients variables. Néanmoins la commande sous optimale en boucle fermée peut être obtenue en utilisant diverses techniques [3, 14, 16, 18, 19, 21, 60, 63]. D'autre part, durant les deux dernières décennies les fonctions orthogonales ont été intensivement utilisées pour déterminer la commande optimale des systèmes linéaires quadratiques en boucle fermée. Par exemple, [61] a utilisé les séries de Walsh, [66] a utilisé les polynômes de Chebyshev de premier type, [62, 64, 69] ont utilisé les polynômes de Chebyshev du second type, [67] a utilisé les BPFs, [87] a utilisé les séries de Fourier. Toutes ces méthodes se basent, pour trouver la solution, sur la matrice opérationnelle d'intégration qui permet de transformer le système Hamiltonien en équations algébriques. Le but de ce chapitre est d'utiliser les BPFs pour déterminer la loi de commande optimale en boucle fermée des systèmes linéaires non stationnaires à critère quadratique (Extension de la méthode décrite dans [67]).

L'objectif de ce problème est de déterminer une commande $u^*(x, t)$ qui minimise la fonctionnelle

$$J = \frac{1}{2} \int_0^{t_f} (x^T Q x + u^T R u) dt \quad (3.22)$$

pour le système régi par les équations d'état suivantes :

$$\dot{x} = Ax + Bu \quad (3.23)$$

sachant que

$$x(t_0 = 0) = x_0 \quad (3.24)$$

où $x(t) \in R^n, u(t) \in R^r$, A est une matrice de dimension $n \times n$, B est une matrice de dimension $n \times r$, Q est une matrice semi définie positive de dimension $n \times n$, R est une matrice définie positive de dimension $r \times r$.

Il faut signaler que le problème que nous allons étudier est un problème non stationnaire, par conséquent il ne possède pas, dans le cas général, une solution analytique.

Les conditions d'optimalité, pour ce problème, sont données par les relations suivantes :

$$\frac{\partial H}{\partial x} = Qx - A^T \lambda = \dot{\lambda} \quad (3.24)$$

$$\frac{\partial H}{\partial u} = Ru - B^T \lambda = 0 \quad (3.25)$$

où H est le Hamiltonien, et est donné par :

$$H = \frac{1}{2} [x^T Q x + u^T R u] - \lambda^T [Ax + Bu] \quad (3.26)$$

Donc, d'après (3.25), la relation suivante est déduite

$$u = R^{-1} B \lambda \quad (3.27)$$

La variable $\lambda(t)$ est un vecteur de dimension n satisfaisant le système d'équations vectorielles suivant :

$$\dot{\lambda} = Ax + BR^{-1} B^T \lambda \quad (3.28)$$

$$\dot{\lambda} = Qx - A^T \lambda \quad (3.29)$$

sachant que

$$x(0) = x_0 \text{ et } \lambda(t_f) = 0 \quad (3.30)$$

Soit $\psi(t_f, t)$ la matrice de transition d'état relative au système (3.28)-(3.29)-(3.30), qui permet d'écrire :

$$\psi(t_f, t) \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} x(t_f) \\ \lambda(t_f) \end{bmatrix} = \begin{bmatrix} x(t_f) \\ 0 \end{bmatrix} \quad (3.31)$$

La matrice de transition d'état est une matrice $2n \times 2n$ telle que $\psi(t_f, t_f) = I_{2n}$ (I_{2n} est la matrice identité de dimension $2n$).

Ecrivons la matrice de transition d'état sous la forme compactée suivante :

$$\psi(t_f, t) = \begin{bmatrix} \psi_{11}(t_f, t) & \psi_{12}(t_f, t) \\ \psi_{21}(t_f, t) & \psi_{22}(t_f, t) \end{bmatrix} \quad (3.32)$$

où chaque élément de cette matrice est une matrice de dimension $n \times n$

De (3.31) et (3.32) nous déduisons la relation donnant $\lambda(t)$ en fonction de $x(t)$ comme suit :

$$\psi_{21}(t_f, t)x(t) + \psi_{22}(t_f, t)\lambda(t) = \lambda(t_f) = 0 \quad (3.33)$$

d'où

$$\lambda(t) = -\psi_{22}^{-1}(t_f, t)\psi_{21}(t_f, t)x(t) \quad (3.34)$$

En substituant (3.34) dans (3.27) on obtient la loi de commande optimale en boucle fermée.

$$u(t) = -k(t)x(t) \quad (3.35)$$

avec :

$$k(t) = R^{-1}(t)B^T(t)\psi_{22}^{-1}(t_f, t)\psi_{21}(t_f, t) \quad (3.36)$$

On remarque bien que si $R(t)$ et $B(t)$ sont données, il suffit que soient trouvées les matrices $\psi_{21}(t_f, t)$ et $\psi_{22}(t_f, t)$ pour que $k(t)$ puisse être déterminé.

En dérivant (3.31) par rapport à t , on a :

$$\frac{\partial}{\partial t} \left\{ \psi(t_f, t) \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} \right\} = \psi(t_f, t) \begin{bmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{bmatrix} + \dot{\psi}(t_f, t) \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = 0 \quad (3.37)$$

et en utilisant (3.28) et (3.29) dans (3.37), on obtient :

$$\psi(t_f, t) \begin{bmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{bmatrix} = \psi(t_f, t) F(t) \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = -\dot{\psi}(t_f, t) \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} \quad (3.38)$$

où

$$F(t) = \begin{bmatrix} A(t) & B(t)R^{-1}(t)B^T(t) \\ Q(t) & -A^T(t) \end{bmatrix} \quad (3.39)$$

De (3.38), par simplification, on en déduit

$$\dot{\psi}(t_f, t) = -\psi(t_f, t) F(t) \quad (3.40)$$

En intégrant (3.40) de t_f à t , on écrit

$$I_{2n} - \psi(t_f, t) = \int_{t_f}^t \psi(t_f, t') F(t') dt' \quad (3.41)$$

Supposons maintenant que tous les éléments des matrices $A(t)=[a_{ij}(t)]_{n \times n}$, $B(t)=[b_{ij}(t)]_{n \times r}$, $R(t)=[r_{ij}(t)]_{r \times r}$, $Q(t)=[q_{ij}(t)]_{n \times n}$, $F(t)=[f_{ij}(t)]_{2n \times 2n}$ et $\psi(t_f, t)=[\psi_{ij}(t_f, t)]_{2n \times 2n}$ appartiennent à $L^2_{[t_0, t_f]}$ et ainsi peuvent tous être développés en séries de BPFs, ce qui mènent aux formules suivantes :

$$R(t) = [r_{ij}(t)]_{r \times r} \cong \sum_{k=1}^m R_k \phi_k(t) \quad (3.42)$$

$$Q(t) = [q_{ij}(t)]_{n \times n} \cong \sum_{k=1}^m Q_k \phi_k(t) \quad (3.43)$$

$$F(t) = [f_{ij}(t)]_{n \times n} \cong \sum_{k=1}^m F_k \phi_k(t) \quad (3.44)$$

$$\psi(t_f, t) = [\psi_{ij}(t_f, t)]_{n \times n} \cong \sum_{k=1}^m \psi_k \phi_k(t) \quad (3.45)$$

où

$$\psi_k = [\psi_{ijk}]_{2n \times 2n} = \begin{bmatrix} \psi_{11k} & \psi_{12k} \\ \psi_{21k} & \psi_{22k} \end{bmatrix} = \begin{bmatrix} \psi_{11k} & \psi_{1nk} & \psi_{1,n+1,k} & \psi_{1,2n,k} \\ \psi_{n1k} & \psi_{nnk} & \psi_{n,n+1,k} & \psi_{n,2n,k} \\ \psi_{n+1,1,k} & \psi_{n+1,n,k} & \psi_{n+1,n+1,k} & \psi_{n+1,2n,k} \\ \psi_{2n,1,k} & \psi_{2n,n,k} & \psi_{2n,n+1,k} & \psi_{2n,2n,k} \end{bmatrix} \quad (3.46)$$

La propriété d'orthogonalité permet d'écrire :

$$F(t)\psi(t_f, t) \cong \sum_{k=1}^m F_k \psi_k \phi_k(t) \quad (3.47)$$

En insérant (3.45) et (3.47) dans (3.41), on obtient :

$$\sum_{k=1}^m [I_{2n} - \psi_k] \phi_k(t) = \sum_{k=1}^m \psi_k F_k \int_{t_f}^t \phi_k(t') dt' \quad (3.48)$$

La propriété d'intégration (3.12) permet de mettre (3.48) sous la forme suivante :

$$\sum_{k=1}^m [I_{2n} - \psi_k] \phi_k(t) = -h \sum_{k=1}^m \psi_k F_k \left[\sum_{l=1}^m \phi_l(t) + \frac{1}{2} \phi_k(t) \right] \quad (3.49)$$

En égalisant les coefficients associés à chaque ϕ_k des deux membres de (3.49) on obtient la relation récursive suivante :

$$I_{2n} - \psi_m = -\frac{h}{2} \psi_m F_m \quad (3.50)$$

$$I_{2n} - \psi_k = -\frac{h}{2} \psi_k F_k - h \sum_{l=k-1}^m \psi_l F_l \quad k=m-1, m-2, \dots, 2, 1 \quad (3.51)$$

Enfin, de (3.50) et (3.51) on en déduit la relation récursive permettant la détermination de ψ_k

$$\psi_m = \left[I_{2n} - \frac{h}{2} F_m \right]^{-1} \quad (3.52)$$

$$\psi_k = \psi_{k+1} \left[I_{2n} + \frac{h}{2} F_{k+1} \right] \left[I_{2n} - \frac{h}{2} F_k \right]^{-1} \quad k=m-1, m-2, \dots, 2, 1 \quad (3.53)$$

En substituant les approximations de A , R , $\psi_{21}(t_f, t)$ et $\psi_{22}(t_f, t)$ dans (3.36), on obtient l'expression approchée de $k(t)$ comme suit :

$$\sum_{k=1}^m k_k \varphi_k(t) = \left[\sum_{k=1}^m R_k^{-1} \varphi_k(t) \right] \left[\sum_{l=1}^m B_l^T \varphi_l(t) \right] \left[\sum_{i=1}^m \psi_{22i}^{-1} \varphi_i(t) \right] \left[\sum_{j=1}^m \psi_{21j} \varphi_j(t) \right] \quad (3.54)$$

ou , en utilisant la propriété d'orthogonalité.

$$\sum_{k=1}^m k_k \varphi_k(t) = \sum_{k=1}^m R_k^{-1} B_k^T \psi_{22k}^{-1} \psi_{21k} \quad (3.55)$$

Par conséquent le gain k_k est obtenu à partir de (3.54) comme suit :

$$k_k = R_k^{-1} B_k^T \psi_{22k}^{-1} \psi_{21k} \quad (3.56)$$

L'exemple ci dessous [95], illustre de façon claire le fonctionnement de l'algorithme établi, et connaissant la solution analytique du problème, on peut mettre en valeur la méthode élaborée.

Exemple 3.1

Considérons le système unidimensionnel à coefficients variables régi par l'équation différentielle suivante :

$$\dot{x}(t) = tx(t) + u(t) \quad x(0) = 1 \quad (3.57)$$

Pour ce système, il faut trouver le gain $k(t)$ de la commande optimale en boucle fermée tel que le critère de performance quadratique :

$$J = \frac{1}{2} \int_0^1 [x^2(t) + u^2(t)] dt \quad (3.58)$$

soit minimal.

Il est facile de déduire les paramètres suivants: $n=1$, $t_f=1$, $A(t)=t$, $B(t)=1$, $Q(t)=1$ et $R(t)=1$.

Alors :

$$F(t) = \begin{bmatrix} A(t) & B(t)R^{-1}(t)B^T(t) \\ Q(t) & -A^T(t) \end{bmatrix}_{2 \times 2} = \begin{bmatrix} t & 1 \\ 1 & -t \end{bmatrix} \quad (3.59)$$

A titre illustratif, essayons de résoudre ce problème pour $m=4$, donc pour $h=\frac{t_f}{m}=\frac{1}{4}$.

L'approximation de $A(t)=t$ est donné par :

$$A(t) \cong \sum_{k=1}^m A_k \phi_k(t) \quad (3.60)$$

où les coefficients A_k sont donnés par la relation, suivante :

$$A_k = \frac{1}{h} \int_{(k-1)h}^{kh} A(t) dt = \frac{1}{h} \int_{(k-1)h}^{kh} t dt = \frac{h}{2} (2k-1) = \frac{1}{8} (2k-1) \quad (3.61)$$

A partir de (3.61), formons les matrices F_k .

$$F_1 = \begin{bmatrix} \frac{1}{8} & 1 \\ 1 & -\frac{1}{8} \end{bmatrix}, F_2 = \begin{bmatrix} \frac{3}{8} & 1 \\ 1 & -\frac{3}{8} \end{bmatrix}, F_3 = \begin{bmatrix} \frac{5}{8} & 1 \\ 1 & -\frac{5}{8} \end{bmatrix}, F_4 = \begin{bmatrix} \frac{7}{8} & 1 \\ 1 & -\frac{7}{8} \end{bmatrix} \quad (3.62)$$

(3.52) et (3.53) permet de déterminer les matrices ψ_k de la manière suivante :

$$\psi_4 = \left[I_2 - \frac{h}{2} F_4 \right]^{-1} = \left[I_2 - \frac{1}{8} F_4 \right]^{-1} \quad (3.63)$$

Ce qui donne

$$\psi_4 = \begin{bmatrix} 1.1408 & 0.1285 \\ 0.1285 & 0.9159 \end{bmatrix} = \begin{bmatrix} \psi_{114} & \psi_{124} \\ \psi_{214} & \psi_{224} \end{bmatrix} \quad (3.64)$$

De même ψ_3 peut être déterminé de la relation suivante :

$$\psi_3 = \psi_4 \left[I_2 + \frac{1}{8} F_4 \right] \left[I_2 - \frac{1}{8} F_3 \right]^{-1} \quad (3.65)$$

Ceci permet de déduire la matrice ψ_3

$$\psi_3 = \begin{bmatrix} 1.4454 & 0.4060 \\ 0.3896 & 0.8167 \end{bmatrix} = \begin{bmatrix} \psi_{113} & \psi_{123} \\ \psi_{213} & \psi_{223} \end{bmatrix} \quad (3.66)$$

et la matrice ψ_2 est calculée a partir de l'expression suivante :

$$\psi_2 = \psi_3 \left[I_2 + \frac{1}{8} F_3 \right] \left[I_2 - \frac{1}{8} F_2 \right]^{-1} \quad (3.67)$$

L'application numérique de cette formule donne :

$$\psi_2 = \begin{bmatrix} 1.7857 & 0.7434 \\ 0.6586 & 0.8443 \end{bmatrix} = \begin{bmatrix} \psi_{112} & \psi_{122} \\ \psi_{212} & \psi_{222} \end{bmatrix} \quad (3.68)$$

Finalemment la matrice ψ_1 est déduite de la formule suivante :

$$\psi_1 = \psi_2 \left[I_2 + \frac{1}{8} F_2 \right] \left[I_2 - \frac{1}{8} F_1 \right]^{-1} \quad (3.69)$$

ce qui permet d'écrire

$$\psi_1 = \begin{bmatrix} 2.1434 & 1.1812 \\ 0.9331 & 0.9883 \end{bmatrix} = \begin{bmatrix} \psi_{111} & \psi_{121} \\ \psi_{211} & \psi_{221} \end{bmatrix} \quad (3.70)$$

La relation (3.55) permet d'écrire le gain en boucle fermé $k(t)$ sous la forme suivante :

$$k(t) = \sum_{k=1}^4 R_k^{-1} B_k^T \psi_{22k}^{-1} \psi_{21k} \quad (3.71)$$

et comme $R_k = B_k = 1$ pour $k = 1, 2, 3, \dots, m$ la formule (3.71) s'écrira :

$$k(t) = \frac{\psi_{211}}{\psi_{221}} \varphi_1 + \frac{\psi_{212}}{\psi_{222}} \varphi_2 + \frac{\psi_{213}}{\psi_{223}} \varphi_3 + \frac{\psi_{214}}{\psi_{224}} \varphi_4 \quad (3.72)$$

ce qui donne, d'après les résultats ci dessus

$$k(t) = 0.9442\varphi_1 + 0.7800\varphi_2 + 0.4771\varphi_3 + 0.1403\varphi_4 \quad (3.73)$$

Par la même méthode nous avons pu élaborer les résultats relatifs à $m = 10$. Ces résultats sont résumés dans le tableau 3.1.

	ψ_{11k}	ψ_{12k}	ψ_{21k}	ψ_{22k}	$\frac{\psi_{21k}}{\psi_{22k}}$
$k=1$	2.2381	1.3217	0.9995	1.0382	0.9628
$k=2$	2.0949	1.1167	0.8908	0.9534	0.9343
$k=3$	1.9522	0.9344	0.7822	0.8880	0.8809
$k=4$	1.8108	0.7714	0.6742	0.8410	0.8017
$k=5$	1.6717	0.6249	0.5670	0.8119	0.6984
$k=6$	1.5360	0.4921	0.4609	0.8008	0.5756
$k=7$	1.4050	0.3706	0.3562	0.8083	0.4407
$k=8$	1.2799	0.2581	0.2530	0.8354	0.3028
$k=9$	1.1620	0.1521	0.1511	0.8841	0.1709
$k=10$	1.0525	0.0502	0.0502	0.9571	0.0525

Tableau 3.1 Valeurs des ψ_{ijk} et $\frac{\psi_{ijk}}{\psi_{22k}}$ en fonction de k

A partir de ce tableau, le gain $k(t)$ peut être exprimé comme suit :

$$k(t) \cong \sum_{k=1}^{10} \frac{\psi_{21k}}{\psi_{22k}} \varphi_k = 0.9628\varphi_1 + 0.9343\varphi_2 + 0.8809\varphi_3 + 0.8017\varphi_4 + 0.6984\varphi_5 + \\ + 0.5756\varphi_6 + 0.4407\varphi_7 + 0.3028\varphi_8 + 0.1709\varphi_9 + 0.0525\varphi_{10} \quad (3.74)$$

Essayons maintenant de trouver l'approximation de $x(t)$. Pour cela reprenons l'équation d'état du système.

$$\dot{x}(t) = tx(t) + u(t) \quad (3.75)$$

Mais, il est établi à partir de (3.35) que :

$$u(t) = -k(t)x(t) \quad (3.76)$$

Introduisons maintenant (3.76) dans (3.75), il s'ensuit

$$\dot{x}(t) = tx(t) - k(t)x(t) = [t - k(t)]x(t) \quad (3.77)$$

Intégrons maintenant (3.77), ce qui permet d'écrire

$$x(t) - x(0) = \int_0^t [t - k(t)]x(t) \quad (3.78)$$

L'approximation de (3.78) via les BPFs donne

$$\sum_{k=1}^m x_k \varphi_k(t) - \sum_{k=1}^m x(0) \varphi_k = \int_0^1 \sum_{k=1}^m [C_k^t - k_k] x_k \varphi_k \quad (3.79)$$

Utilisons la propriété d'intégration (3.11), la relation (3.19) devient donc

$$\sum_{k=1}^m x_k \varphi_k(t) - \sum_{k=1}^m x(0) \varphi_k(t) = h \sum_{k=1}^m [C_k^t - k_k] x_k \left\{ \frac{1}{2} \varphi_k(t) + \sum_{l=k+1}^m \varphi_l(t) \right\} \quad (3.80)$$

ce qui permet d'établir la relation récursive suivante permettant le calcul de $x(t)$

$$x_1 - x(0) = \frac{1}{2} h [C_1^t - k_1] x_1 \quad (3.81)$$

$$x_k - x(0) = \frac{1}{2} h [C_k^t - k_k] x_k + h \sum_{l=1}^{k-1} [C_l^t - k_l] x_l \quad (3.82)$$

ou sous une autre forme plus appropriée

$$x_1 = \left[1 - \frac{1}{2} h (C_1^t - k_1) \right]^{-1} x_0 \quad (3.83)$$

$$x_{k+1} = \left[1 - \frac{1}{2} h (C_{k+1}^t - k_{k+1}) \right]^{-1} \left[1 + \frac{1}{2} h (C_k^t - k_k) \right] x(k) \quad (3.84)$$

Le tableau 3.2 indique à titre indicatif les coefficients x_i et les paramètres u_i (déduits à partir de la relation (3.76)) correspondants à $m=4$. De même les mêmes paramètres correspondants à $m=10$ sont présentés dans le tableau 3.3.

	$k=1$	$k=2$	$k=3$	$k=4$
x_i	0.9071	0.7750	0.7496	0.8407
u_i	-0.8565	-0.6045	-0.3576	-0.1180

Tableau 3.2 Valeurs des x_i et des u_i en fonction de k

	x_i	u_i
$k=1$	0.9564	-0.9208
$k=2$	0.8783	-0.8206
$k=3$	0.8180	-0.7206
$k=4$	0.7747	-0.6211
$k=5$	0.7479	-0.5224
$k=6$	0.7377	-0.4246
$k=7$	0.7445	-0.3281
$k=8$	0.7695	-0.2330
$k=9$	0.8144	-0.1392
$k=10$	0.8816	-0.0463

Tableau 3.3 Valeurs des x_i et des u_i relatives à $m=10$

A partir de ces deux tableaux, nous pouvons déduire les expressions approchées de $x(t)$ et de $u(t)$ relatives ,respectivement, à $m=4$ et $m=10$. Elles ont les formes ci dessus.

$$x(t) \cong x_4(t) = 0.9071\varphi_1(t) + 0.7750\varphi_2(t) + 0.7496\varphi_3(t) + 0.8407\varphi_4(t) \quad (3.85)$$

$$u(t) \cong u_4(t) = -0.8565\varphi_1(t) - 0.6045\varphi_2(t) - 0.3576\varphi_3(t) - 0.1180\varphi_4(t) \quad (3.86)$$

$$x(t) \cong x_{10}(t) = 0.9564\varphi_1(t) + 0.8783\varphi_2(t) + 0.8180\varphi_3(t) + 0.7747\varphi_4(t) + 0.7479\varphi_5 \\ + 0.7377\varphi_6(t) + 0.7445\varphi_7(t) + 0.7695\varphi_8(t) + 0.8144\varphi_9(t) + 0.8816\varphi_{10} \quad (3.87)$$

$$u(t) \cong u_{10}(t) = -0.9208\varphi_1(t) - 0.8206\varphi_2(t) - 0.7206\varphi_3(t) - 0.6211\varphi_4(t) - 0.5224\varphi_5 \\ - 0.4246\varphi_6(t) - 0.3281\varphi_7(t) - 0.2330\varphi_8(t) - 0.1392\varphi_9(t) - 0.0463\varphi_{10} \quad (3.88)$$

Les fonctions $k_i(t)$, $x_i(t)$ et $u_i(t)$ relatives respectivement à $i=4$ et $i= 5$ sont représentées par les figures 3.1, 3.2, 3.3, 3.4, 3.5 et 3.6.

Remarques :

1°) Pour l'exemple précédent ; il est possible de trouver une solution analytique du gain $k(t)$. Pour cela il faut résoudre l'équation de Riccati suivante :

$$\dot{k}(t) = k^2(t) - 2tk(t) - 1 \quad k(1)=0 \quad (3.89)$$

La représentation du gain selon la formule (3.89) permet d'apprécier la méthode étudiée.

2°) Les valeurs de l'indice de performance calculé, respectivement, pour $m=4$ et $m=10$ sont : $J=0.4916$ et $J=0.4854$. les deux valeurs ne diffèrent que de peu l'un par rapport à l'autre, ce qui laisse à dire que la méthode étudiée constitue une bonne approximation du problème étudié même pour des valeurs faibles de m .

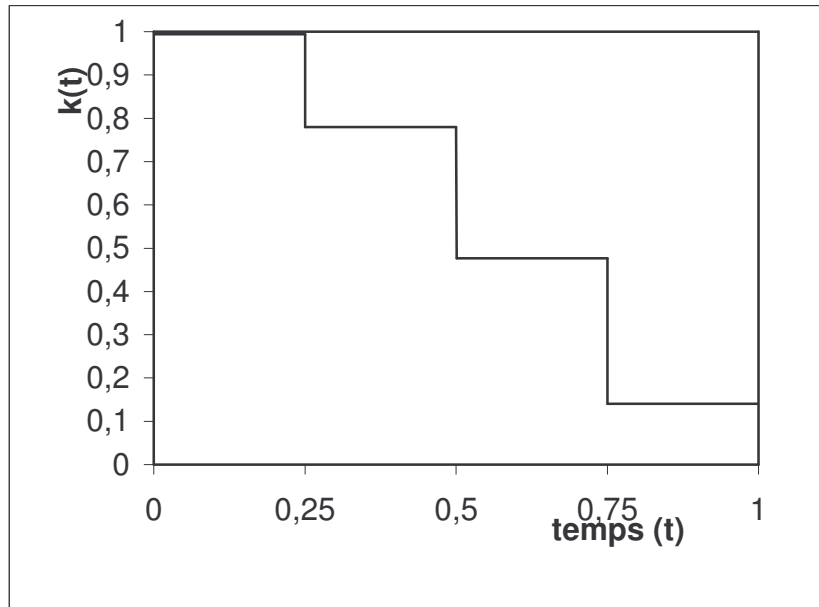


Fig. 3.1 Variation de $k(t)$ en fonction du temps pour $m=4$

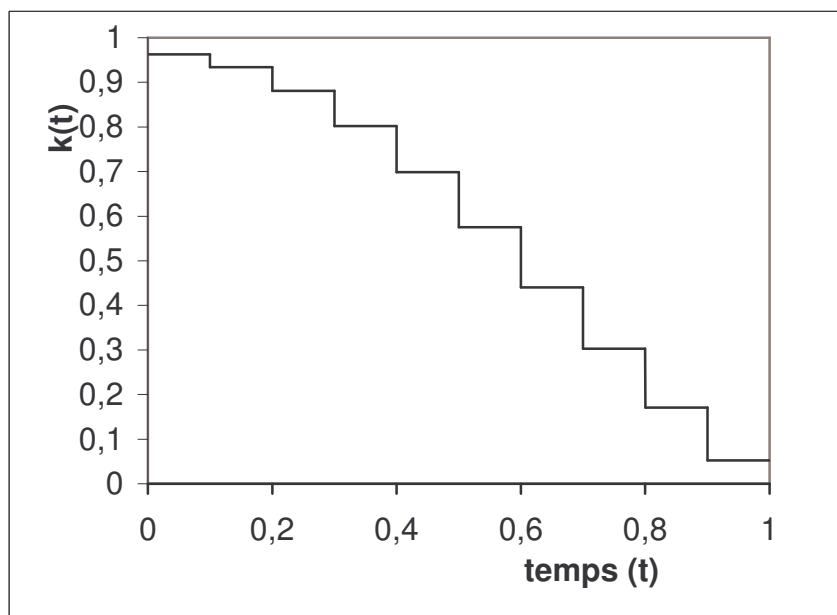
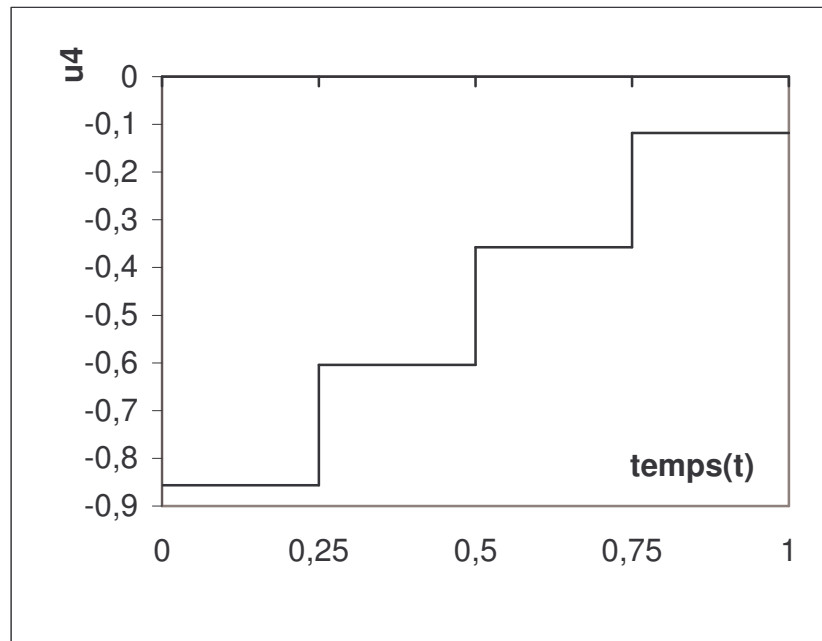
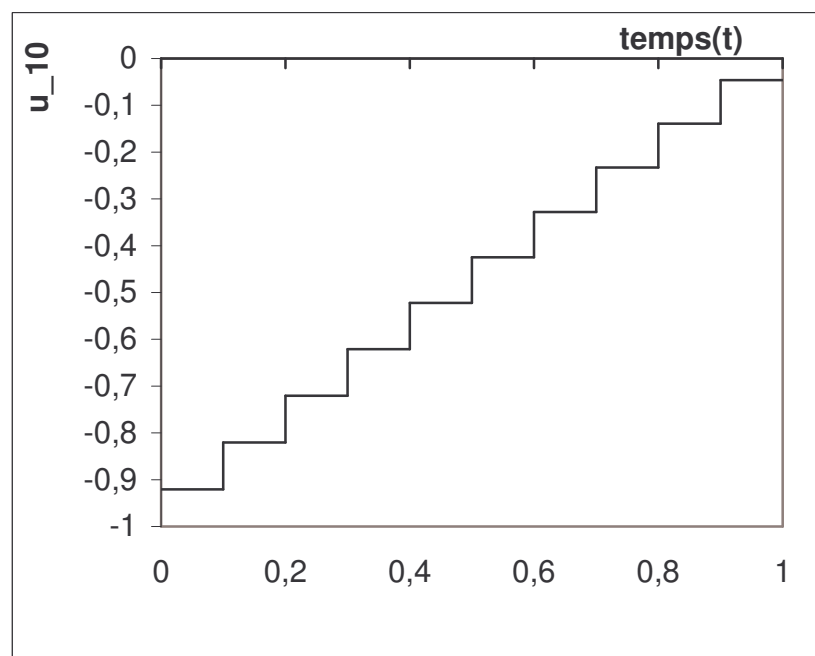
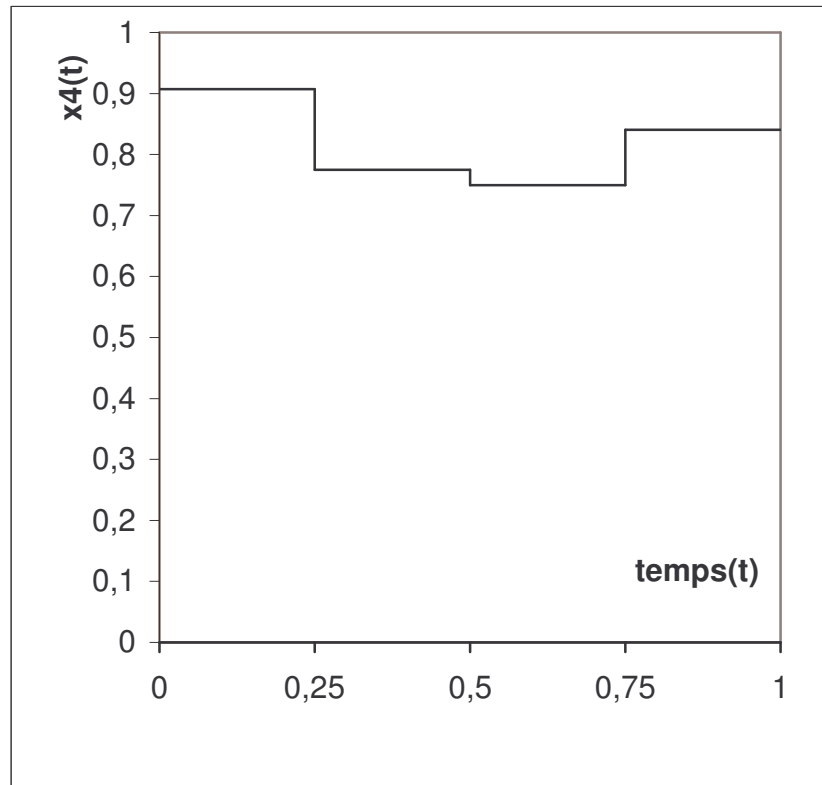
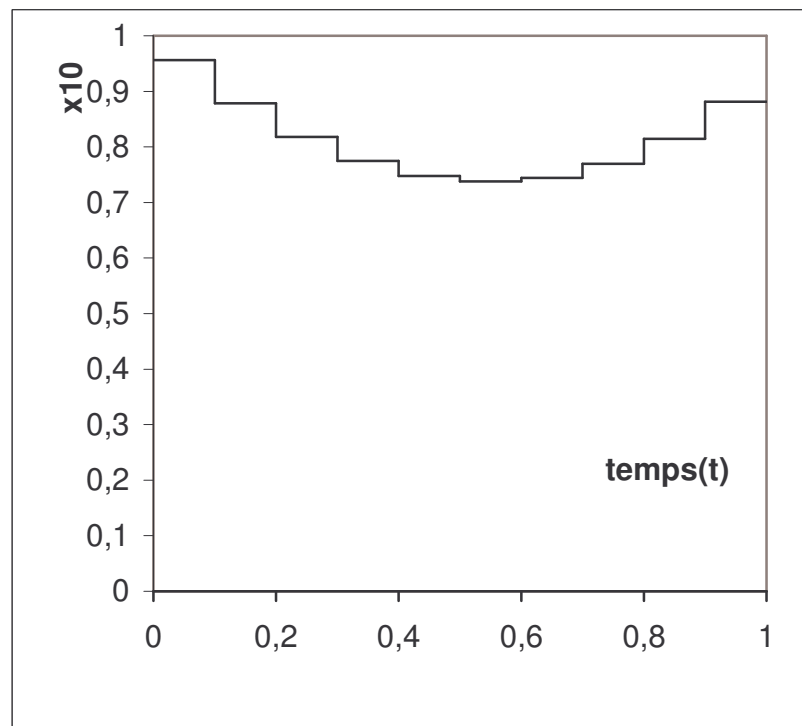


Fig. 3.2 Variation de $k(t)$ en fonction du temps pour $m=10$

Fig. 3.3 Variation de $u(t)$ en fonction du temps pour $m=4$ Fig. 3.4 Variation de $u(t)$ en fonction du temps pour $m=10$

Fig. 3.5 Variation de $x(t)$ en fonction du temps pour $m=4$ Fig. 3.6 Variation de $x(t)$ en fonction du temps pour $m=10$

3.3.3 Etude De La Commande Optimale En Boucle Ouverte Sans Contraintes

Dans ce cas, le problème peut s'énoncer comme suit : Etant donné un système régi par (3.23) et (3.24), il s'agit de trouver la commande en boucle ouverte $u^*(t)$ sur $t_0 \leq t \leq t_f$ qui minimise l'indice de performance quadratique (3.22).

La méthode utilisée pour résoudre ce problème de la commande optimale est basée sur sa transformation directe en un problème de programmation quadratique en utilisant les fonctions à blocs d'impulsion. Cette méthode sera généralisée dans le chapitre suivant pour résoudre le problème de la commande optimale non linéaire.

En appliquant la propriété d'intégration (3.11) et la propriété de multiplication (3.20), l'équation d'état (3.23) se réduit à :

$$x_{1-}x(0) = \frac{1}{2}h(A_1x_1 + B_1u_1) \quad (3.90)$$

$$x_{k-}x(0) = \frac{1}{2}h(A_kx_k + B_ku_k) + h\sum_{l=0}^{k-1}(A_lx_l + B_lu_l) \quad k = 1, \dots, m \quad (3.91)$$

où A_k, B_k, x_k , et u_k sont donnés comme suit :

$$A_k = \begin{bmatrix} C_k^{a_{11}} & C_k^{a_{12}} & \cdot & \cdot & C_k^{a_{1n}} \\ C_k^{a_{21}} & C_k^{a_{22}} & \cdot & \cdot & C_k^{a_{2n}} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ C_k^{a_{n1}} & C_k^{a_{n2}} & \cdot & \cdot & C_k^{a_{nn}} \end{bmatrix}, \quad B_k = \begin{bmatrix} C_k^{b_{11}} & C_k^{b_{12}} & \cdot & \cdot & C_k^{b_{1r}} \\ C_k^{b_{21}} & C_k^{b_{22}} & \cdot & \cdot & C_k^{b_{2r}} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ C_k^{b_{n1}} & C_k^{b_{n2}} & \cdot & \cdot & C_k^{b_{nr}} \end{bmatrix},$$

$$x_k = \begin{bmatrix} C_k^{x_1} \\ C_k^{x_2} \\ \cdot \\ \cdot \\ C_k^{x_n} \end{bmatrix} \quad \text{et} \quad u_k = \begin{bmatrix} C_k^{u_1} \\ C_k^{u_2} \\ \cdot \\ \cdot \\ C_k^{u_r} \end{bmatrix}$$

Les $C_k^{a_{ij}}, C_k^{b_{ij}}, C_k^{x_i}$ et $C_k^{u_i}$ $k = 1, \dots, m$ sont les $k^{\text{ième}}$ coefficients de fourrier associés, respectivement, aux éléments a_{ij}, b_{ij}, x_i et u_i . Nous constatons que les n équations différentielles données par (3.23) sont transformés, grâce aux BPFs, à $n \times m$ équations algébriques données par (3.90)-(3.91).

De la même manière l'indice de performance (3.22) se réduit à une seule équation algébrique ayant la forme suivante :

$$J = \frac{1}{2} h \sum_{k=1}^m (x_k^T Q_k x_k + u_k^T R_k u_k) \quad (3.92)$$

Avec :

$$Q_k = \begin{bmatrix} C_k^{q_{11}} & C_k^{q_{12}} & \cdot & \cdot & C_k^{q_{1n}} \\ C_k^{q_{21}} & C_k^{q_{22}} & \cdot & \cdot & C_k^{q_{2n}} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ C_k^{q_{n1}} & C_k^{q_{n2}} & \cdot & \cdot & C_k^{q_{nn}} \end{bmatrix} \quad \text{et} \quad R_k = \begin{bmatrix} C_k^{r_{11}} & C_k^{r_{12}} & \cdot & \cdot & C_k^{r_{1r}} \\ C_k^{r_{21}} & C_k^{r_{22}} & \cdot & \cdot & C_k^{r_{2r}} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ C_k^{r_{r11}} & C_k^{r_{r2}} & \cdot & \cdot & C_k^{r_{rr}} \end{bmatrix}$$

Où les $C_k^{q_{ij}}$ et $C_k^{r_{ij}}, C_k^{x_i}$ $k = 1, \dots, m$ sont les $k^{\text{ième}}$ coefficients de Fourier correspondants, respectivement, aux éléments q_{ij} et r_{ij} des matrices Q et R .

Le problème, donc, de la commande optimale (3.22), (3.23) et (3.24) s'est réduit à un problème d'optimisation sans contraintes (3.90), (3.91) et (3.92).

Exemple 3.2 [88]

Pour le système décrit par les équations d'état suivantes :

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= -x_2 + u \end{aligned}$$

il s'agit de déterminer la commande $u(t)$ qui minimise le critère de performance quadratique suivant :

$$J = \int_0^1 (x_1^2 + x_2^2 + 0.005u^2) dt$$

avec $x_1(0) = 0$ et $x_2(0) = -1$

Il est bien connu que ce problème peut être réduit à un problème d'optimisation linéaire, et en vertu de (3.90), (3.91) et (3.92) il s'énonce comme suit : il s'agit de déterminer les coefficients de Fourier C_i^u , $C_i^{x_1}$ et $C_i^{x_2}$ qui minimisent le critère de performance :

$$J = h \sum_{i=1}^m \left((C_i^{x_1})^2 + (C_i^{x_2})^2 + 0.005(C_i^u)^2 \right)$$

pour le système décrit par les équations algébriques suivantes :

$$\begin{aligned} C_1^{x_1} - x_1(0) &= \frac{1}{2} h C_1^{x_2} \\ C_i^{x_1} - x_1(0) &= \frac{1}{2} h C_i^{x_2} + h \sum_{j=1}^{i-1} C_j^{x_2} \\ & \qquad \qquad \qquad i=1, \dots, m \\ C_1^{x_2} - x_2(0) &= \frac{1}{2} h (-C_1^{x_2} + C_1^u) \\ C_i^{x_2} - x_2(0) &= \frac{1}{2} h (-C_i^{x_2} + C_i^u) + h \sum_{j=1}^{i-1} (-C_j^{x_2} + C_j^u) \end{aligned}$$

La solution de ce problème pour $m=10$ donne les paramètres résumés dans le tableau 3.4.

	$C_k^{x_1}$	$C_k^{x_2}$	C_k^u
$k=1$	-0.0286	-0.5714	8.0003
$k=2$	-0.0599	-0.0551	1.6990
$k=3$	-0.0611	0.0305	-0.0115
$k=4$	-0.0585	0.0217	-0.1128
$k=5$	-0.0566	0.0174	0.0671
$k=6$	-0.0545	0.0241	0.1088
$k=7$	-0.0519	0.0274	0.0087
$k=8$	-0.0495	0.0200	-0.1111
$k=9$	-0.0483	0.0057	-0.1484
$k=10$	-0.0482	-0.0052	-0.0690

Tableau 3.4. Valeurs des éléments $C_k^{x_1}$, $C_k^{x_2}$ et C_k^u pour $m=10$

Les figures 3.7, 3.8, et 3.9 sont des représentations, respectivement, de $x_1(t)$, $x_2(t)$ et $u(t)$. A titre de comparaison nous donnons aussi les représentations des mêmes fonctions pour $m=50$ sur les figures 3.10, 3.11 et 3.12.

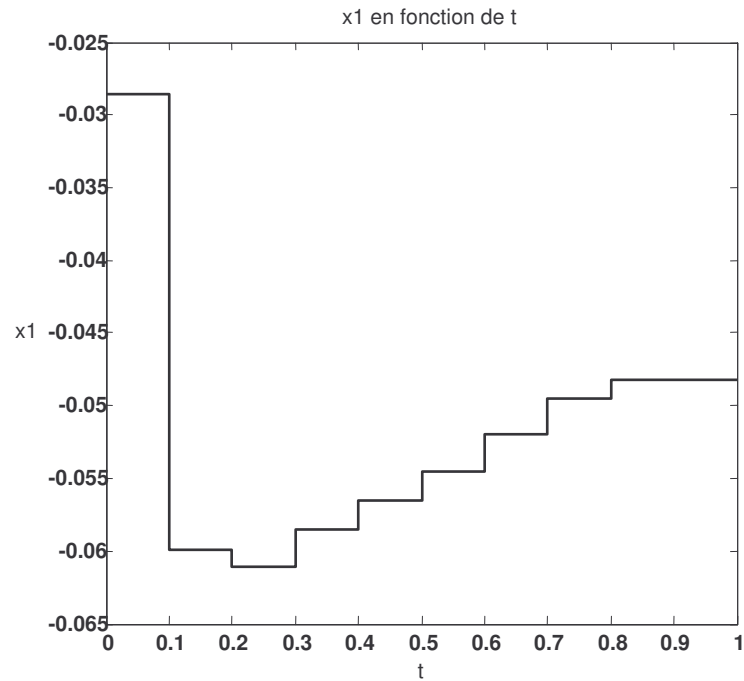


Fig. 3.7. Variation de $x_1(t)$ en fonction de t ($m=10$)

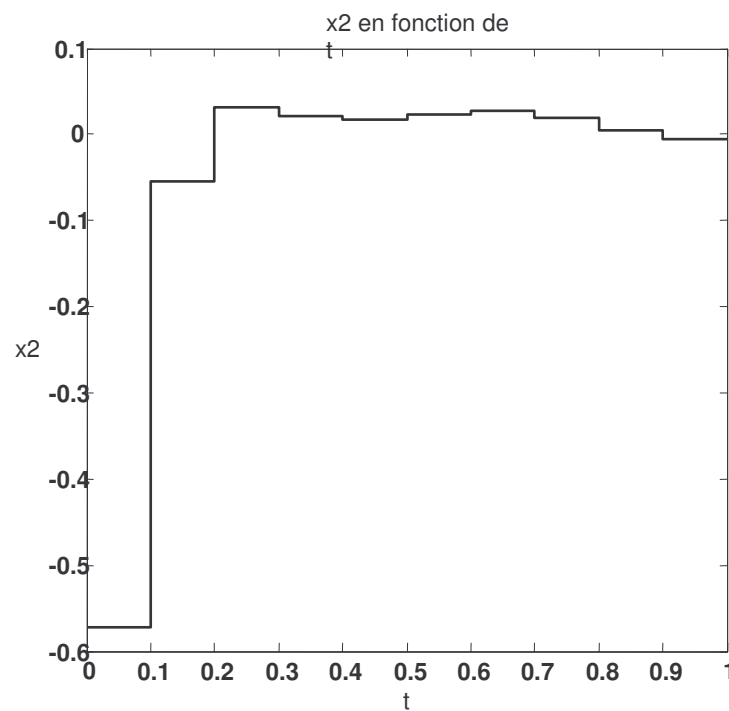


Fig. 3.8. Variation de $x_2(t)$ en fonction de t

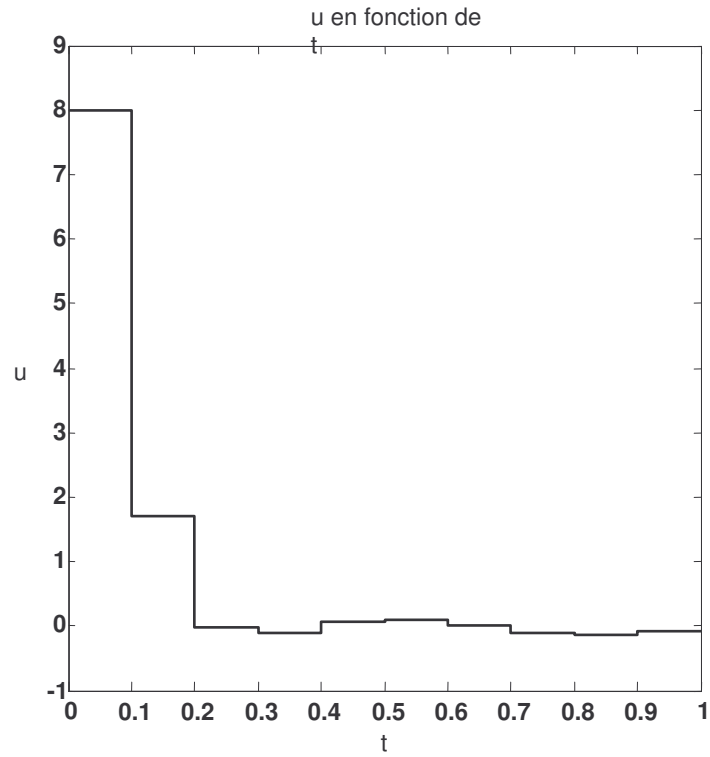


Fig. 3.9. Variation de u en fonction de t ($m=10$)

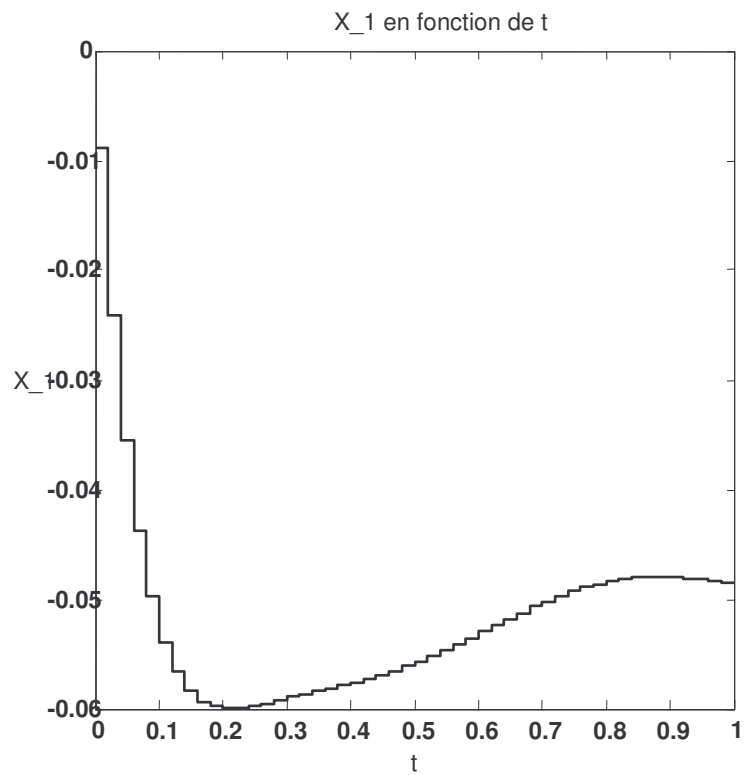


Fig. 3.10. Variation de $x_1(t)$ en fonction de t ($m=50$)

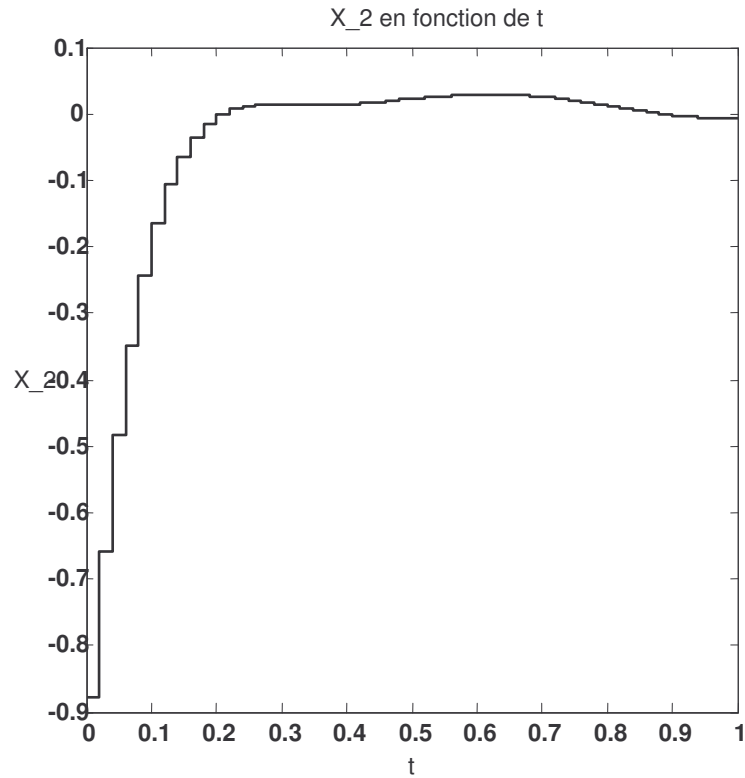


Fig. 3.11. Variation de $x_2(t)$ en fonction de t ($m=50$)

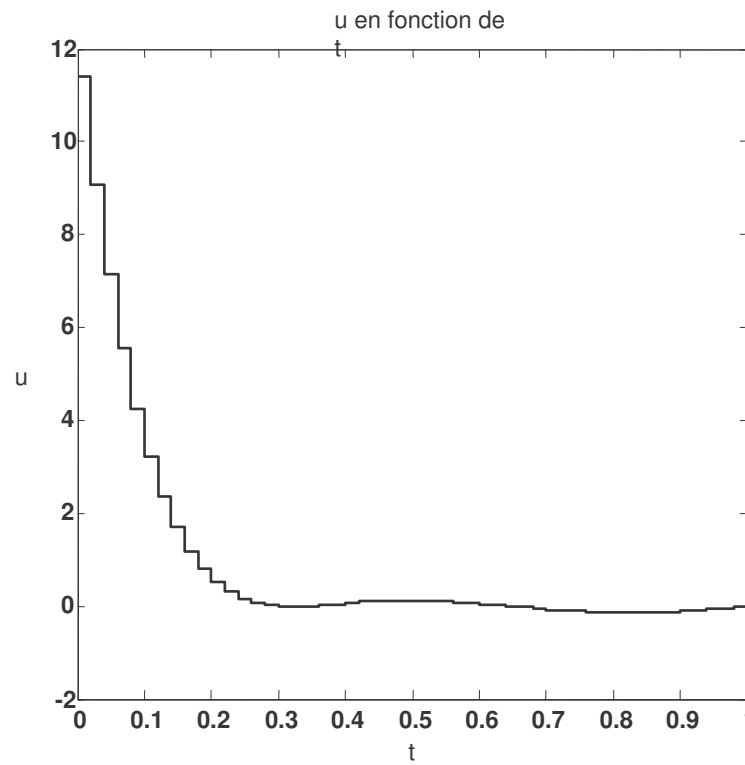


Fig. 3.12. Variation de u en fonction de t ($m=50$)

Cet exemple a été étudié par plusieurs auteurs, le tableau ci dessous donne la valeur de l'indice de performance selon la méthode utilisée.

Source	J	Erreur de déviation
Valeur exacte	0.06936094	0
Hsieh [51]	0.0702	8.4×10^{-4}
Neuman et Sen [52]		
N=4	0.0703	9.4×10^{-4}
N=5	0.06989	5.3×10^{-4}
Vlassenbroeck [40]		
N=5	0.07595	6.6×10^{-3}
N=9	0.069368	5.3×10^{-6}
Jaddu [88]		
N=5	0.07595646	6.59×10^{-3}
N=9	0.0693689	7.96×10^{-6}
Notre méthode		
$m=4$	0.0694	3.9×10^{-5}
$m=10$	0.0696	3.39×10^{-4}

Tableau 3.6 Valeurs de J de l'exemple relatives à la méthode utilisée

3.5 Etude De La Commande Optimale Linéaire Avec Contraintes

La méthode étudiée précédemment ne permet pas l'étude de la commande optimale des systèmes linéaires soumise à des contraintes. Ce type de problème est le plus fréquemment rencontré dans les applications. Parmi les contraintes rencontrées dans les applications courantes, nous pouvons mentionner : les contraintes de saturation sur la commande, les contraintes d'état à l'instant final, les contraintes d'égalité et d'inégalité sur les fonctions dépendant de l'état et de la commande et les contraintes du point intérieur. La présence de ces contraintes entraîne souvent des difficultés analytiques et numériques. Le traitement théorique des problèmes de la commande optimale en présence des contraintes peut être trouvé dans [3,4]. L'une des méthodes les plus utilisées pour résoudre les problèmes de commande optimale avec contraintes est de les convertir en un problème de programmation mathématique via les techniques de discrétisation

et de paramétrisation [33, 34, 40, 52, 75, 78]. La méthode que nous allons étudiée dans ce paragraphe permet de résoudre le problème de la commande optimale linéaire avec indice de performance quadratique soumis, contrairement à d'autre méthode [79], à divers types de contraintes. L'objectif de cette méthode est donc de transformer directement le problème de commande optimale à un problème de programmation linéaire dont les algorithmes ne manquent pas pour résoudre ce problème.

Le problème de la commande optimale linéaire avec contraintes étudié dans cette thèse s'énonce comme suit :

Il s'agit de minimiser l'indice de performance quadratique

$$J = \frac{1}{2} \int_0^{t_f} (x^T(t)Qx(t) + u^T(t)Ru(t)) \quad (3.93)$$

pour le système régi par l'équation d'état suivante :

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(0) = x_0 \quad (3.94)$$

sachant que :

$$E(t)x(t) + F(t)u(t) \leq G(t) \quad (3.95)$$

$$\eta(x(t_f)) \leq 0 \quad (3.96)$$

où $x(t) \in R^n$, $u(t) \in R^r$, A est une matrice de dimension $n \times n$, B est une matrice de dimension $n \times r$, Q est une matrice semi définie positive de dimension $n \times n$, R est une matrice définie positive de dimension $r \times r$, $E(t)$ est une matrice $s \times n$, $F(t)$ est une matrice $s \times r$ et $\eta(x(t_f)) \leq 0$ est linéaire par rapport à $x(t_f)$. Les vecteurs $x_0 \in R^n$ et $G(t)$ sont donnés.

Supposons que tous les éléments fonction du temps appartiennent à $L^2_{[t_0, t_f]}$; Ils peuvent donc être développés en séries de BPFs.

En développant (3.93)-(3.96), le problème de la commande optimale se réduira à un problème d'optimisation qui s'énonce comme suit :

Il s'agit de minimiser la fonction multidimensionnelle suivante :

$$J = \frac{1}{2} h \sum_{k=1}^m (x_k^T Q_k x_k + u_k^T R_k u_k) \quad (3.97)$$

soumise aux contraintes d'égalité et contraintes d'inégalité suivantes :

$$x_{10} - x_{10} = \frac{1}{2} h (A_1 x_1 + B_1 u_1) \quad (3.98)$$

$$x_{k0} - x_{k0} = \frac{1}{2} h (A_k x_k + B_k u_k) + h \sum_{l=0}^{k-1} (A_l x_l + B_l u_l) \quad k=2,3,\dots,m \quad (3.99)$$

$$E(k)x(k) + F(k)u(k) \leq G(k) \quad (3.100)$$

$$\eta(x(t_f)) = \eta(x_1(t_f), x_2(t_f), \dots, x_n(t_f)) \cong \eta(C_m^{x_1}, C_m^{x_2}, \dots, C_m^{x_n}) \leq 0 \quad (3.101)$$

L'instruction `fmincon` du Matlab résout facilement ce problème en « medium scale » sinon le logiciel `tomlab/snopt` [96] qui s'interface avec le Matlab doit être utilisé pour résoudre le problème en « large scale ».

Pour illustrer cette méthode, essayons de résoudre l'exemple 3.3 en le réduisant à un problème d'optimisation avec contraintes.

Exemple 3.3 [88]

Pour le système régi par les équations d'état suivantes :

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) & x_1(0) &= 0 \\ \dot{x}_2(t) &= -x_2(t) + u(t) & x_2(0) &= -1 \end{aligned}$$

Il s'agit de minimiser le critère de performance donné par

$$J = \int_0^1 (x_1^2 + x_2^2 + 0.005u^2) dt$$

Sachant que

$$x_1(t) - 8(t - 0.5)^2 + 0.5 \leq 0 \quad \text{pour } 0 \leq t \leq 1$$

En utilisant La technique de paramétrisation via les BPFs Les équations d'état se réduisent à un ensemble d'équations algébriques formulées comme suit :

$$C_1^{x_1} - x_1(0) = \frac{1}{2}hC_1^{x_2}$$

$$C_i^{x_1} - x_1(0) = \frac{1}{2}hC_i^{x_2} + h\sum_{j=1}^{i-1} C_j^{x_2}$$

$$i=2,\dots,m$$

$$C_1^{x_2} - x_2(0) = \frac{1}{2}h(-C_1^{x_2} + C_1^u)$$

$$C_i^{x_2} - x_2(0) = \frac{1}{2}h(-C_i^{x_2} + C_i^u) + h\sum_{j=1}^{i-1} (-C_j^{x_2} + C_j^u)$$

Alors que l'indice de performance se réduit à une simple équation algébrique ayant la forme suivante :

$$J_m = h\sum_{i=1}^m \left((C_i^{x_1})^2 + (C_i^{x_2})^2 + 0.005(C_i^u)^2 \right)$$

De même la contrainte d'inégalité aura la forme suivante :

$$C_i^{x_1} - 8\left[\frac{h}{2}(2k-1) - 0.5\right]^2 + 0.5 \leq 0 \quad i=1,\dots,m$$

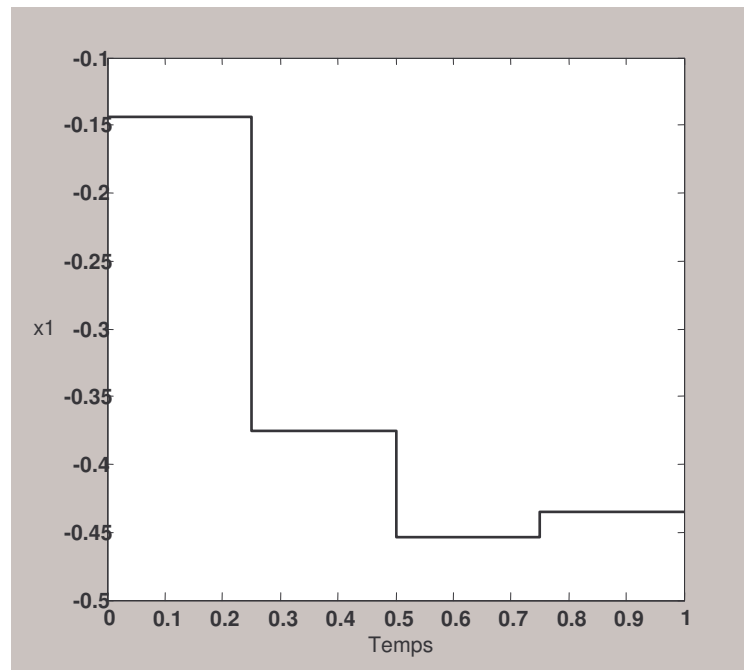
Donc le problème de commande optimale s'est rendu grâce aux BPFs à un problème d'optimisation non Linéaire à critère quadratique dont les paramètres à déterminer sont rassemblés dans le vecteur $X = [C_i^{x_1}, C_i^{x_2}, C_i^u]$. Le résultat de l'implémentation pour $m=10$ est résumé dans le tableau 3.4.

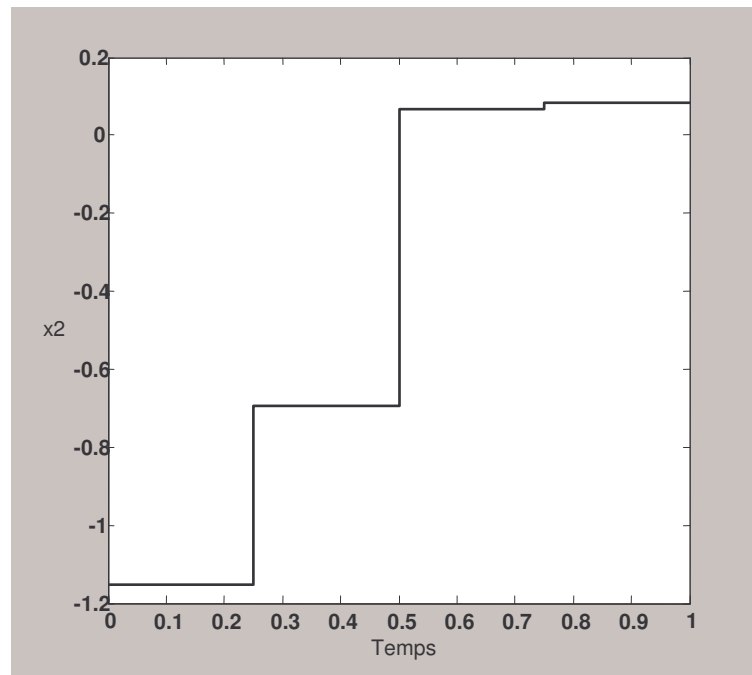
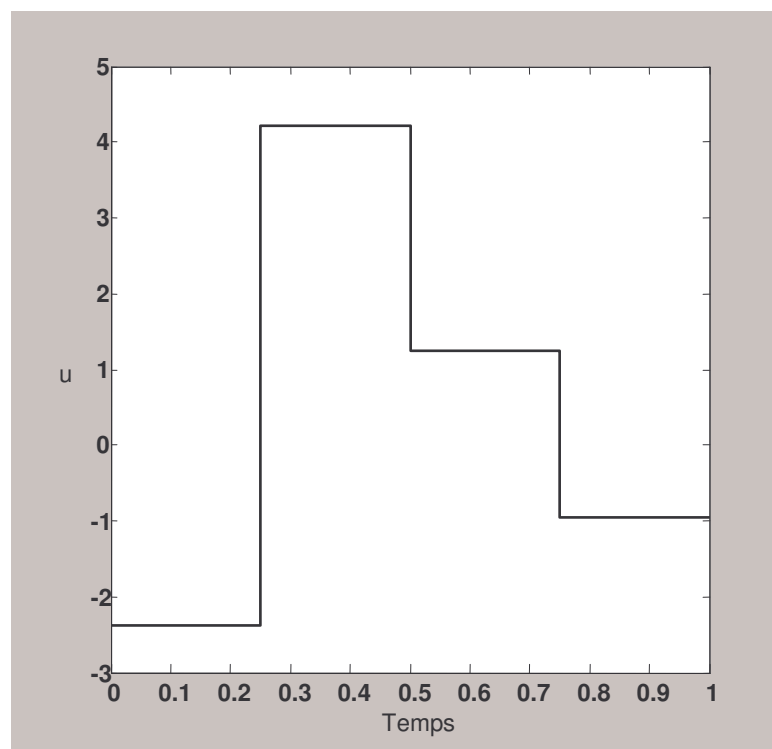
	$C_k^{x_1}$	$C_k^{x_2}$	$C_k^{x_2}$
$k=1$	-0.0545	-1.0894	-2.8766
$k=2$	-0.1692	-1.2045	-1.7210
$k=3$	-0.2901	-1.2140	-0.8875
$k=4$	-0.4019	-1.0217	2.4995
$k=5$	-0.4800	-0.5408	5.5550
$k=6$	-0.5100	-0.0593	3.4760
$k=7$	-0.5064	0.1306	0.3925
$k=8$	-0.4939	0.1193	-0.3693

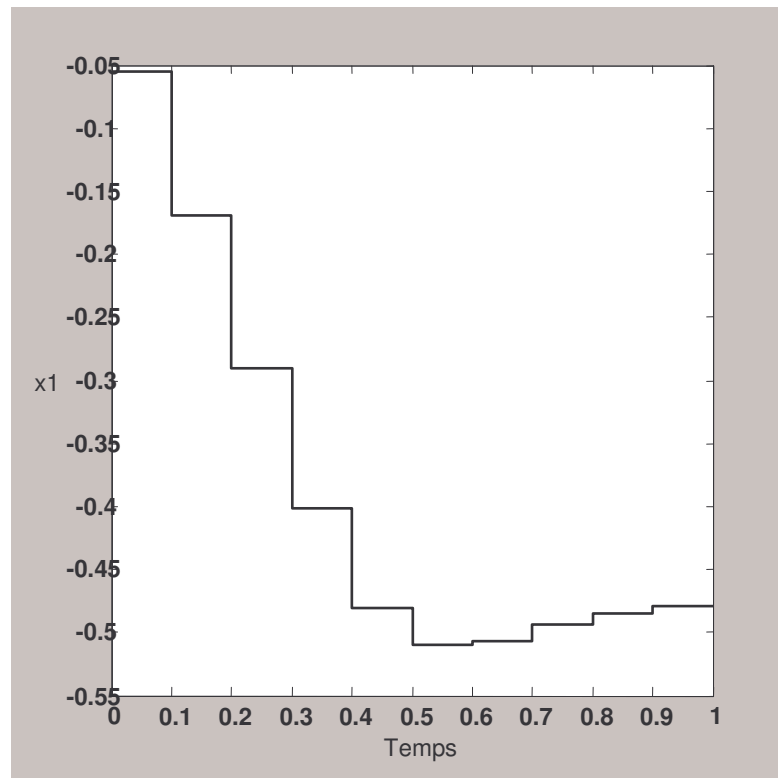
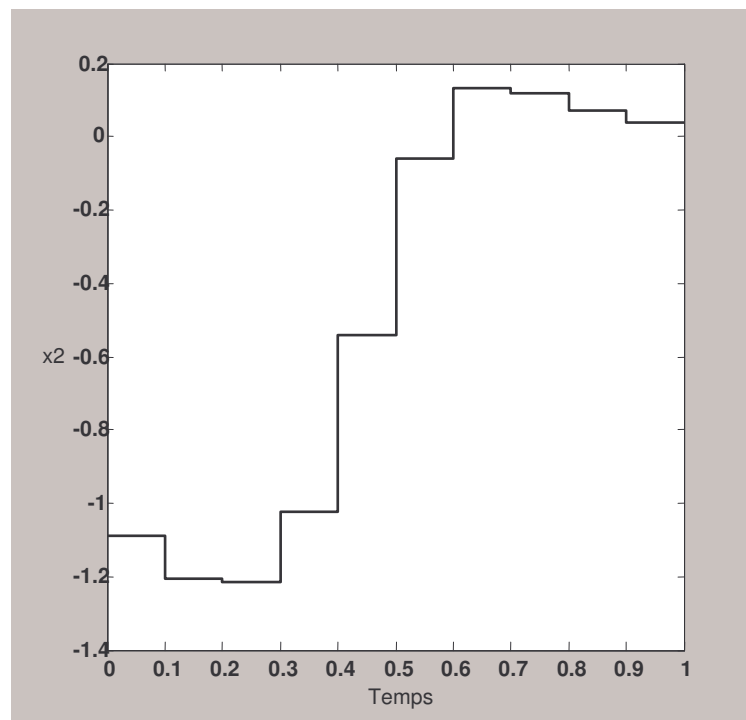
$k=9$	-0.4844	0.0718	-0.3896
$k=10$	-0.4789	0.0385	-0.1664

Tableau 3.5 Valeurs des éléments de X relatifs à $m=10$

Les trajectoires $x_1(t)$, $x_2(t)$ ainsi que la commande $u(t)$ sont représentées sur les figures 3.13 , 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.20, et 3.21 respectivement, pour $m=4$ et $m=10$ et $m=50$.

Fig. 3.13. Variation de $x_1(t)$ en fonction du temps pour $m=4$

Fig. 3.14. Variation de $x_2(t)$ en fonction du temps pour $m=4$ Fig. 3.15. Variation de $u(t)$ en fonction du temps pour $m=4$

Fig. 3.16. Variation de $x_1(t)$ en fonction du temps pour $m=10$ Fig. 3.17. Variation de $x_2(t)$ en fonction du temps pour $m=10$

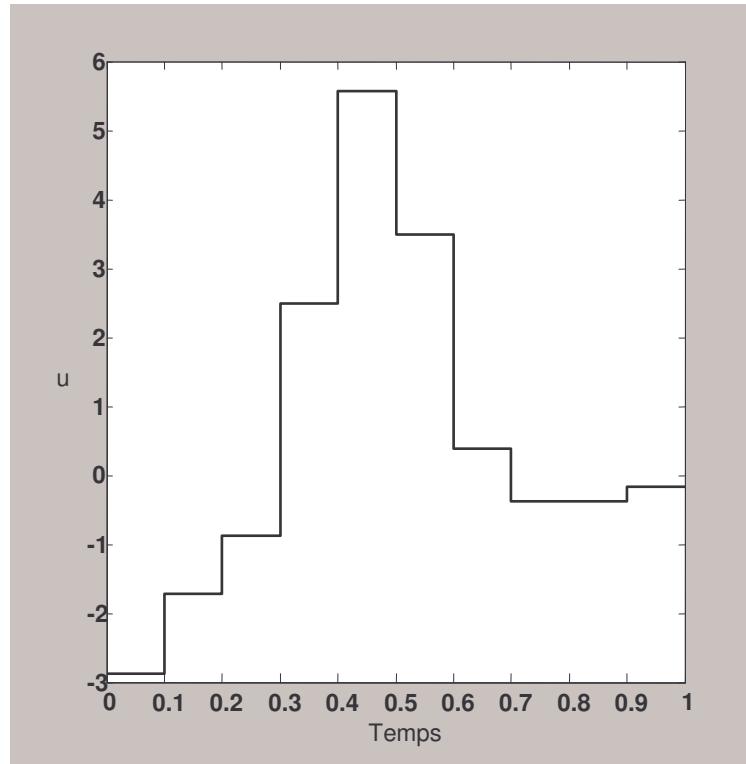


Fig. 3.18. Variation de $u(t)$ en fonction du temps pour $m=10$

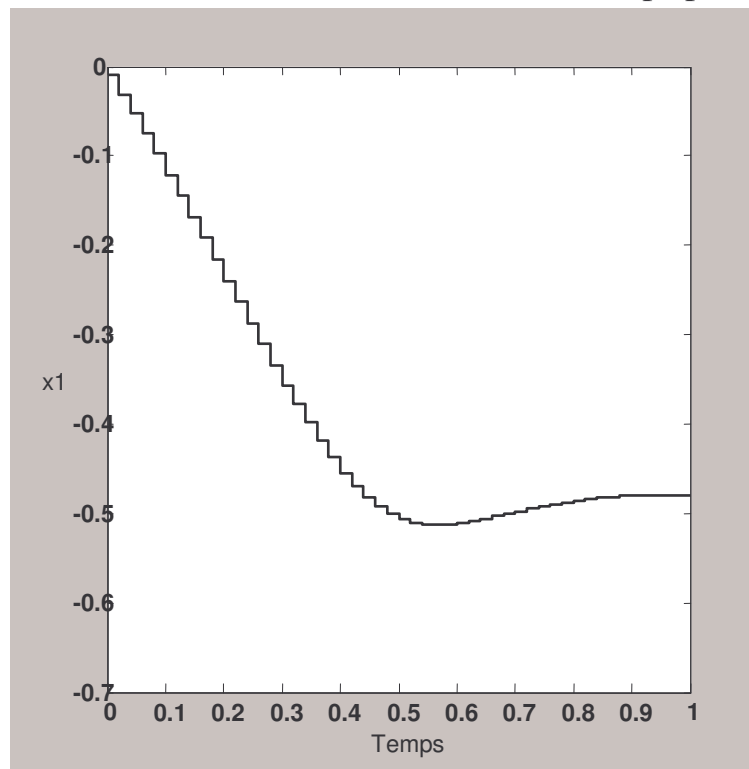


Fig. 3.19. Variation de $x_1(t)$ en fonction du temps pour $m=50$

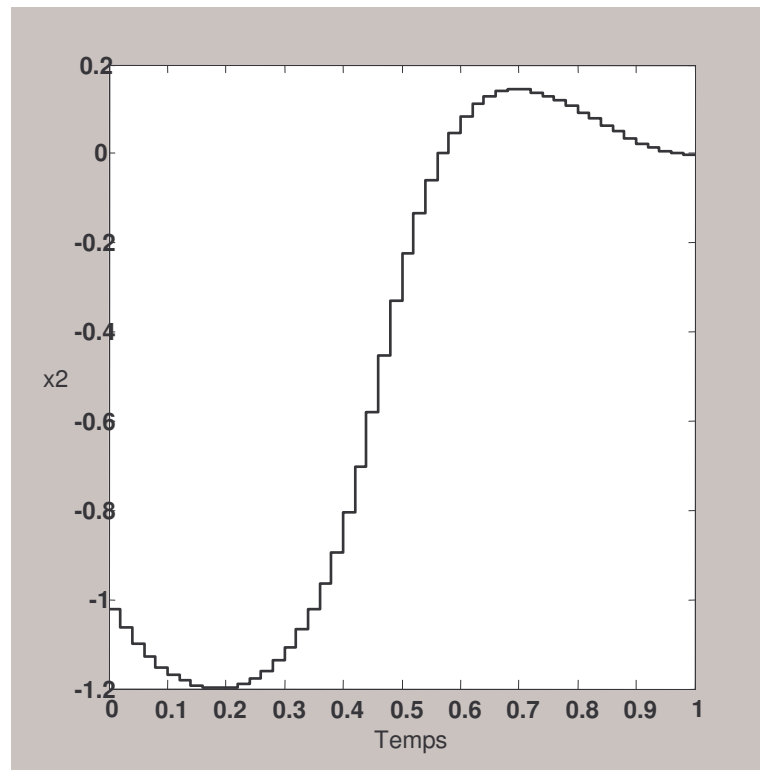


Fig. 3.20. Variation de $x_2(t)$ en fonction du temps pour $m=50$

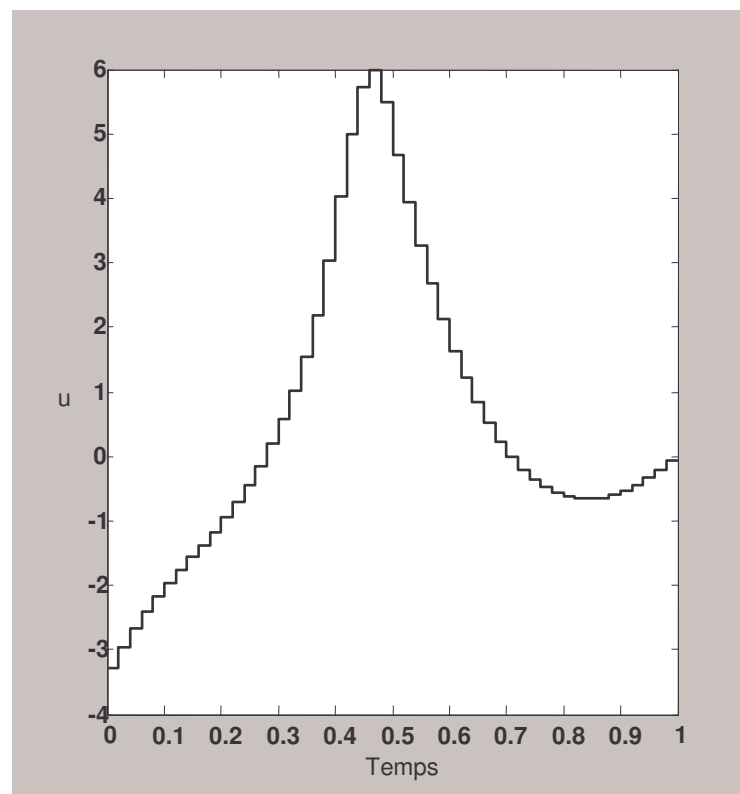


Fig. 3.21. Variation de $u(t)$ en fonction du temps pour $m=50$

Cet exemple a été étudié par H. Jaddu [88] et par Vlassenbroeck [40]. Le premier a obtenu $J=0.7409643$ en 1.67 secondes, alors que le second a obtenu un indice de performance $J=0.74096$ en 31 secondes. Notre méthode a permis d'obtenir $J=0.7429$ pour $m=50$ et $J=0.7530$ pour $m=10$ en un temps inférieur à la seconde ; donc notre méthode donne des résultats meilleurs ou comparables par rapport aux méthodes citées.

Chapitre 4

Problème De La Commande Optimale Non Linéaire

4.1 Commande Optimale Sans Contraintes

4.1.1 Introduction

Dans ce chapitre, nous étendons la méthode décrite précédemment pour résoudre les problèmes de la commande optimale non linéaire. L'une des méthodes pour résoudre le problème de la commande optimale non linéaire sans contraintes est de le convertir à un problème de programmation non linéaire en utilisant les méthodes directes. Par exemple, Sirisena [37] a proposé une méthode basée sur la paramétrisation des variables de commande utilisant les fonctions polynomiales par morceau, Frich et Stech [41] ont utilisé les fonctions de Walsh pour paramétriser les variables d'état et les variables de commande, Vlassenbroeck et Van Doreen [39] ont utilisé la paramétrisation d'état-commande via les polynômes de Chebyshev pour convertir le problème de commande optimale à un problème de programmation mathématique non linéaire. Quelques autres méthodes peuvent être trouvées dans [32, 42, 75, 76].

Dans cette thèse, le problème de la commande optimale non linéaire est converti directement à un problème d'optimisation non linéaire qui, en admettant certaines hypothèses, converge vers le problème original. Par conséquent le problème de la convergence est discuté dans ce chapitre.

4.1.2 Formulation Du Problème

Considérons le problème de la commande optimale non linéaire qui peut être formulé ainsi : Trouver une commande optimale $u^*(t)$ qui minimise l'indice de performance

$$J = \int_{t_0}^{t_f} F(x(t), u(t), t) dt \quad (4.1)$$

sachant que le système est décrit par les équations d'état suivantes :

$$\dot{x}_i(t) = f_i(x(t), u(t), t) \quad x_i(0) = x_{i0} \quad i = 1, 2, \dots, n \quad (4.2)$$

où $x(t) \in R^n$, $u(t) \in R^r$, les f_i , ainsi que F sont des fonctions scalaires continues, bornées et qui satisfont la condition de Lipschitz.

La solution de ce problème par l'application des conditions nécessaires conduit à un problème aux limites non linéaires, sinon elle conduit à une équation différentielle partielle de HJB par l'application des conditions suffisantes. Dans ce chapitre, la solution est obtenue par la méthode de paramétrisation d'état-commande.

4.1.3 Paramétrisation Du Problème

En utilisant les fonctions à blocs d'impulsion, les équations d'état (4.2) peuvent être rapprochées par un ensemble d'équations algébriques et l'indice de performance (4.1) peut être rapproché par une relation scalaire.

D'abord commençant par la propriété suivante spécifique à l'utilisation des BPFs pour l'approximation des fonctions multivariables :

$$\begin{aligned} f_i(x(t), u(t), t) &= f_i(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_r, t) \cong \\ &\cong f_i\left(\sum_{k=1}^m C_k^{x_1} \varphi_k(t), \dots, \sum_{k=1}^m C_k^{x_n} \varphi_k(t), \sum_{k=1}^m C_k^{u_1} \varphi_k(t), \dots, \sum_{k=1}^m C_k^{u_r} \varphi_k(t)\right) = \\ &= \sum_{k=1}^m f_i(C_k^{x_1}, \dots, C_k^{x_n}, C_k^{u_1}, \dots, C_k^{u_r}) \varphi_k \end{aligned} \quad (4.3)$$

L'expression (4.3) est vraie pour $i=1, 2, \dots, n$

D'où, en utilisant la propriété d'intégration du chapitre précédent, l'expression (4.2) peut être réduite à :

$$C_1^{x_i} - x_{i0} = \frac{1}{2} h f_i(C_1^{x_1}, \dots, C_1^{x_n}, C_1^{u_1}, \dots, C_1^{u_r}) \quad (4.4)$$

$$C_k^{x_i} - x_{i0} = \frac{1}{2} h f_i(C_k^{x_1}, \dots, C_k^{x_n}, C_k^{u_1}, \dots, C_k^{u_r}) + h \sum_{l=1}^{k-1} f_i(C_l^{x_1}, \dots, C_l^{x_n}, C_l^{u_1}, \dots, C_l^{u_r}) \quad (4.5)$$

$$k = 1, \dots, m$$

donc les n équations d'états (4.2) sont rapprochées par $n \times m$ équations algébriques (4.4)-(4.5).

De même l'indice de performance (4.1) peut être rapproché par une seule équation algébrique comme suit :

$$J \cong J_m = h \sum_{k=1}^m F(C_k^{x_1}, \dots, C_k^{x_n}, C_k^{u_1}, \dots, C_k^{u_r}) \quad (4.6)$$

Le problème P de commande optimale régi par (4.1)-(4.2) s'est transformé à un problème Pm d'optimisation non linéaire régi par (4.4)-(4.5)-(4.6).

4.1.2.1 Etude De La Convergence

A- Convergence des équations d'état

Résoudre (4.4)-(4.5) est équivalente à trouver les paramètres $C_k^{x_i}$ si les paramètres $C_k^{u_i}$ sont donnés. Le théorème de contraction peut être utilisé pour résoudre numériquement (4.4)-(4.5).

Soit

$$x_{i,m}(t) = \sum_{k=1}^m C_k^{x_i} \varphi_k(t) = [C_1^{x_i}, C_2^{x_i}, \dots, C_m^{x_i}] [\varphi_1, \varphi_2, \dots, \varphi_m]^T = C^{x_i} \Phi(t) \quad (4.7)$$

une solution approchée de (4.2) ; sachant que le vecteur C^{x_i} est une solution de (4.4)-(4.5).

Soit, pour une fonction $f(t) \in L_2^k[t_0, t_f]$, l'expression ω_f telle que :

$$\omega_f = \sup_{|t_1 - t_2| \leq h} |f(t_1) - f(t_2)|_{R^k} \quad (4.8)$$

Nous introduisons maintenant les différents lemmes suivants :

Lemme 1

Soit $x(t)$ un vecteur de dimension n constitué des éléments scalaires $x_i(t)$.
Si $x(t) \in L_2^n[t_0, t_f]$, alors :

$$(1) \sup_{t_0 \leq t \leq t_f} |x_m - x(t)|_{R^n} \leq \omega_x$$

$$(2) |x_m(t) - x(t)|_{L_2^n} \leq \sqrt{t_f} \omega_x$$

Où $x_m(t)$ est le vecteur approché de $x(t)$ dont les éléments $x_{i,m}(t)$ sont définis par (4.7).

Lemme 2

Si $f_i \in L_2[t_0, t_f]$ et $|f_i(t)| < \infty$, alors

$$\left| C_k^{t_0} \int_{t_0}^{t_f} f_i(\tau) d\tau - (C^{f_i} H_m)_k \right| \leq \frac{h}{2} \omega_f$$

où, H_m est la matrice opérationnelle d'intégration donnée par :

$$H_m = h \begin{bmatrix} 1/2 & 1 & 1 & \cdot & \cdot & \cdot & 1 \\ 0 & 1/2 & 1 & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{m \times m} \quad (4.9)$$

C^{f_i} est un vecteur de m éléments $C_k^{f_i}$ ($k = 1, 2, \dots, m$), et $(C^{f_i} H_m)_k$ est l'élément pondérant φ_k dans le calcul de $C^{f_i} H_m \Phi$.

Théorème 1

Supposons que $u(t)$ est donnée, et notons $U = [U_1, U_2, \dots, U_m]$ la matrice qui rapproche $u(t)$ tel que : $U_k = [C_k^{u_1}, C_k^{u_2}, \dots, C_k^{u_r}]$ $k = 1, 2, \dots, m$. Si $x(t)$ est l'unique solution de (4.2) et $X = [X_1, X_2, \dots, X_m]$ tel que $X_k = [C_k^{x_1}, C_k^{x_2}, \dots, C_k^{x_n}]$ est la solution unique de (4.4)-(4.5), alors :

$$\sigma_1 = \max_{t_0 \leq t \leq t_f} |x(t) - x_m(t)|_{R^n} \leq o(s_m) + o\left(\frac{1}{m}\right), \quad \text{où } s_m = \omega_f + L_1 \omega_u$$

Démonstration

Pour la démonstration, adoptons la notation suivante :

$$\frac{1}{h} \int_{t_0+(k-1)h}^{t_0+kh} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} dt = (Bpx)_k, \quad \frac{1}{h} \int_{t_0+(k-1)h}^{t_0+kh} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix} dt = (Bpu)_k,$$

$$\frac{1}{h} \int_{t_0+(k-1)h}^{t_0+kh} [f_i(x(t), u(t), t)] dt = (Bpf_i)_k \quad \text{Pour } k = 1, 2, \dots, m \text{ et } i = 1, 2, \dots, n$$

Soient $\varepsilon_k = |(Bpx)_k - X_k|_{R^n}$, $\xi_k = |(Bpu)_k - U_k|_{R^r}$, et

$z_k = (Bpf_i)_k - F_k$ avec $F_k = C_k^{f_i}$ où $C_k^{f_i} = f_i(X_k, U_k, t_k)$. Donc, on peut écrire

$$\begin{aligned}
 |z_k|_{R^n} &= \frac{1}{h} \left| \int_{t_0+(k-1)h}^{t_0+kh} (f_i(x(t), u(t), t) - f_i(X_k, U_k, t_k)) dt \right|_{R^n} \leq \\
 &\leq \frac{1}{h} \int_{t_0+(k-1)h}^{t_0+kh} |f_i(x(t), u(t), t) - f_i(X_k, U_k, t_k)|_{R^n} dt \leq \quad (4.10) \\
 &\leq \omega_f + \frac{1}{h} \int_{t_0+(k-1)h}^{t_0+kh} |f_i(x(t_k), u(t_k), t_k) - f_i(X_k, U_k, t_k)|_{R^n} dt \leq \omega_f + L_1 \omega_x + L_1 \xi_k + L_1 \varepsilon_k
 \end{aligned}$$

La dernière expression est justifiée par le fait que f_i est Lipschitzienne, et donc elle doit vérifiée l'inégalité suivante :

$$|f_i(x, u, t) - f_i(y, v, t)|_{R^n} \leq L_1 |x - y|_{R^n} + L_1 |u - v|_{R^n} \quad (4.11)$$

Essayons maintenant d'estimer ε_k .

$$\begin{aligned}
 \varepsilon_k &= |(Bpx)_k - X_k|_{R^n} = \left| \left(Bp \int_{t_0}^t f_i(x(\tau), u(\tau), \tau) d\tau - X_k \right) \right|_{R^n} \leq \\
 &\leq \left| \left(Bp \int_{t_0}^t f_i d\tau \right)_k - ((Bpf_i)H_m)_k \right|_{R^n} + |((Bpf_i)H_m)_k - X_k|_{R^n} \quad (4.12)
 \end{aligned}$$

Utilisant le lemme 2, tout en sachant que X_k est donné par (4.4)-(4.5), on a

$$\begin{aligned}
 \varepsilon_1 &\leq \frac{h}{2} \omega_f + \frac{h}{2} |z_1| \\
 \varepsilon_k &\leq \frac{h}{2} \omega_f + \frac{h}{2} |z_k| + h \sum_{l=1}^{k-1} |z_l|
 \end{aligned}$$

Finalement, les relations récursives suivantes sont obtenues en utilisant (4.10)

$$\varepsilon_1 \leq \left(1 - \frac{h}{2} L_1 \right)^{-1} h \rho_{m1} \quad (4.13)$$

$$\varepsilon_k \leq \alpha_m \sum_{l=1}^{k-1} \varepsilon_l + \left(1 - \frac{h}{2} L_1\right)^{-1} hm\rho_{m1} \quad (4.14)$$

$$\text{où } \alpha_m = \left(1 - \frac{h}{2} L_1\right)^{-1} hL_1 \text{ et } \rho_{m1} = \omega_f + L_1\xi_1 + o\left(\frac{1}{m}\right)$$

Par conséquent, à partir de (4.13)-(4.14), nous pouvons obtenir la relation réursive suivante :

$$\varepsilon_k \leq (1 - \alpha_m)^{k-1} \left(1 - \frac{h}{2} L_1\right)^{-1} hm\rho_m$$

$$\text{où } \rho_m = s_m + o\left(\frac{1}{m}\right) \text{ si bien que } \varepsilon_k \leq o\left(\frac{1}{m}\right)$$

En plus, on a

$$|x(t) - x_m(t)|_{R^n} \leq |(Bpx(t))\psi(t) - x_m(t)|_{R^n} + |x(t) - (Bpx(t))\psi(t)|_{R^n} \leq o\left(\frac{1}{m}\right)$$

Ainsi le théorème 1 est démontré

B- Convergence Du Problème D'approximation

Il est démontré au paragraphe précédent que la solution des équations d'état définies par (4.2) convergent vers la solution rapprochée définie par (4.4)-(4.5) lorsque m tend vers l'infini. Il reste donc à prouver la convergence pour tout le problème. Pour se faire introduisons certains lemmes.

Lemme 3

Soient $u, v \in L_2^r[t_0, t_f]$ et la condition (4.11) vérifiée. Supposons que x et y sont solutions de (4.2) correspondants, respectivement, à u et v . Il existe, donc, une constante c_1 telle que :

$$\sigma_2 = \max_{t_0 \leq t \leq t_f} |x(t) - y(t)|_{R^n} \leq c_1 |u - v|_{L_2^r}$$

Démonstration

D'après (4.2), on a

$$x(t) - x(0) = \int_{t_0}^t f(x(s), u(s), s) ds \quad \text{et} \quad y(t) - x(0) = \int_{t_0}^t f(y(s), v(s), s) ds$$

Où :

$$x(t) = [x_1(t), x_2(t), \dots, x_n(t)], \quad y(t) = [y_1(t), y_2(t), \dots, y_n(t)], \quad f = [f_1, f_2, \dots, f_n]$$

$$u(t) = [u_1(t), u_2(t), \dots, u_r(t)], \quad \text{et} \quad v(t) = [v_1(t), v_2(t), \dots, v_r(t)] \quad .$$

En utilisant la condition (4.11) et l'inégalité de Schwarz on a :

$$|x(t) - y(t)|_{R^n} \leq \int_{t_0}^t |f(x(s), u(s), s) - f(y(s), v(s), s)|_{R^n} ds$$

$$\leq 2L_1 \int_{t_0}^t |x(s) - y(s)|_{R^n} ds + L_1 \sqrt{T} |u - v|_{L_2^r}$$

En utilisant l'inégalité de Gronwalls [97] on a

$$\sigma_2 \leq c_1 |u - v|_{L_2^r} \quad \text{Ce qu'il fallait démontrer (C.Q.F.D)}$$

où $c_1 = L_1 \sqrt{T} \exp(2L_1 T)$ et $T = t_f - t_0$

Lemme 4

Les hypothèses du lemme 3 sont maintenues, en plus la fonction F est supposée Lipschitzienne, ie elle doit vérifiée la condition suivante :

$$|F(x, u, t) - F(y, v, t)| \leq L_2 |x - y|_{R^n} + L_2 |u - v|_{R^r} \quad (4.15)$$

Donc, il existe une constante c_2 telle que

$$|J(u) - J(v)| \leq c_2 |u - v|_{L_2^r}$$

Démonstration

Les conditions (4.11), (4.15) et l'inégalité de Schwarz permet décrire

$$\begin{aligned} |J(u) - J(v)| &\leq \int_{t_0}^{t_f} |F(x(s), u(s), s) - F(y(s), v(s), s)| ds \\ &\leq L_2 \int_{t_0}^{t_f} |x(s) - y(s)|_{R^n} ds + L_2 \int_{t_0}^{t_f} |u(s) - v(s)|_{R^r} ds \end{aligned}$$

En appliquant le lemme 3 et l'inégalité de Schwarz, on a

$$|J(u) - J(v)| \leq L_2 c_1 T |u - v|_{L_2^r} + L_2 \sqrt{T} |u - v|_{L_2^r} \quad \text{C.Q.F.D}$$

Lemme 5

Les conditions (4.11)-(4.15) sont maintenues et supposons que $X=[X_1, X_2, \dots, X_m]$ et $U=[U_1, U_2, \dots, U_m]$ (X_k et U_k sont définis d'après le théorème 1) satisfont les équations (4.4)-(4.5). Si $\lim_{m \rightarrow \infty} h \sum_{k=1}^m U_k^T U_k < \infty$, alors ils existent $u \in L_2^r[t_0, t_f]$ et $x \in C_n^1[t_0, t_f]$ tels que :

$$|u(t) - U \psi(t)|_{L_2^r} \rightarrow 0 \quad \text{si } m \rightarrow \infty$$

$$\max_{t_0 \leq t \leq t_f} |x(t) - X \psi(t)|_{R^n} \rightarrow 0 \quad \text{si } m \rightarrow \infty$$

Où : $\psi = [\psi_{11}, \dots, \psi_{n1}, \psi_{12}, \dots, \psi_{n2}, \dots, \psi_{1m}, \dots, \psi_{nm}]$,
 $\psi_{ij}(t) = [0, \dots, 0, \varphi_j(t), 0, \dots, 0]^T$ et $\varphi_j(t)$ est le $j^{i\text{ème}}$ élément se trouvant à la $i^{\text{ème}}$ place $i=1, \dots, n, j=1, \dots, m$ et x et u satisfont les équations (4.2)

Démonstration

Soit $U_k = [C_k^{u_1}, C_k^{u_2}, \dots, C_k^{u_r}]^T$. Par hypothèse on a : $\sum_{k=1}^{\infty} C_k^{u_i} < \infty, \quad i = 1, 2, \dots, r$

En se référant au théorème de Riesz-Fisher, on trouve qu'il existe $u_i(t) \in L_2[t_0, t_f]$ avec $\{C_k^{u_i}\}_{i=1, 2, \dots, r}$ sont les coefficients de Fourier y associés, tel que

$$|u(t) - U\psi(t)| \rightarrow 0 \quad \text{Si } m \rightarrow \infty \quad (4.16)$$

Supposons, maintenant, que $x(t)$ et $x_m(t)$ sont les solutions de (4.2) correspondantes à $u(t)$ et $U\psi(t)$ respectivement. A partir du lemme 3 et le théorème 1, on a

$$\max_{t_0 \leq t \leq t_f} |x_m(t) - X\psi(t)|_{R^n} \rightarrow 0 \quad \text{si } m \rightarrow \infty \quad (4.17)$$

Mais : $|x(t) - X\psi(t)|_{R^n} \leq c_1 |u(t) - U\psi(t)|_{L_2^r} + |x_m(t) - X\psi(t)|_{R^n}$. Donc, vu (4.16) et (4.17) le lemme est démontré.

Lemme 6

Supposons que les conditions (4.11) et (4.15) sont vérifiées et que

$$U \in \left\{ U : U \in R^{r \times m}, h \sum_{k=1}^m U_k^T U_k < \infty \right\}, \text{ donc}$$

$$(1) P_1 = |J_m(U) - J(U\psi(t))| = o(1) \quad (4.18)$$

$$(2) \text{ Si } \left| \frac{df}{dt} \right|_{R^n} \leq \infty \text{ et } \left| \frac{dL}{dt} \right|_{R^n} \leq \infty, \text{ donc } P_1 = o\left(\frac{1}{m}\right) \text{ si } m \rightarrow \infty$$

$$(3) \text{ Si } f \text{ et } L \text{ sont indépendantes de } t, \text{ alors } P_1 = o\left(\frac{1}{m}\right) \text{ si } m \rightarrow \infty$$

Démonstration

(1) Soient X une solution de (4.4)-(4.5) correspondante à U et $x_m(t)$ est la solution de (4.2) correspondante à $u(t) = U\psi(t)$. Il est clair que

$$\begin{aligned}
 |J_m(U) - J(U\psi(t))| &= \left| h \sum_{k=1}^m F(X_k, U_k, t_k) - \int_{t_0}^{t_f} F(x_m(s), U\psi(s), s) ds \right| \leq \\
 &\leq \sum_{k=1}^m \int_{(k-1)h}^{kh} |F(X_k, U_k, t_k) - F(X_k, U_k, s)| ds + \int_{t_0}^{t_f} |F(X\psi(s), U\psi(s), s) - F(x_m(s), U\psi(s), s)| ds \\
 &\leq T\omega_L + L_2 T \max_{t_0 \leq t \leq t_f} |x_m(t) - X\psi(t)|_{R^n} \tag{4.19}
 \end{aligned}$$

Par le théorème 1 et sachant que $\omega_L = o(1)$ le lemme est démontré

(2) Par hypothèses, il est aisé de voir que :

$$\begin{aligned}
 \omega_L &= o\left(\frac{1}{m}\right) \text{ si } m \rightarrow \infty \\
 \max_{t_0 \leq t \leq t_f} |x_m(t) - X\psi(t)|_{R^n} &= o\left(\frac{1}{m}\right)
 \end{aligned}$$

Par conséquent avec l'utilisation du processus mentionné dans la démonstration de (4.18), on peut achever la démonstration.

(3) Dans ce cas, on peut obtenir facilement

$$|J_m(U) - J(U\psi(t))| \leq L_2 T \max_{t_0 \leq t \leq t_f} |x_m(t) - X\psi(t)|_{R^n} = o\left(\frac{1}{m}\right) \text{ C.Q.F.D}$$

Théorème 2

Supposons que

- (a) Les conditions (4.11) et (4.15) sont vérifiées.
- (b) Il existe une solution unique $u^*(t)$ et $x^*(t)$ du problème P défini par (4.1)-(4.2).
- (c) Il existe une solution unique U^* et X^* du problème P_m défini par (4.4)-(4.5)-(4.6). Alors

$$(1) J_m(U^*) \rightarrow J(u^*(t)) \text{ si } m \rightarrow \infty$$

(2) Ils existent des séquences

$$u_{m_k}(t) = [U_1 \quad U_2 \quad \dots \quad U_{m_k}] [\Phi_{m_k} \quad \Phi_{m_k} \quad \dots \quad \Phi_{m_k}]^T$$

et $x_{m_k}(t) = [X_1 \quad X_2 \quad \dots \quad X_{m_k}] [\Phi_{m_k} \quad \Phi_{m_k} \quad \dots \quad \Phi_{m_k}]^T$ tels que :

$$g_1 = \max_{t_0 \leq t \leq t_f} |x^*(t) - x_{m_k}|_{R^n} \rightarrow 0 \quad \text{si } m_k \rightarrow \infty$$

$$g_2 = |u^*(t) - u_{m_k}|_{L_2^r} \rightarrow 0 \quad \text{si } m_k \rightarrow \infty$$

Démonstration

(1) D'abord il faut noter que

$$\begin{aligned} J_m(U^*) - J(u^*(t)) &\leq \\ &\leq J_m(Bpu^*(t)) - J((Bpu^*(t))\psi(t)) + J((Bpu^*(t))\psi(t)) - J(u^*(t)) = a_m \end{aligned} \quad (4.20)$$

Donc, tenant compte du lemme 3 et du lemme 6 on constate que

$$a_m = o(1) \text{ si } m \rightarrow \infty \quad (4.21)$$

D'autre part, tenant compte de l'optimalité de $u^*(t)$, on a

$$J_m(U^*) - J(u^*(t)) \geq J_m(U^*) - J(U^*\psi(t)) = \beta \quad (4.22)$$

donc, d'après le lemme 6 on a aussi $\beta = o(1) \text{ si } m \rightarrow \infty \quad (4.23)$

En combinant (4.20)-(4.23) on achève la démonstration

(2) Par hypothèse, on a : $g_m = \int_{t_0}^{t_f} u_m^T(t)u_m(t)dt = h \sum_{k=1}^m (U_k^*)^T U_k^* < \infty$

En se servant du théorème de Bolzano-Weirstrass, on sait qu'il existe une sous suite de convergence g_{m_k} . En utilisant le lemme 5, il doit exister $\bar{u}(t)$ et $\bar{x}(t)$ tels que :

$$b_1 = \max_{t_0 \leq t \leq t_f} \left| \bar{x}(t) - x_{m_k}(t) \right|_{R^n} \rightarrow 0 \text{ si } m_k \rightarrow \infty \quad (4.24)$$

$$b_2 = \left| \bar{u}(t) - u_{m_k}(t) \right|_{L_2^r} \rightarrow 0 \text{ si } m_k \rightarrow \infty \quad (4.25)$$

D'autre part, vu la condition (4.15), on obtient

$$\begin{aligned} \left| J_{m_k}(U^*) - J(\bar{u}(t)) \right| &\leq \eta = \left| \frac{T}{m_k} \sum_{k=1}^{m_k} F(X_k, U_k, t_k) - \int_{t_0}^{t_f} F(\bar{x}(s), \bar{u}(s), s) ds \right| \\ &\leq \sum_{k=1}^{m_k} \int_{t_0+(k-1)h}^{t_0+kh} |F(X_k, U_k, t_k) - F(X_k, U_k, s)| ds + L_2 T b_1 + L_2 \sqrt{T} b_2 \leq T \omega_L + L_2 T b_1 + L_2 \sqrt{T} b_2 \end{aligned}$$

Sachant que la fonction F est continue sur $[t_0, t_f]$ et à partir de (4.24)-(4.25) on obtient $\eta \rightarrow 0 \Rightarrow J(\bar{u}(t)) \rightarrow J(u^*(t))$. Par conséquent $\bar{u}(t) = u^*(t)$ et $\bar{x}(t) = x^*(t)$ (vu l'unicité de la solution du problème P).

Théorème 3

Supposons que les hypothèses du théorème 2 sont vérifiées. Si

(1) la dérivée de $u(t)$ est bornée.

(2) $\left| \frac{df_i}{dt} \right|_{R^n} < \infty$ et $\left| \frac{dF}{dt} \right|_{R^n} < \infty$ les f_i et F sont indépendantes de t , c'est à dire

$f_i = f_i(x(t), u(t)) \quad i = 1, 1, \dots, n$ et $F = F(x(t), u(t))$. Alors :

$$(1) \quad \left| J_m(U^*) - J(u^*(t)) \right| = o\left(\frac{1}{m}\right) \quad \text{si } m \rightarrow \infty$$

$$(2) \quad g_1 = o\left(\frac{1}{m}\right) \quad \text{et} \quad g_2 = o\left(\frac{1}{m}\right) \quad \text{si } m \rightarrow \infty$$

La démonstration de ce théorème est triviale . Les exemples qui suivent illustrent la méthode développée ci-dessus.

Exemple1 : Problème de Rayleigh.

Il s'agit de trouver la commande $u^*(t)$ qui minimise la fonctionnelle :

$$J = \int_0^{2.5} (x_1^2 + u^2) dt \quad (4.26)$$

Sachant que :

$$\dot{x}_1 = x_2 \quad x_1(0) = -5 \quad (4.27)$$

$$\dot{x}_2 = -x_1 + 1.4x_2 - 0.14x_2^3 + 4u \quad x_2(0) = -5 \quad (4.28)$$

L'approximation de ce problème via les BPFs conduit au problème d'optimisation non linéaire suivant : Il s'agit de trouver la commande

$u_m(t) = \sum_{i=1}^m C_i^u \phi_i(t)$ qui minimise la fonction coût suivante :

$$J_m = h \sum_{i=1}^m \left[(C_i^{x_1})^2 + (C_i^u)^2 \right] \quad (4.29)$$

Sachant que :

$$C_1^{x_1} + 5 = \frac{h}{2} C_1^{x_2}$$

$$C_i^{x_1} + 5 = \frac{h}{2} C_i^{x_2} + h \sum_{j=1}^{i-1} C_j^{x_2}$$

$$C_1^{x_2} + 5 = \frac{h}{2} \left(-C_1^{x_1} + 1.4C_1^{x_2} - 0.14(C_1^{x_2})^3 + 4C_1^u \right) \quad (4.30)$$

$$C_i^{x_2} + 5 = \frac{h}{2} \left(-C_i^{x_1} + 1.4C_i^{x_2} - 0.14(C_i^{x_2})^3 + 4C_i^u \right) + h \sum_{j=1}^{i-1} \left(-C_j^{x_1} + 1.4C_j^{x_2} - 0.14(C_j^{x_2})^3 + 4C_j^u \right)$$

Donc le problème P donné par (4.26), (4.27) et (4.28) s'est ramené à un problème d'optimisation P_m donné par (4.29) et (4.30). La commande optimale ainsi que les trajectoires d'état correspondantes sont données par les figures de 4.1 à 4.6 pour $m=50$ et $m=70$

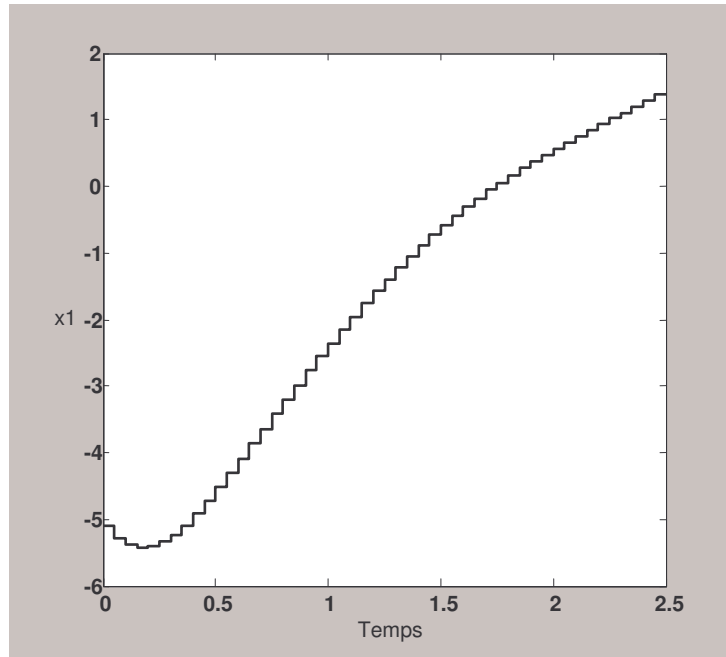


Fig. 4.1. Variation de x_1 en fonction du temps pour $m=50$

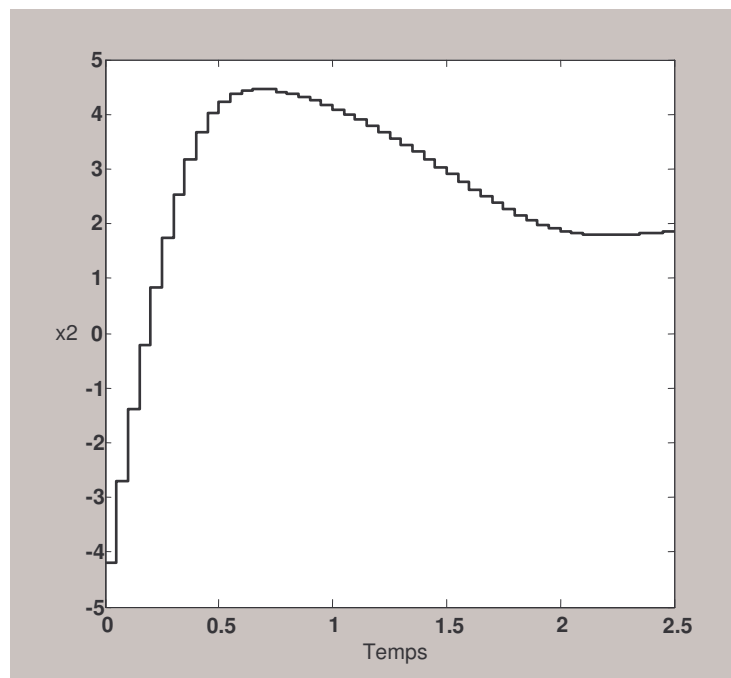
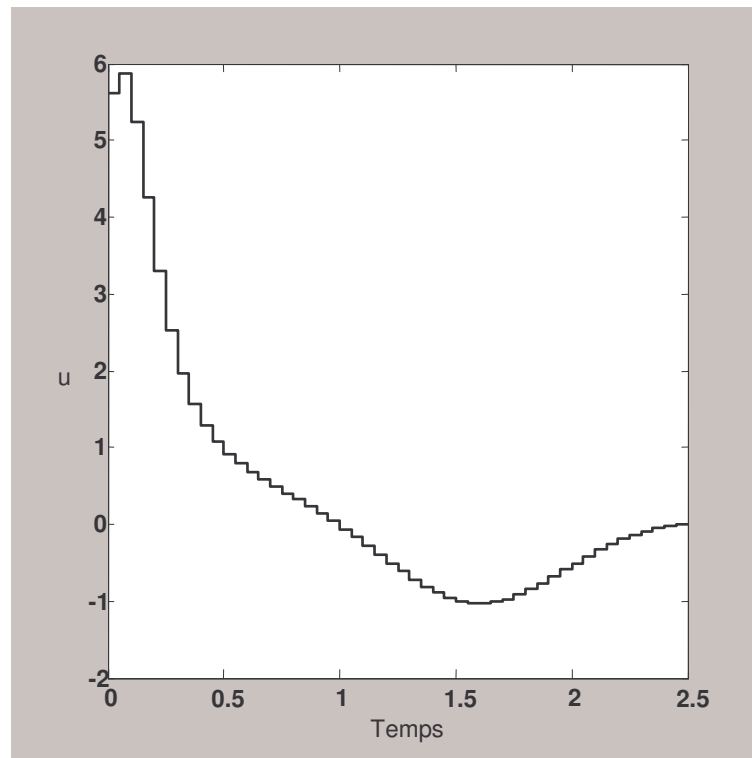
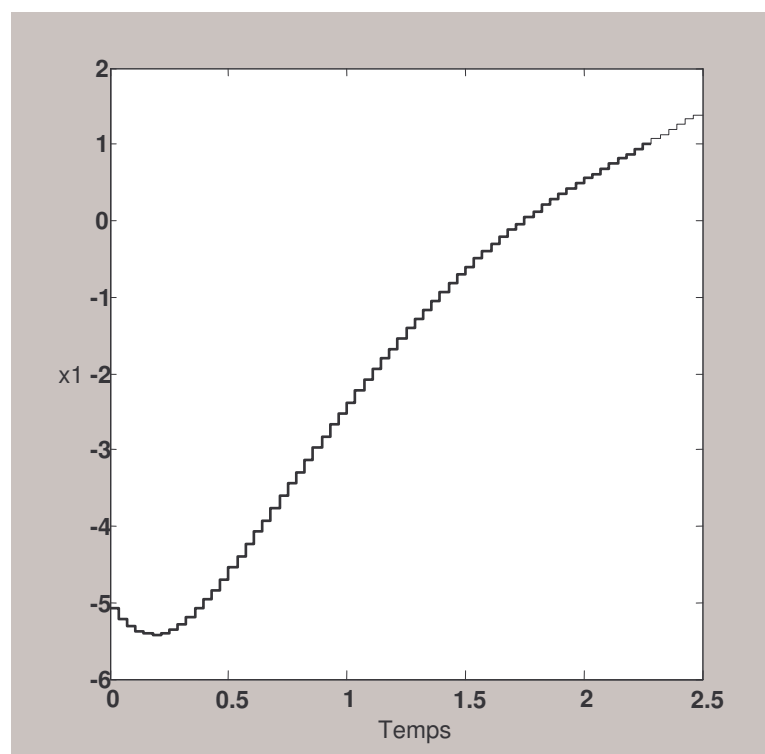


Fig. 4.2. Variation de x_2 en fonction du temps pour $m=50$

Fig. 4.3. Variation de u en fonction du temps pour $m=50$ Fig. 4.4. Variation de x_1 en fonction du temps pour $m=70$

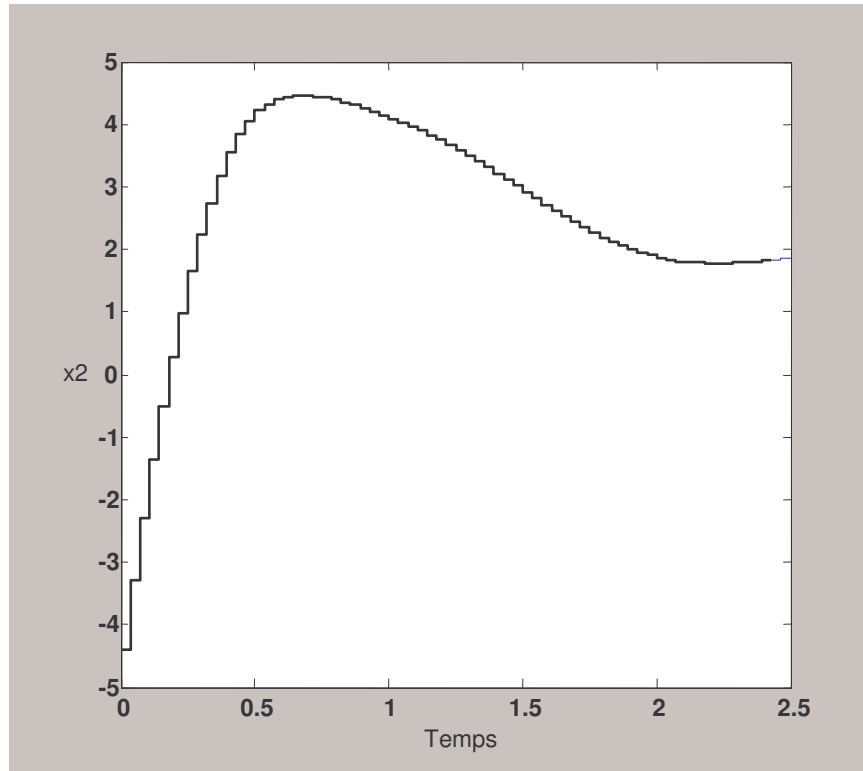


Fig. 4.5. Variation de x_2 en fonction du temps pour $m=70$

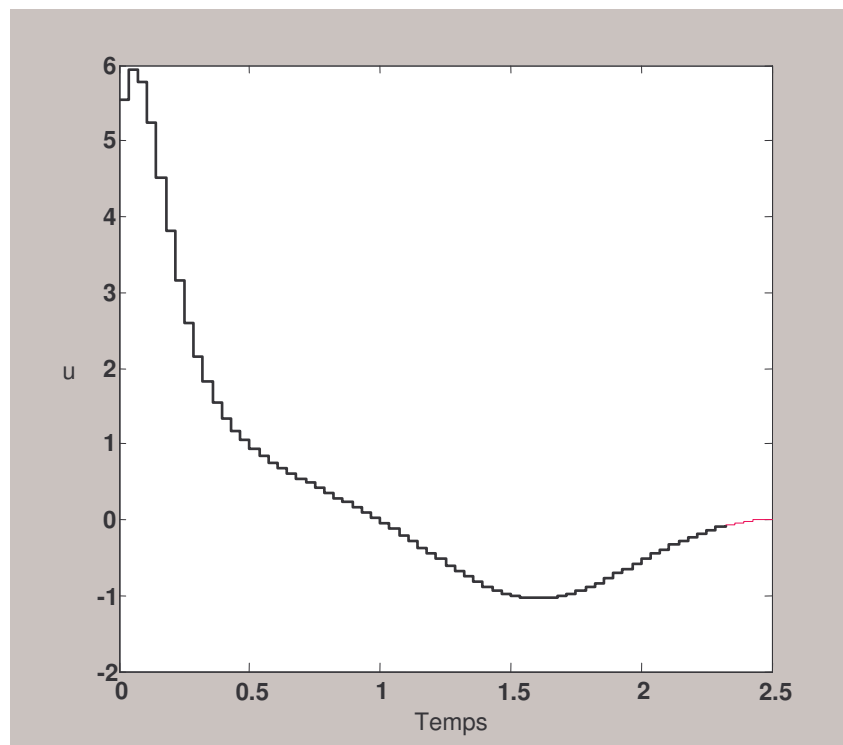


Fig. 4.6. Variation de u en fonction du temps pour $m=70$

Le tableau 4.1 montre une comparaison entre la valeur optimale de notre méthode avec les résultats obtenus par d'autres chercheurs.

Source	J
Nedeljkovic [29]	29.419
Sirisena [37]	29.451
H. Jaddu [88]	29.4022
Notre méthode pour $m=70$	29.4079

Tableau 4.1. Valeurs optimales de J pour le problème de Rayleigh

Exemple 2 : Problème de l'oscillateur de Van der Pol.

Les équations d'états du système sont :

$$\dot{x}_1 = x_2 \quad x_1(0) = 1 \quad (4.31)$$

$$\dot{x}_2 = -x_1 + (1 - x_1^2)x_2 + u \quad x_2(0) = 0 \quad (4.32)$$

La fonctionnelle coût à minimiser est donnée par :

$$J = \frac{1}{2} \int_0^5 (x_1^2 + x_2^2 + u^2) dt \quad (4.33)$$

Ce problème peut être réduit à un problème d'optimisation non linéaire défini comme suit : Il s'agit de déterminer la commande u_m ainsi que les trajectoires x_{1m} et x_{2m} minimisant la fonction :

$$J_m = \frac{h}{2} \sum_{i=1}^m \left[(C_i^{x_1})^2 + (C_i^{x_2})^2 + (C_i^u)^2 \right] \quad (4.34)$$

Sachant que :

$$C_1^{x_1} - 1 = \frac{h}{2} C_1^{x_2}$$

$$C_i^{x_1} - 1 = \frac{h}{2} C_i^{x_2} + h \sum_{j=1}^{i-1} C_j^{x_2} \quad (4.35)$$

$$C_1^{x_2} = \frac{h}{2} \left(-C_1^{x_1} + C_1^{x_2} - C_1^{x_2} (C_1^{x_1})^2 + C_1^u \right)$$

$$C_i^{x_2} = \frac{h}{2} \left(-C_i^{x_1} + C_i^{x_2} - C_i^{x_2} (C_i^{x_1})^2 + C_i^u \right) + h \sum_{j=1}^{i-1} \left(-C_j^{x_1} + C_j^{x_2} - C_j^{x_2} (C_j^{x_1})^2 + C_j^u \right)$$

L'implémentation de cet algorithme permet l'obtention des figures de (4.7) à ((4.12) relatives à la commande optimale approximée ainsi qu'aux trajectoires correspondantes pour $m=30$ et $m=50$.

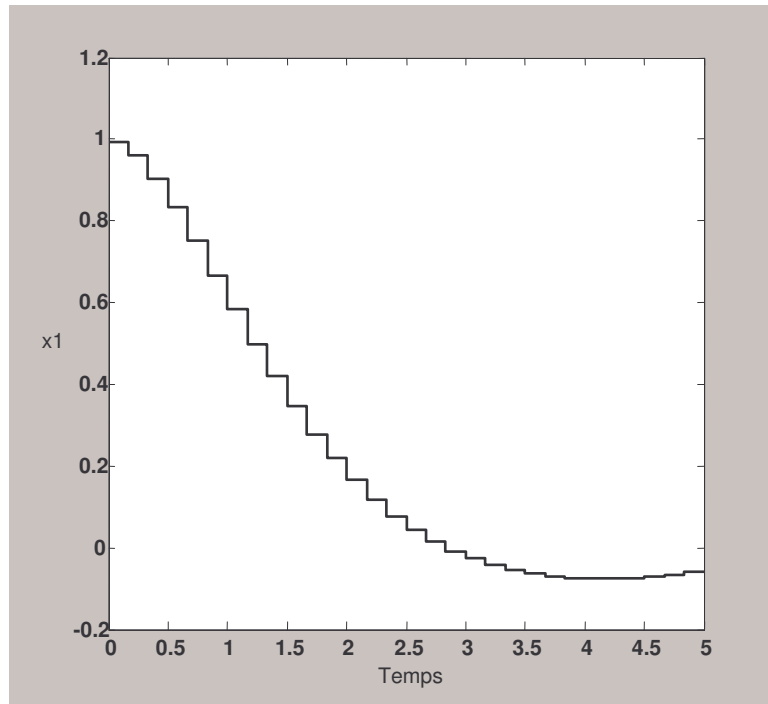


Fig. 4.7. Variation de x_1 en fonction du temps pour $m=30$

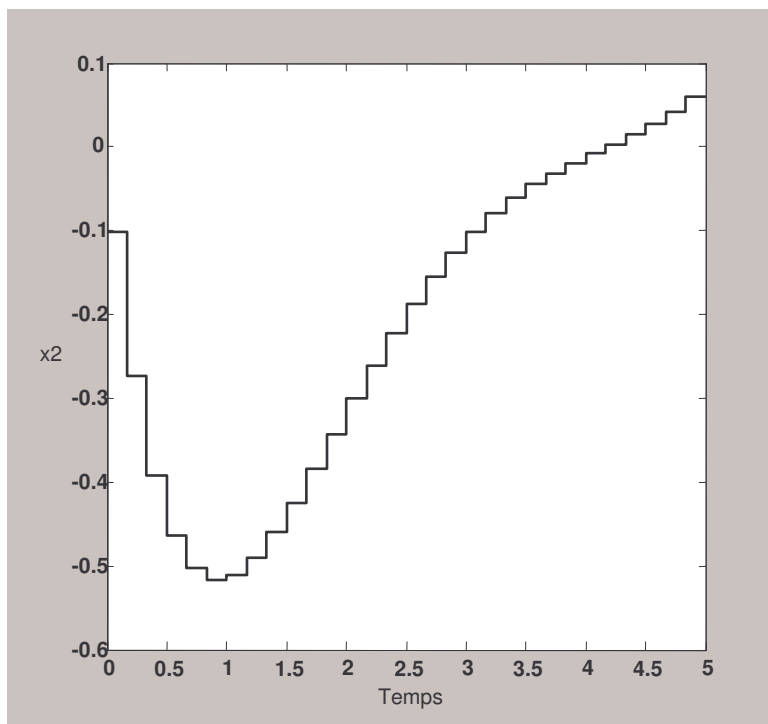
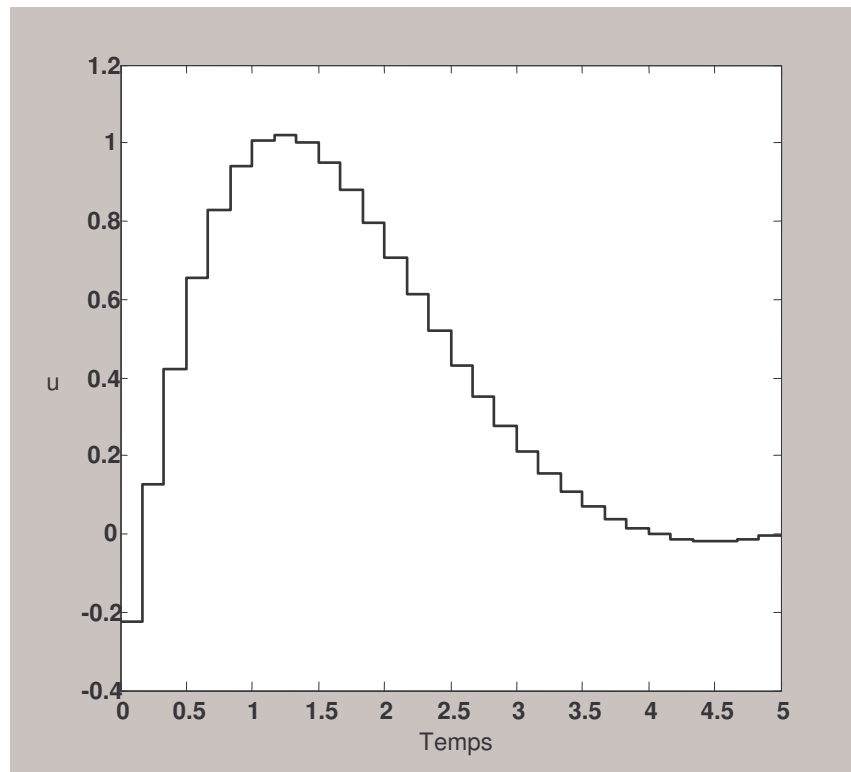
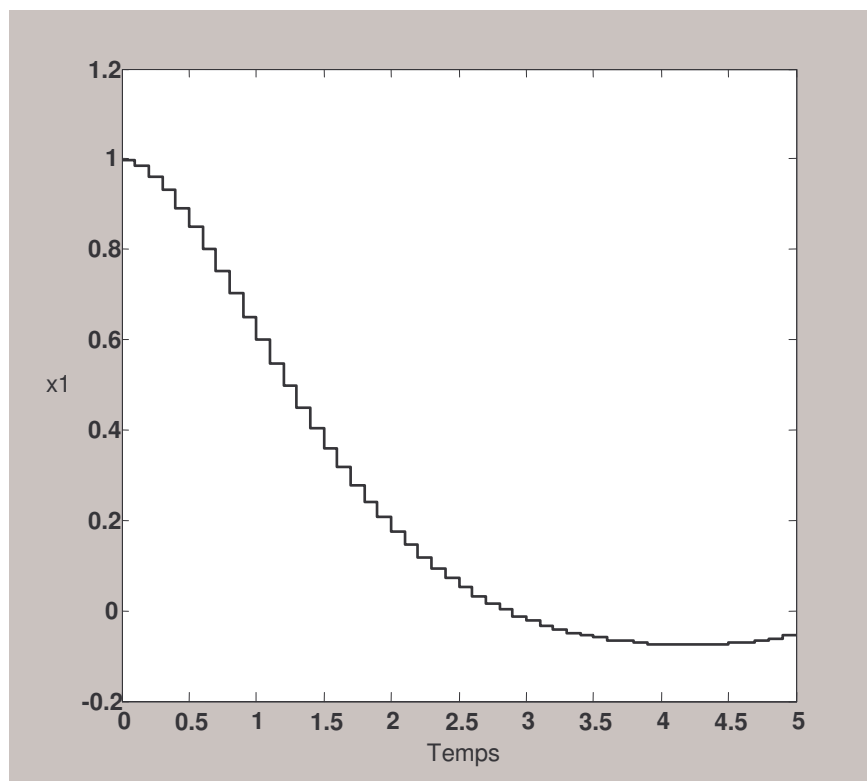
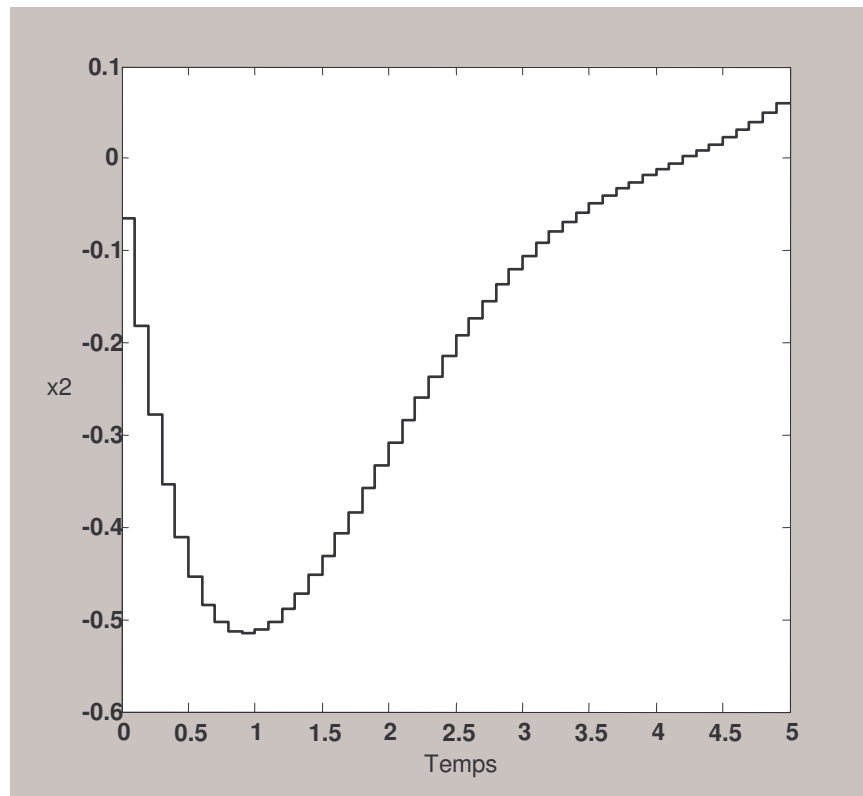
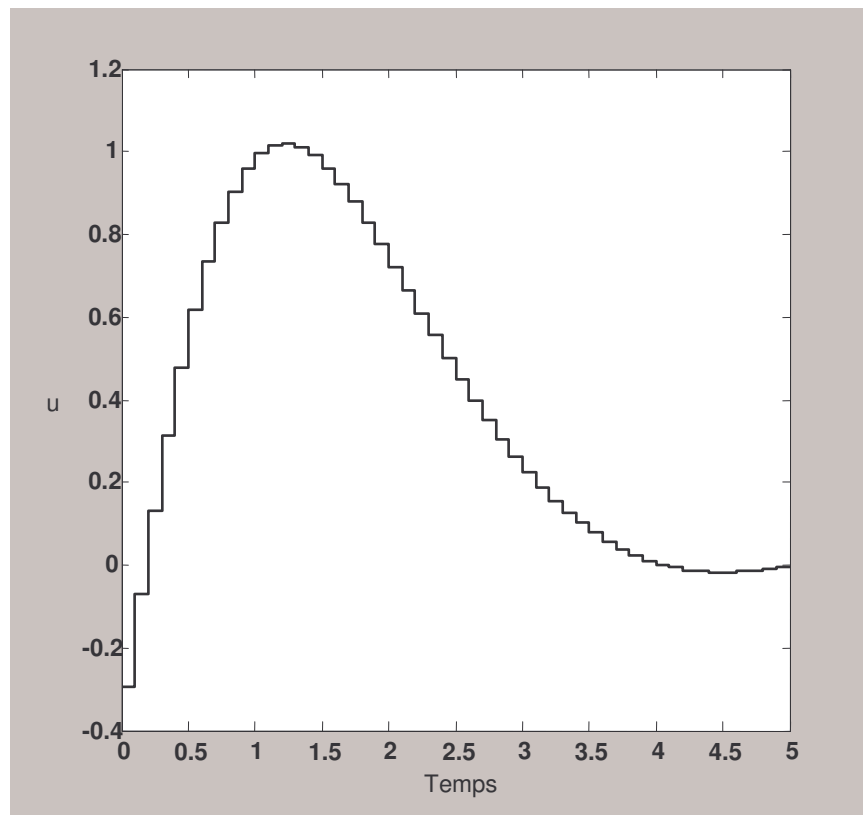


Fig. 4.8. Variation de x_2 en fonction du temps pour $m=30$

Fig. 4.9. Variation de u en fonction du temps pour $m=30$ Fig. 4.10. Variation de x_1 en fonction du temps pour $m=50$

Fig. 4.11. Variation de x_2 en fonction du temps pour $m=50$ Fig. 4.12. Variation de u en fonction du temps pour $m=50$

Le tableau 4.2 montre une comparaison entre la valeur optimale de notre méthode avec les résultats obtenus par d'autres chercheurs.

Source	J
Bullock et Franklin [59]	1.433508
Bashein et Enns [55]	1.438097
H. Jaddu [88]	1.4334872
Notre méthode pour $m=30$	1.4348
Notre méthode pour $m=50$	1.4339

Tableau 4.2. Valeurs optimales de J pour le problème de Van der Pol

On remarque bien que nos valeurs sont comparables à celles obtenues par les autres chercheurs même si on opte pour des valeurs faibles de m .

4.2 Commande Optimale Avec Contraintes

4.2.1 Introduction

Dans ce chapitre, nous traitons le problème de la commande optimale non linéaire avec contraintes. La méthode que nous proposons est basée sur l'approximation via les fonctions à blocs d'impulsions (BPFs).

Les méthodes directes ont été appliquées sur différentes classes de problèmes de commande optimale avec contraintes. Cullum [72], Kraft [75], Stryk et Bulirsch [73] et Betts [78] ont utilisé la méthode de discrétisation pour ramené le problème à un problème de programmation non linéaire. D'autre part, en utilisant la technique de paramétrisation de la commande, Goh et Teo [36], Troch et al. [33] ont pu résoudre le problème général de la commande optimale non linéaire avec contraintes, alors que Sirisina et tan [50] ont pu résoudre le problème avec des contraintes de saturation sur la commandes et des contraintes sur les variables d'état à l'instant final. L'utilisation de la paramétrisation d'état a été utilisé par [79] pour résoudre le problème quadratique linéaire avec des contraintes d'inégalité commande-état, ainsi que par H.Jaddu [88] pour résoudre le problème de commande optimale non linéaire avec contraintes de saturation sur la commandes et des contraintes sur l'état terminal. Enfin la paramétrisation commande-état a été appliquée par Vlassenbroeck [40] en utilisant les polynômes de Chebyshev pour résoudre le problème de commande optimale non linéaire avec contraintes, elle a été appliquée par Frick et Stech [42] via l'utilisation des fonctions de Walsh pour résoudre le problème de commande optimale non linéaire avec des contraintes de saturation sur la commande.

Ce chapitre sera consacré à l'étude de la commande optimale des systèmes non linéaires soumis à des contraintes de saturation sur la commande et des contraintes sur les variables d'état à l'instant terminal via l'utilisation des BPFs. Les résultats de simulation seront présentés à la fin du chapitre.

4.2.2 Formulation Du Problème

Considérons le problème de commande optimale suivant : Trouver la commande optimale $u(t)$ qui minimise l'indice de performance suivant :

$$J = \int_{t_0}^{t_f} F(x, u, t) dt \quad (4.36)$$

Sachant :

1- Les équations d'état et les conditions initiales

$$\dot{x}_k = f_k(x, u, t) \quad x_k(t_0) = x_{k0} \quad k = 1, \dots, n \quad (4.37)$$

Avec : $x \in R^n$, $u \in R^r$ et $r \leq n$

2- Les contraintes sur l'état à l'instant terminal

$$\eta(x(t_f), t_f) = 0 \quad (4.38)$$

3- Les contraintes de saturation sur la commande

$$U_{\min} \leq u(t) \leq U_{\max} \quad (4.39)$$

4.2.3 Paramétrisation du problème

Le problème de commande optimale avec contraintes (4.37)-(4.39) peut être résout en le ramenant à un problème de programmation non linéaire en utilisant l'approximation via les BPFs. Dans ce cas le problème résultant s'énonce de la manière suivante : Il s'agit de trouver la commande u_m , approximation de $u(t)$ qui minimise la fonction :

$$J_m = h \sum_{i=1}^m F(C_i^{x_1}, C_i^{x_2}, \dots, C_i^{x_n}, C_i^{u_1}, C_i^{u_2}, \dots, C_i^{u_r}) \quad (4.40)$$

Sachant :

1- L'approximation des équations d'état du système

$$C_1^{x_k} - x_{k0} = \frac{h}{2} f_k(C_1^{x_1}, C_1^{x_2}, \dots, C_1^{x_n}, C_1^{u_1}, C_1^{u_2}, \dots, C_1^{u_r}) \quad k=1, \dots, n \quad i=1, \dots, m \quad (4.41)$$

$$C_i^{x_k} - x_{k0} = \frac{h}{2} f_k(C_i^{x_1}, C_i^{x_2}, \dots, C_i^{x_n}, C_i^{u_1}, C_i^{u_2}, \dots, C_i^{u_r}) + h \sum_{j=1}^{i-1} f_k(C_j^{x_1}, \dots, C_j^{x_n}, C_j^{u_1}, \dots, C_j^{u_r})$$

2- Les contraintes d'état terminales

$$\eta(C_m^{x_1}, C_m^{x_2}, \dots, C_m^{x_n}) = 0 \quad (4.42)$$

3- Les contraintes de saturation

Soient : $U_{\max} = [u_{1\max}, u_{2\max}, \dots, u_{r\max}]$ et $U_{\min} = [u_{1\min}, u_{2\min}, \dots, u_{r\min}]$ par conséquent (4.39) peut être approximée comme suit :

$$u_{j\min} \leq C_i^{u_j} \leq u_{j\max} \quad j=1, \dots, r \quad i=1, \dots, m \quad (4.43)$$

Le problème P décrit par (4.36)-(4.39) converge vers le problème Pm décrit par (4.40)-(4.43) lorsque m tend vers l'infini. La démonstration est omise mais les exemples qui suivent étayent cette affirmation.

Exemple 3 [88] :

Considérons le problème de l'oscillateur de Van der Pol. Les équations d'état du système sont :

$$\dot{x}_1 = x_2 \quad x_1(0) = 1 \quad (4.44)$$

$$\dot{x}_2 = -x_1 + (1 - x_1^2)x_2 + u \quad x_2(0) = 0 \quad (4.45)$$

La fonction coût à minimiser est donnée par :

$$J = \frac{1}{2} \int_0^5 (x_1^2 + x_2^2 + u^2) dt \quad (4.46)$$

La contrainte d'état terminale est :

$$\eta(x(t_f)) = 1 - x_2(t_f) + x_1(t_f) = 0 \quad (4.47)$$

Après Paramétrisation, ce problème se réduit à un problème de programmation non linéaire décrit par :

- L'approximation des équations d'état

$$\begin{aligned} C_1^{x_1} - 1 &= \frac{h}{2} C_1^{x_2} \\ C_i^{x_1} - 1 &= \frac{h}{2} C_i^{x_2} + h \sum_{j=1}^{i-1} C_j^{x_2} \\ C_1^{x_2} &= \frac{h}{2} \left(-C_1^{x_1} + C_1^{x_2} - C_1^{x_2} (C_1^{x_1})^2 + C_1^u \right) \\ C_i^{x_2} &= \frac{h}{2} \left(-C_i^{x_1} + C_i^{x_2} - C_i^{x_2} (C_i^{x_1})^2 + C_i^u \right) + h \sum_{j=1}^{i-1} \left(-C_j^{x_1} + C_j^{x_2} - C_j^{x_2} (C_j^{x_1})^2 + C_j^u \right) \end{aligned} \quad (4.48)$$

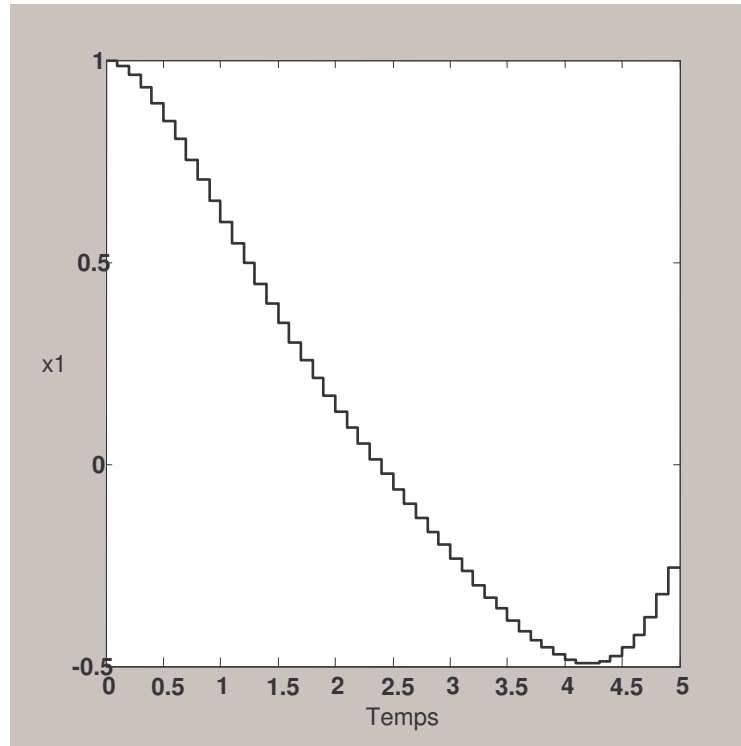
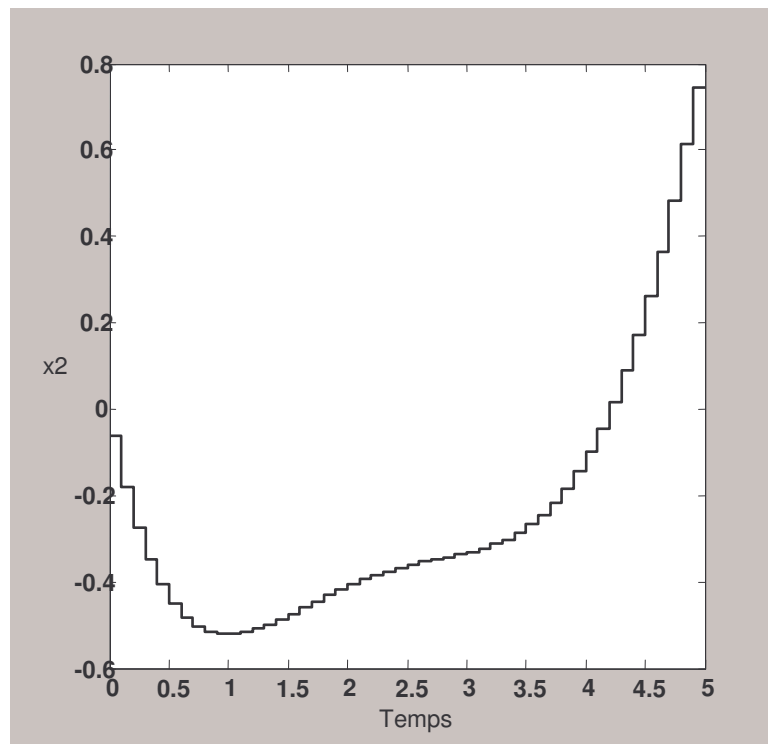
- La fonction coût à minimiser

$$J_m = \frac{h}{2} \sum_{i=1}^m \left[(C_i^{x_1})^2 + (C_i^{x_2})^2 + (C_i^u)^2 \right] \quad (4.49)$$

- La contrainte d'état terminale

$$1 - C_m^{x_2} + C_m^{x_1} = 0 \quad (4.50)$$

L'exécution de l'algorithme (4.48)-(4.50) permet l'obtention des figures de (4.13) à (4.18) correspondantes à $m=50$ et $m=100$.

Fig. 4.13. Variation de x_1 en fonction du temps pour $m=50$ Fig. 4.14. Variation de x_2 en fonction du temps pour $m=50$

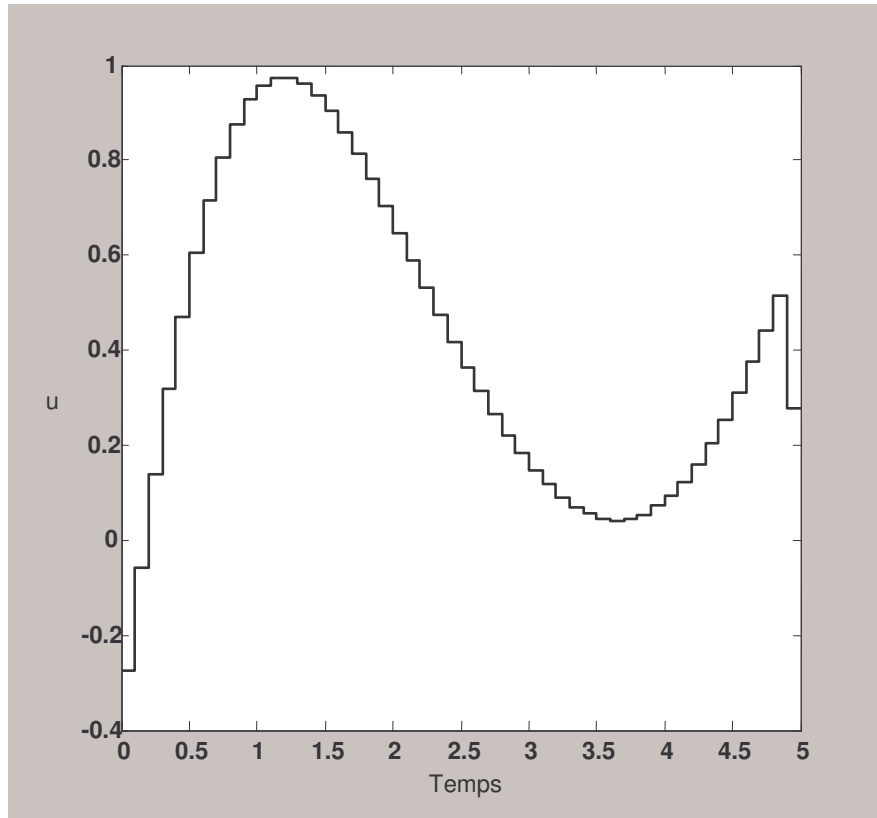


Fig. 4.15. Variation de u en fonction du temps pour $m=50$

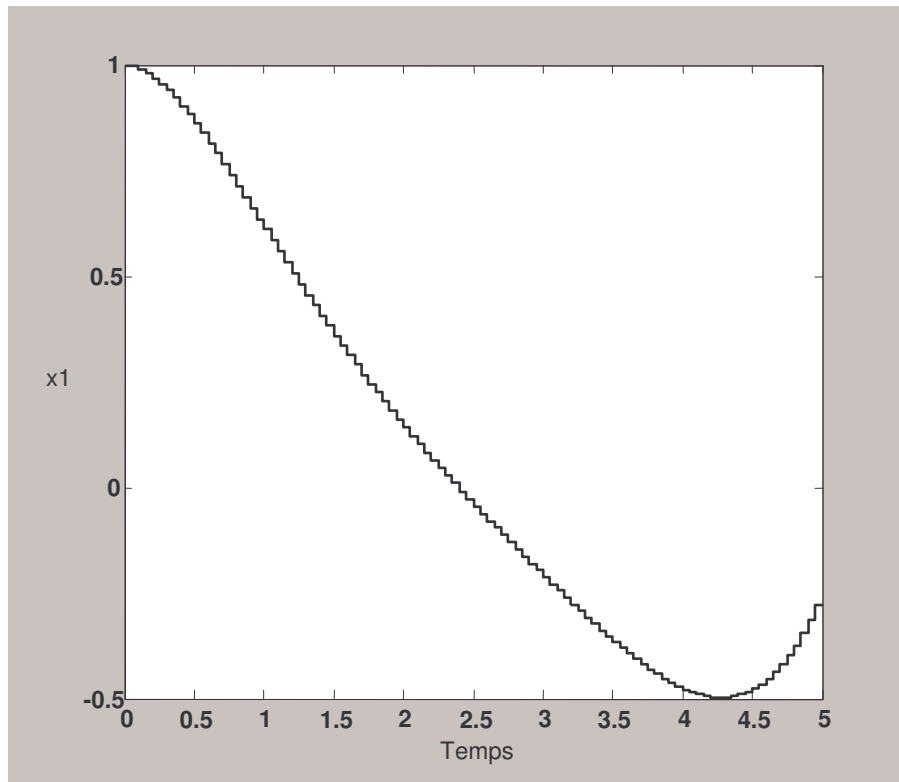


Fig. 4.16. Variation de x_1 en fonction du temps pour $m=100$

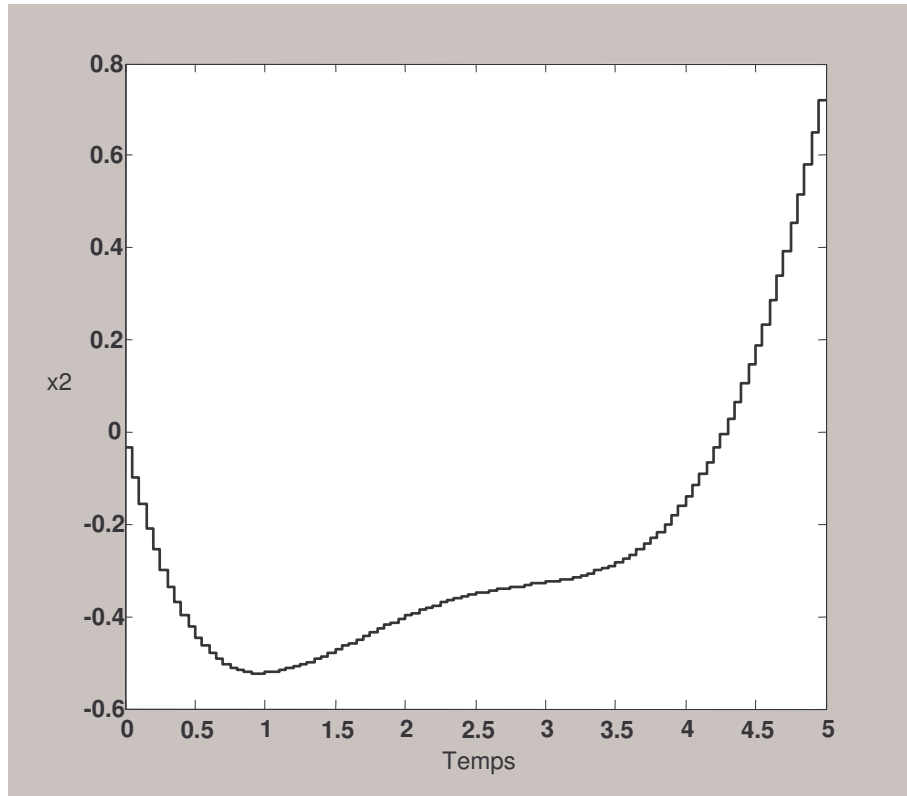


Fig. 4.17. Variation de x_2 en fonction du temps pour $m=100$

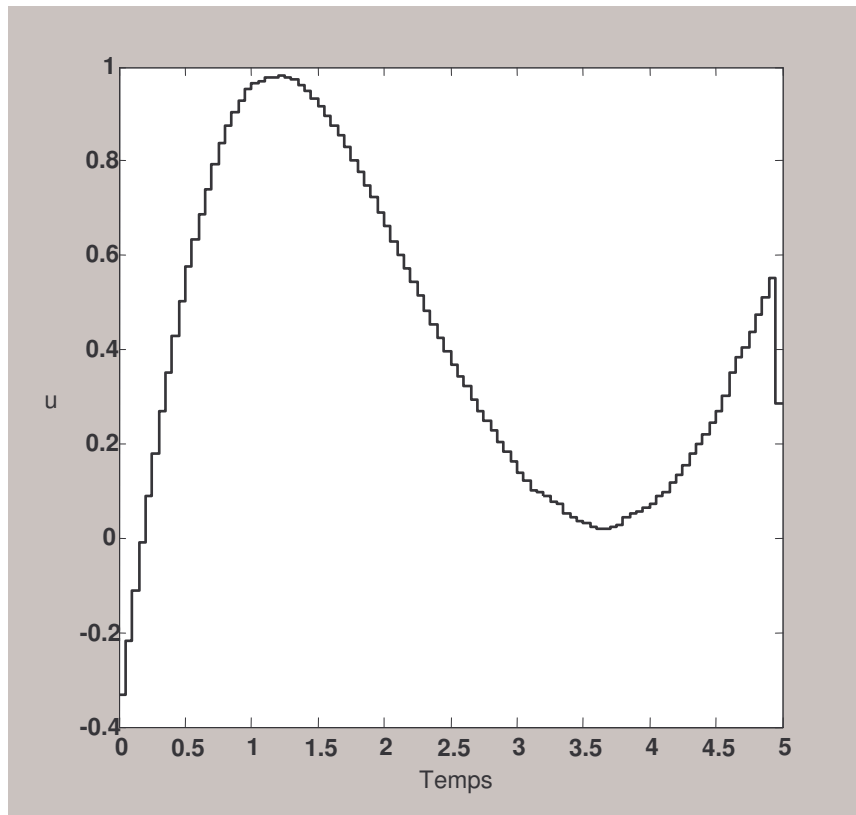


Fig. 4.18. Variation de u en fonction du temps pour $m=100$

Le tableau 4.3 montre une comparaison entre la valeur de l'indice de performance obtenue par notre méthode et celles obtenues par d'autres méthodes.

Source	J
Bullock et Franklin [59]	1.6857
Bashein et Enns [55]	1.6905756
H. Jaddu [88]	1.6857113
Notre méthode pour $m=50$	1.7051
Notre méthode pour $m=100$	1.6962

Tableau 4.3. Valeurs optimales de J relatif à l'exemple 3

Malgré la simplicité de notre méthode, elle donne des résultats comparables à celles obtenues par d'autres chercheurs.

Exemples 4 [88]

C'est le même problème défini dans l'exemple 3, seulement le système est soumis aux contraintes suivantes :

$$|u(t)| \leq 0.75 \quad (4.51)$$

$$\eta_1 = x_1(t_f) + 1 = 0 \quad (4.52)$$

$$\eta_2 = x_2(t_f) = 0 \quad (4.53)$$

Donc le problème d'optimisation est défini (4.44), (4.45), (4.46), (4.51), (4.52) et (4.53). En utilisant l'approximation via les BPFs (4.51), (4.52) et (4.53) prennent la forme suivante :

$$-0.75 \leq C_i^u \leq 0.75 \quad i = 1, \dots, m \quad (4.54)$$

$$C_m^{x_1} + 1 = 0 \quad (4.55)$$

$$C_m^{x_2} = 0 \quad (4.56)$$

Donc le problème original s'est converti à un problème de programmation non linéaire défini par (4.48), (4.49), (4.54), (4.55) et (4.56). L'implémentation de cet algorithme permet d'obtenir les courbes de (4.19) à (4.24) relatives à $m=50$ et $m=100$.

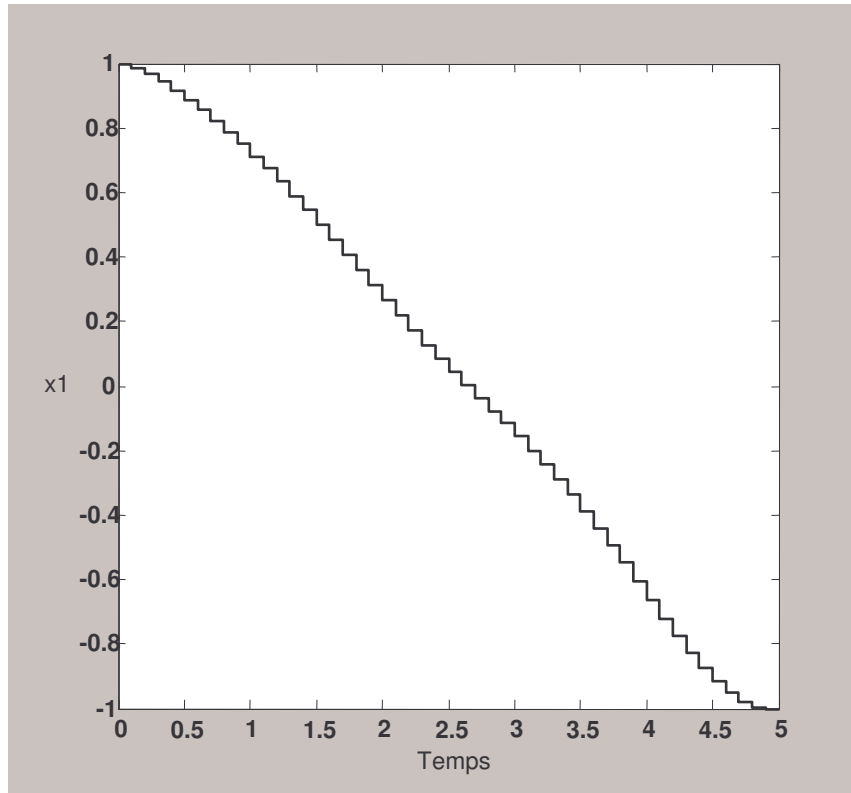


Fig. 4.19. Variation de x_1 en fonction du temps pour $m=50$

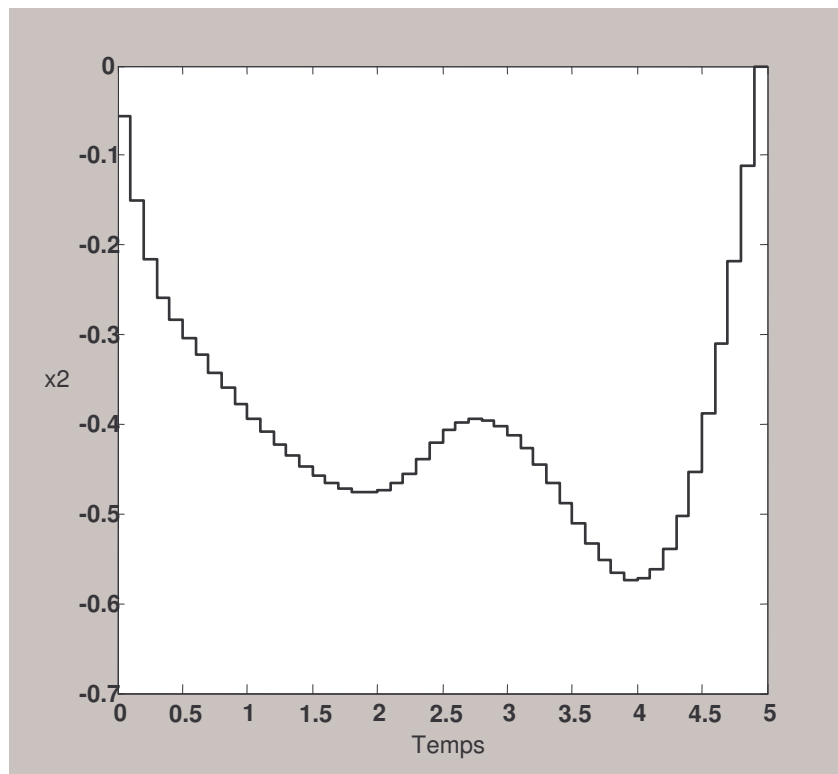
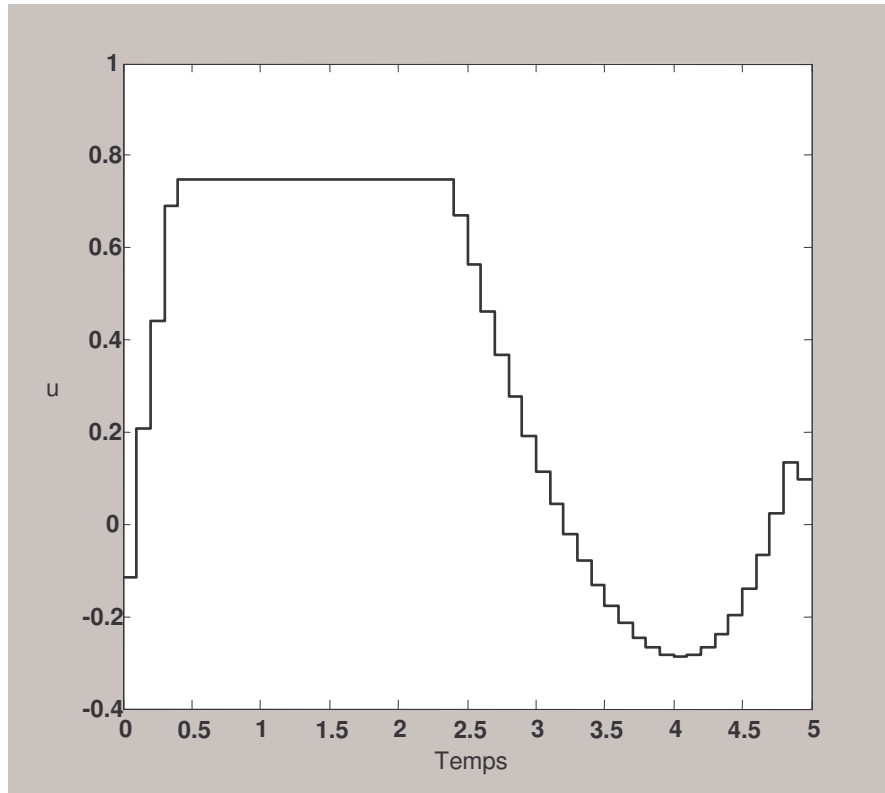
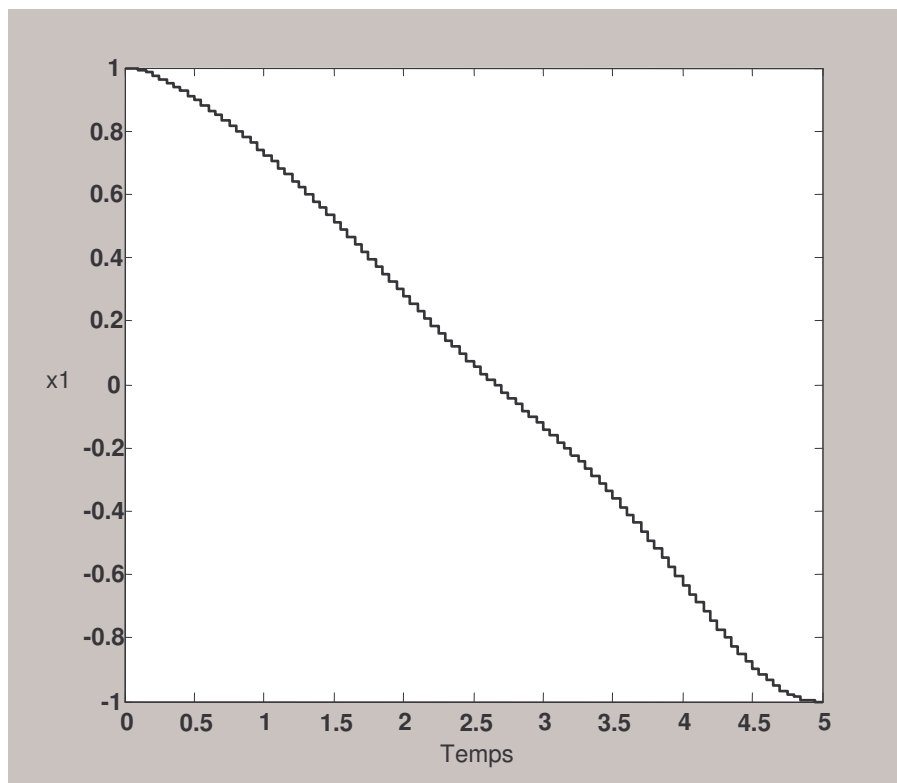


Fig. 4.20. Variation de x_2 en fonction du temps pour $m=50$

Fig. 4.21. Variation de u en fonction du temps pour $m=50$ Fig. 4.22. Variation de x_1 en fonction du temps pour $m=100$

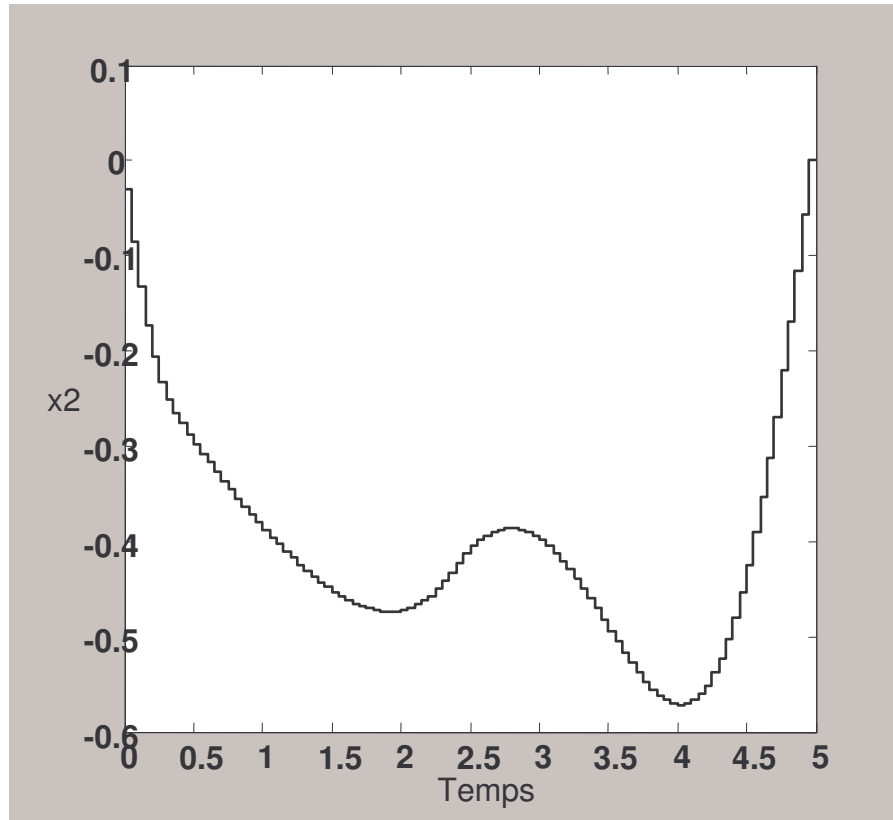
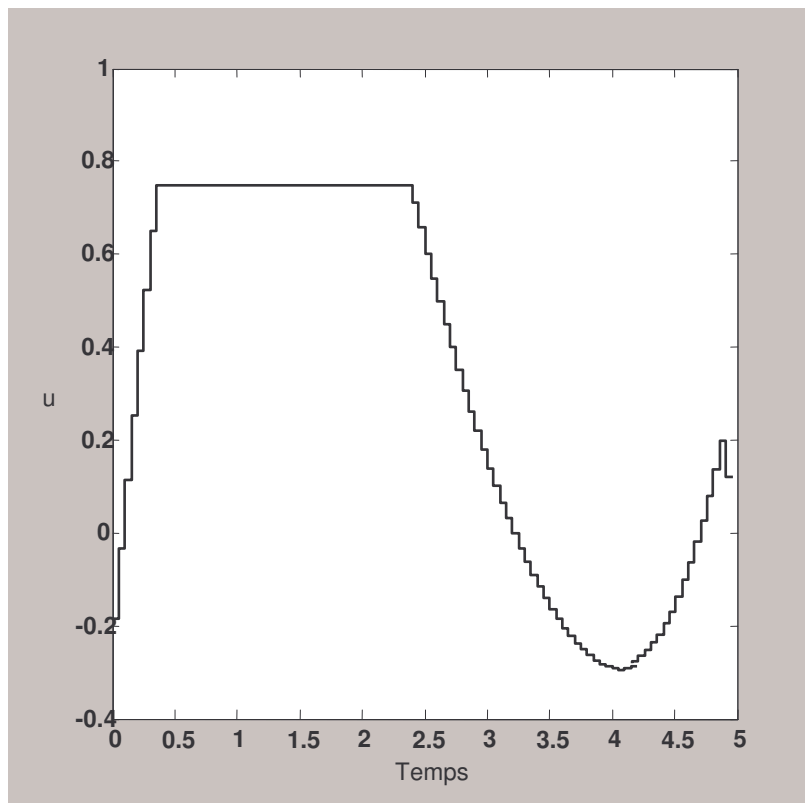


Fig. 4.23. Variation de x_2 en fonction du temps pour $m=100$



4.24. Variation de u en fonction du temps pour $m=100$

Ce problème a été étudié par certains auteurs, le tableau 4.4 inclut à titre de comparaison leurs résultats et les nôtres. On remarque que nos résultats sont comparables à ceux obtenus par les autres auteurs.

Source	J
Bashein et Enns [55]	2.1439039
H. Jaddu [88]	2.1443893
Notre méthode pour $m=50$	2.1537
Notre méthode pour $m=100$	2.1447

Chapitre 5

Problème de la Commande Optimale Non Linéaire Via les Algorithmes Génétiques et les BPFs

5.1 Introduction

Les méthodes utilisées jusqu'ici pour résoudre les problèmes d'optimisation résultants de l'approximation du problème de commande optimale via les BPFs étaient basées sur la programmation quadratique séquentielle (SQP methods). Ces méthodes sont, généralement, très rapides mais la solution globale peut s'avérer difficile à atteindre lorsque le système considéré est fortement non linéaire ou multimodale comme, par exemple, le système du réacteur étudié par Hartig et al. [98]. De même, dans le problème d'optimisation du réacteur à catalyseur bi fonctionnel, Luus et al. [99] ont montré que l'utilisation de la SQP conduit à plus de 20 optimums locaux. Cependant la recherche pour l'élaboration de méthodes alternatives servant à résoudre les problèmes de commande optimale est d'actualité.

Dans les années soixante plusieurs auteurs ont essayé d'utiliser la programmation dynamique pour résoudre les problèmes de commande optimale, cependant cette méthode souffrait de nombreux inconvénients qui limitait son utilisation à des problèmes de faibles dimensions. Pour surmonter les imperfections de la programmation dynamique, Luus a introduit l'utilisation de la programmation dynamique de manière itérative. Depuis, cette méthode a connu des améliorations successives et a été utilisé pour résoudre des problèmes fortement non linéaires de commande optimale des systèmes multivariables [100].

Une autre alternative pour résoudre les problèmes de commande optimale est l'utilisation des méthodes évolutionnaires. Récemment Pham a appliqué la méthode évolutionnaire dans la commande optimale de plusieurs processus chimiques où pour la discrétisation des variables de commande, il a utilisé la technique de paramétrisation de la commande, en revanche une méthode évolutionnaire d'optimisation, qui est une classe de procédures aléatoires de recherche qui imitent

les mécaniques de l'évolution biologique naturelle, est utilisée au lieu de la programmation non linéaire. En raison du caractère aléatoire de cette recherche évolutionnaire, la méthode est tout à fait robuste mais conduit parfois à une convergence lente. Pour améliorer la vitesse de convergence Storn et Price ont présenté l'algorithme de l'évolution différentielle (DE) [101]. Wang et Chiou ont utilisé un algorithme hybride basé sur la combinaison de l'algorithme DE avec l'algorithme SQP pour résoudre quelques problèmes de génie chimique [102].

Dans le cadre de cette thèse, une autre méthode hybride est proposée. Elle consiste en la combinaison des algorithmes génétiques avec l'algorithme de Newton pour résoudre le problème de la commande optimale non linéaire (sans contraintes ou soumis à des contraintes de saturation sur la commande) paramétrisé via les BPFs. Notre méthode présente l'avantage d'être simple à implémenter et capable de résoudre des problèmes de dimension très élevée.

Ce chapitre rappelle, d'abord, le principe des algorithmes génétiques. Ensuite le problème de commande optimale est formulé et paramétrisé via les BPFs. Les étapes de l'algorithme proposé pour résoudre le problème d'optimisation non linéaire sont présentées. Enfin un exemple d'application réel traité par d'autres auteurs clôturant ce chapitre.

5.2 Principe des algorithmes génétiques simples [104].

Les algorithmes génétiques (AG) font partie d'une famille de méthodes stochastiques appelée *méthodes évolutionnistes* qui reposent sur une analogie avec la *théorie de l'évolution naturelle de Darwin*, selon laquelle les individus d'une population les mieux adaptés à l'environnement ont une plus grande probabilité de survivre et de se reproduire de génération en génération en donnant des descendants encore mieux adaptés.

Les AG ont été proposés par Holland en 1975, puis développées par d'autres chercheurs tels que De Jong, Goldberg et Michalewics. Ils sont actuellement une des méthodes les plus diffusées et les plus utilisées dans la résolution de problèmes d'optimisation dans de nombreux domaines d'application.

Depuis le modèle proposé par Holland en 1975, il y a eu différentes implémentations concernant les AG. Néanmoins, le principe sur lequel la méthode est basée reste toujours le même. Nous allons présenter ici les notions de base des AG ainsi que ses aspects les plus importants.

5.2.1 Terminologie

Les AG définissent l'ensemble des paramètres du problème à optimiser comme étant l'*individu* de la théorie de Darwin. L'*environnement* de cet individu est représenté par l'espace de recherche du problème d'optimisation, tandis que son *adaptation* à l'environnement est donnée par la valeur de la fonction objectif évaluée sur lui. Finalement, la *population* auquel il appartient est donnée par un ensemble de différentes configurations de paramètres, alors que les *générations* sont représentées par les itérations du processus d'optimisation.

5.2.2 Description de la Méthode

L'algorithme de résolution commence avec la création d'une population P de taille $N > 0$ constituée par des individus générés aléatoirement. Ensuite, on mesure l'adaptation de chacun des individus de P à partir de la fonction objectif évaluée sur eux. La prochaine étape consiste à faire évoluer cette population vers une population plus adaptée à chaque génération en utilisant trois différents opérateurs : la *sélection*, le *croisement* et la *mutation*. Lorsque nous n'avons plus d'amélioration dans l'adaptation des individus de la population, l'algorithme s'arrête. La figure 5.1 illustre le processus d'optimisation développé par les AG.

A. Opérateur de Sélection : La sélection est un opérateur génétique appliqué sur la population courante de façon à sélectionner les individus qui iront former la population de la prochaine génération. La sélection de ces individus est basée sur leur valeur d'adaptation. Ainsi, les individus les plus adaptés sont généralement sélectionnés pour constituer la génération suivante, alors que les plus faibles sont exclus sans avoir la possibilité d'avoir des descendants, comme nous montre la figure 5.2.

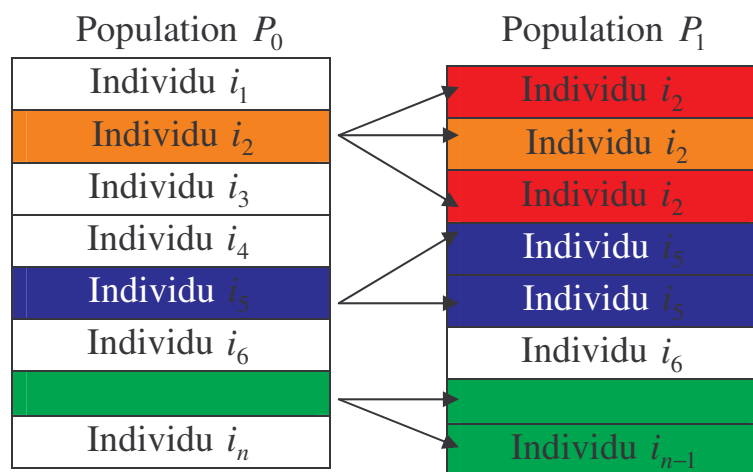


Fig. 5.2 Sélection des individus d'une population

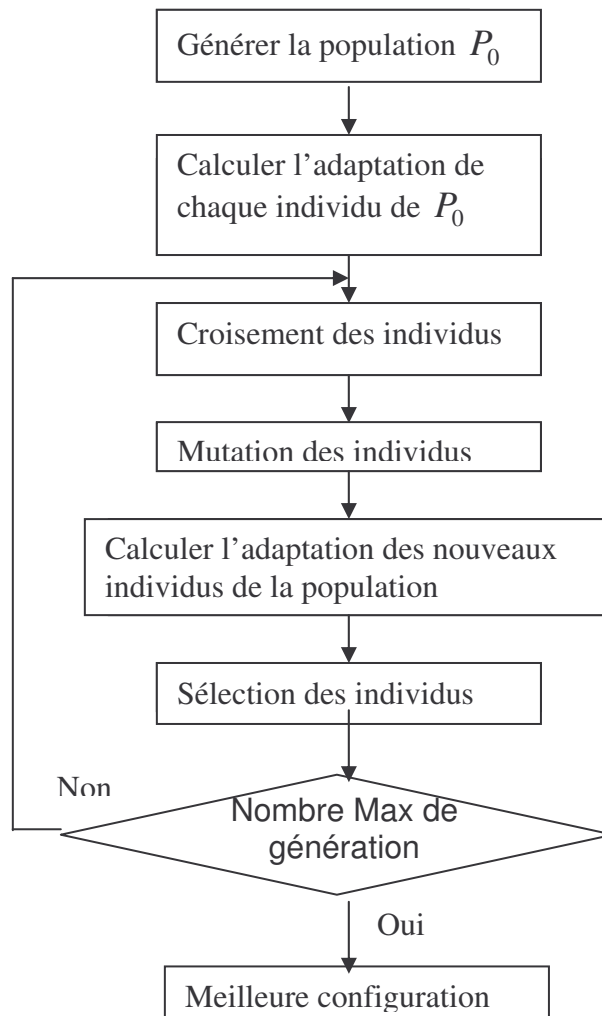


Fig.5.1 Algorithme génétique standard

Il existe différentes façons d'implémenter un opérateur de sélection, parmi lesquelles nous citons : la *sélection proportionnelle* et la *sélection par rang*. Néanmoins, le processus de sélection développé par ces différents mécanismes est toujours divisé en deux étapes. La première étape consiste à attribuer à chaque individu i_j un nombre réel p_j qui représente le nombre de descendants attendu pour lui dans la génération suivante. Selon le mécanisme de sélection utilisé, la valeur de p_j est calculée directement ou indirectement à partir de la valeur d'adaptation de l'individu.

Dans le cas d'une *sélection proportionnelle*, p_j est calculé directement par :

$$p_j = \frac{f(i_j)}{\sum_{k=1}^N f(i_k)} N, \text{ où } N \text{ est la taille de la population}$$

Dans la *sélection par rang*, on calcule la valeur de p_j en fonction du rang k_j que l'individu occupe dans la population. Cette valeur est obtenue à partir d'une liste où les meilleurs individus sont dans les premières positions, tandis que les moins performants y occupent les dernières. La valeur de p_j est alors donnée par :

$$p_j = 1 + p_{sel} - \frac{2k_j p_{sel}}{N-1}, \text{ où } p_{sel} \text{ est la pression de sélection, définie entre 0 et 1, qui permet de spécifier si tous les individus auront une chance d'être sélectionnés } (p_{sel} = 0) \text{ ou seulement les individus les plus performants } (p_{sel} = 1).$$

La deuxième étape du processus de sélection consiste à convertir la valeur du p_j de chaque individu en un nombre de descendants que chacun entre eux aura effectivement dans la prochaine génération. Cette conversion est obtenue à l'aide d'un algorithme d'échantillonnage qui transforme les valeurs réelles des p_j en valeurs entières. Les deux algorithmes d'échantillonnage les plus utilisés sont la *Roue de Loterie* et la *Roue de Loterie Généralisée*.

Dans l'algorithme d'échantillonnage proposé par Holland, on crée une roue de loterie divisée en secteurs proportionnels au p_j de chaque individu. Ensuite, on fait tourner la roue un nombre de fois égal à la taille de la population. A chaque coup, on prend une copie (descendant) de l'individu désigné par l'aiguille de la roue pour faire partie de la nouvelle population. Ainsi, les individus avec plus grand p_j auront un plus grand secteur dans la roue et en conséquence une plus grande probabilité de participer à la génération suivante. La figure 5.3 montre une représentation graphique de l'algorithme.

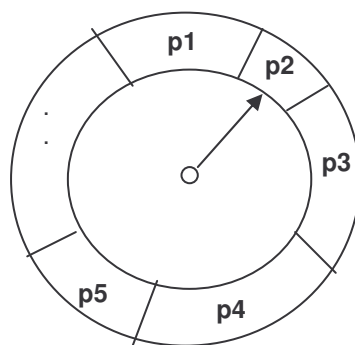


Fig.5.3 Echantillonnage par roue de loterie

L’algorithme d’échantillonnage proposé par Baker est basé sur le même principe que celui proposé par Holland, à la différence que la roue de loterie contient un nombre d’aiguilles égal à la taille de la population et on ne la fait tourner qu’une seule fois. Le nombre de copies de chaque individu sera alors donné par la position indiquée par chaque aiguille (figure 5.4).

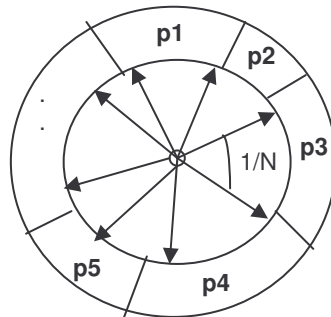


Fig.5.4.Echantillonnage par roue de loterie généralisée

A la fin de l’application de l’opérateur de sélection, la nouvelle population contiendra une plus grande portion des meilleurs individus de la génération précédente. On passe alors à l’étape de reproduction, dans laquelle seront utilisés les opérateurs de *croisement* et de *mutation*.

B. Opérateur de Croisement. L’opérateur de croisement est utilisé pour échanger les caractéristiques génétiques entre les différents individus d’une génération quelconque. Cet échange s’effectue en choisissant deux individus au hasard qui seront croisés avec une certaine probabilité de croisement p_c de façon à générer deux nouveaux individus. Dans le cas où nous utilisons le codage réel pour représenter les individus, ce croisement peut être obtenu à partir d’un simple échange entre les deux parents, comme nous montre la figure 5.5.

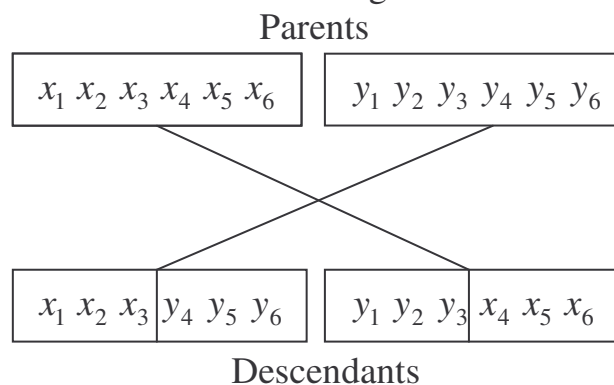


Fig.5.5 Croisement entre deux individus

Le croisement sur la figure 5.5 est du type 1-point. Nous avons encore d’autres implémentations de croisement, tels que le type 2-points, le croisement uniforme et le croisement arithmétique. Malgré ces différentes façons de croiser les individus, le

but de ces opérateurs reste toujours la conquête de nouvelles régions de l'espace de recherche à partir de l'échange de caractéristiques entre les individus de la population.

C. Opérateur de Mutation. L'opérateur de mutation est appliqué sur les individus d'une population de façon à obtenir d'autres individus avec des nouvelles caractéristiques génétiques. Dans le cas d'un codage réel, le mécanisme de mutation peut être implémenté en choisissant un individu de la génération courante au hasard et en modifiant un de ses paramètres aléatoirement avec une probabilité de mutation p_m . Ce mécanisme est dénommé mutation uniforme et il est illustré dans la figure 5.6.

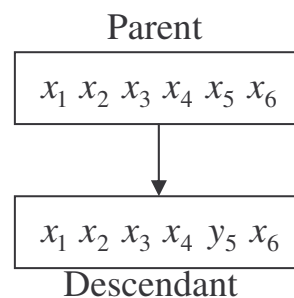


Fig.5.6 Mutation d'un individu

Il existe encore d'autres manières d'implémenter une mutation, telles que la mutation non uniforme et la mutation aux bornes. Ainsi comme les opérateurs de croisement, le but de tous ces opérateurs de mutation est d'atteindre des nouvelles régions de l'espace de recherche.

En utilisant les trois opérateurs que nous venons de décrire, les meilleurs individus se propagent de génération en génération en se combinant ou en échangeant leurs meilleurs caractéristiques. En favorisant les meilleurs individus, les régions les plus prometteuses de l'espace de recherche sont explorées, ce qui permet d'atteindre un optimum global.

Dans la suite, les AG sont utilisés pour résoudre le problème de la commande optimale des systèmes non linéaires soumis à des contraintes de saturation sur la commande.

5.3. Commande Optimale Basé sur les BPFs et les AG.

5.3.1 Position du Problème.

L'objectif est de trouver la commande optimale $u^*(t)$ qui minimise la fonctionnelle suivante :

$$J = S(x(t_f)) \quad (5.1)$$

Sachant les contraintes suivantes :

- Equations d'état du système : $\dot{x}_i(t) = f_i(x(t), u(t)) \quad i = 1, 2, \dots, n$ (5.2)

- La contrainte de saturation : $u_{\min_i} \leq u_i(t) \leq u_{\max_i} \quad i = 1, 2, \dots, r$ (5.3)

- Les conditions initiales : $x_i(t_0) = x_{i0}$ (5.4)

Où :

$x(t) \in R^n, u(t) \in R^r, f_i$ et S sont des fonctions scalaires définies sur R et t_0 et t_f sont, respectivement, les instants initial et final supposés fixes.

5.3.2 Paramétrisation du Problème via les BPFs

En utilisant l'approximation via les BPFs, le problème se réduira à un problème d'optimisation qui peut être énoncé comme suit : Il s'agit de minimiser la fonction :

$$J_m = S(C_m^{x_1}, C_m^{x_2}, \dots, C_m^{x_n}) \quad (5.5)$$

Sachant les contraintes suivantes :

- Approximation de l'équation d'état :

$$\begin{aligned} C_1^{x_i} - x_{i0} &= \frac{h}{2} f_i(C_1^x, C_1^u) \quad i = 1, 2, \dots, n \quad k = 1, 2, \dots, m \\ C_k^{x_i} - x_{i0} &= \frac{h}{2} f_i(C_k^x, C_k^u) + h \sum_{l=1}^{k-1} f_i(C_l^x, C_l^u) \end{aligned} \quad (5.6)$$

Où : $C_j^x = [C_j^{x_1}, C_j^{x_2}, \dots, C_j^{x_n}]$ et $C_j^u = [C_j^{u_1}, C_j^{u_2}, \dots, C_j^{u_r}] \quad j = 1, 2, \dots, m$

- Approximation de la contrainte de saturation :

$$u_{\min_i} \leq C_k^{u_i} \leq u_{\max_i} \quad i = 1, 2, \dots, r. \quad k = 1, 2, \dots, m \quad (5.7)$$

Donc le problème initial de commande optimale décrit par (5.1), (5.2), (5.3) et (5.4) s'est réduit à un problème d'optimisation décrit par (5.5), (5.6) et (5.7).

5.3.3. Solution du Problème via les AG.

L'objectif souhaiter est de trouver la commande $u^*(t)$, qui est spécifiée par les constantes de Fourier $C_1^{u_i}, C_2^{u_i}, \dots, C_m^{u_i}$ $i=1,2,\dots,r$, qui minimise un critère de performance J . Par conséquent la topologie du chromosome, dans le cas général, est le suivant :

$C_1^{u_1}$	$C_1^{u_2}$...	$C_1^{u_r}$	$C_2^{u_1}$	$C_2^{u_2}$...	$C_2^{u_r}$...	$C_m^{u_1}$	$C_m^{u_2}$...	$C_m^{u_r}$
-------------	-------------	-----	-------------	-------------	-------------	-----	-------------	-----	-------------	-------------	-----	-------------

Les paramètres $C_k^{u_i}$ $i=1,2,\dots,r$ $k=1,2,\dots,m$ peuvent être codés en binaire ou en utilisant un codage réel. Dans cette thèse, on a opté pour un codage binaire. Par conséquent, si chaque coefficient est codé par q bits, on aura un vecteur chromosome de dimension qmr . Si en plus, la taille de la population est p on obtient, pour la configuration de la population, une matrice de dimension $qmr \times p$. Ainsi la solution du problème de la commande optimale est menée selon les étapes suivantes.

Etape 1 : Générer la population de façon aléatoire.

Etape 2 : Décoder chaque élément de la population.

Etape 3 : Résoudre le système d'équation non linéaire (5.6) de façon itérative.

Etape 4 : Evaluer la population initiale en calculant la fonction objective (5.5) associée à chaque chromosome de la population.

Etape 5 : Tant que la précision voulue ou le nombre maximal de génération ne sont pas atteints, répéter les opérations suivantes :

- Assigner une valeur dite Fitness à l'ensemble de la population.
- Sélection des individus pour la reproduction.
- Croisement des individus avec une probabilité de croisement p_c .
- Mutation des individus avec une probabilité de mutation p_m .
- Résoudre le système d'équation (5.6) de façon itérative.
- Evaluation des individus sélectionnés.
- Réinsertion dans la population courante.

Etape 6 : Enregistrer les coefficients correspondants à la commande et aux trajectoires optimales ainsi que la valeur optimale du critère de performance.

5.4. Exemple d'Application.

L'exemple que nous allons traiter dans ce paragraphe a été décrit et étudié par R.Luus [103]; c'est un problème de commande optimale non linéaire et multivariable relatif à un réacteur chimique. Le lecteur intéressé par la modélisation pourra se référer à la référence.

Enoncé du Problème : Pour le système décrit par les équations d'état suivantes :

$$\begin{aligned}
 \frac{dx_1}{dt} &= u_4 - qx_1 - 17.6x_1x_2 - 23x_1x_6u_3 \\
 \frac{dx_2}{dt} &= u_1 - qx_2 - 17.6x_1x_2 - 146x_2x_3 \\
 \frac{dx_3}{dt} &= u_2 - qx_3 - 73x_2x_3 \\
 \frac{dx_4}{dt} &= -qx_4 + 35.2x_1x_2 - 51.3x_4x_6 \\
 \frac{dx_5}{dt} &= -qx_5 + 219x_2x_3 - 51.3x_4x_5 \\
 \frac{dx_6}{dt} &= -qx_6 + 102.6x_4x_5 - 23x_1x_6u_3 \\
 \frac{dx_7}{dt} &= -qx_7 + 46x_1x_6u_3
 \end{aligned} \tag{5.8}$$

$$\frac{dx_8}{dt} = 5.8(qx_1 - u_4) - 3.7u_1 - 4.1u_2 + q(23x_4 + 11x_5 + 28x_6 + 35x_7) - 5u_3^2 - 0.099$$

$$q = u_1 + u_2 + u_4$$

Il s'agit de trouver la commande $u = [u_1, u_2, u_3, u_4]$ qui minimise le critère de performance suivant :

$$J = x_8(t_f) \tag{5.9}$$

Sachant les contraintes suivantes :

- Les contraintes de saturation sur la commande :

$$0 \leq u_1 \leq 20; 0 \leq u_2 \leq 6; 0 \leq u_3 \leq 4; 0 \leq u_4 \leq 20 \tag{5.10}$$

- Les conditions initiales :

$$\begin{aligned} x_1(0) = 0.1883 \quad x_2(0) = 0.2507 \quad x_3(0) = 0.0467 \quad x_4(0) = 0.0899 \\ x_5(0) = 0.1804 \quad x_6(0) = 0.1394 \quad x_7(0) = 0.1046 \quad x_8(0) = 0 \end{aligned} \quad (5.11)$$

- Les instants initial et final :

$$t_0 = 0 \quad t_f = 0.2 \text{ heure}$$

Il est connu que ce problème peut être transformé à un problème de programmation non linéaire, et en vertu de (5.5), (5.6) et (5.7) il est exprimé comme suit : Trouver les coefficients $C_i^{u_1}, C_i^{u_2}, C_i^{u_3}$ et $C_i^{u_4}$ $i=1,2,\dots,m$ qui maximisent le critère de performance :

$$J_m = C_m^{x_8} \quad (5.12)$$

pour le système décrit par les équations algébriques non linéaires suivantes :

$$\left\{ \begin{aligned} C_1^{x_1} - x_1(0) &= \frac{h}{2} l_1^1 \\ C_1^{x_2} - x_2(0) &= \frac{h}{2} l_1^2 \\ C_1^{x_3} - x_3(0) &= \frac{h}{2} l_1^3 \\ C_1^{x_4} - x_4(0) &= \frac{h}{2} l_1^4 \\ C_1^{x_5} - x_5(0) &= \frac{h}{2} l_1^5 \\ C_1^{x_6} - x_6(0) &= \frac{h}{2} l_1^6 \\ C_1^{x_7} - x_7(0) &= \frac{h}{2} l_1^5 \\ C_1^{x_8} - x_8(0) &= \frac{h}{2} l_1^8 \end{aligned} \right. \quad (5.13')$$

$$\left\{ \begin{array}{l}
 C_k^{x_1} - x_1(0) = \frac{h}{2} l_k^1 + h \sum_{j=1}^{k-1} l_j^1 \\
 C_k^{x_2} - x_2(0) = \frac{h}{2} l_k^2 + h \sum_{j=1}^{k-1} l_j^2 \\
 C_k^{x_3} - x_3(0) = \frac{h}{2} l_k^3 + h \sum_{j=1}^{k-1} l_j^3 \\
 C_k^{x_4} - x_4(0) = \frac{h}{2} l_k^4 + h \sum_{j=1}^{k-1} l_j^4 \\
 C_k^{x_5} - x_5(0) = \frac{h}{2} l_k^5 + h \sum_{j=1}^{k-1} l_j^5 \\
 C_k^{x_6} - x_6(0) = \frac{h}{2} l_k^6 + h \sum_{j=1}^{k-1} l_j^6 \\
 C_k^{x_7} - x_7(0) = \frac{h}{2} l_k^7 + h \sum_{j=1}^{k-1} l_j^7 \\
 C_k^{x_8} - x_8(0) = \frac{h}{2} l_k^8 + h \sum_{j=1}^{k-1} l_j^8
 \end{array} \right. \quad k = 2, 3, \dots, m \quad (5.13'')$$

Où :

$$l_i^1 = C_i^{u_4} - Q_i C_i^{x_1} - 17.6 C_i^{x_1} C_i^{x_2} - 23 C_i^{x_1} C_i^{x_6} C_i^{u_3}$$

$$l_i^2 = C_i^{u_1} - Q_i C_i^{x_2} - 17.6 C_i^{x_1} C_i^{x_2} - 146 C_i^{x_2} C_i^{x_3}$$

$$l_i^3 = C_i^{u_2} - Q_i C_i^{x_3} - 73 C_i^{x_2} C_i^{x_3}$$

$$l_i^4 = -Q_i C_i^{x_4} + 35.2 C_i^{x_1} C_i^{x_2} - 51.3 C_i^{x_4} C_i^{x_6} \quad i = 1, 2, \dots, m$$

$$l_i^5 = -Q_i C_i^{x_5} + 219 C_i^{x_2} C_i^{x_3} - 51.3 C_i^{x_4} C_i^{x_5}$$

$$l_i^6 = -Q_i C_i^{x_6} + 102.6 C_i^{x_4} C_i^{x_5} - 23 C_i^{x_1} C_i^{x_6} C_i^{u_3}$$

$$l_i^7 = -Q_i C_i^{x_7} + 46 C_i^{x_1} C_i^{x_6} C_i^{u_3}$$

$$l_i^8 = Q_i (5.8 C_i^{x_1} + 23 C_i^{x_4} + 11 C_i^{x_5} + 28 C_i^{x_6} + 35 C_i^{x_7}) - 5.8 C_i^{u_4} -$$

$$- 3.7 C_i^{u_1} - 4.1 C_i^{u_2} - 5 (C_i^{u_3})^2 - 0.099$$

$$Q_i = C_i^{u_1} + C_i^{u_2} + C_i^{u_4}$$

et soumis aux contraintes suivantes :

$$0 \leq C_i^{u_1} \leq 20, 0 \leq C_i^{u_2} \leq 6, 0 \leq C_i^{u_3} \leq 4, 0 \leq C_i^{u_4} \leq 20 \tag{5.14}$$

$$i=1,2,\dots, m$$

L’algorithme génétique décrit précédemment est appliqué pour résoudre ce problème d’optimisation avec les données suivantes :

- Sélection uniforme (*Sampling Uniform Selection: SUS*)
- Croisement un point avec probabilité de croisement $p_c = 0.7$
- Mutation uniforme avec probabilité de mutation $p_m = 0.01$
- la taille de la population=60
- Le nombre maximal des générations=1000.
- Le fitness est calculé par la méthode de rangement (*Ranking Method*)

Le résultat de l’implémentation est observé dans les figures de 5.1 à 5.24 relatives à $m=10$ et $m=30$.

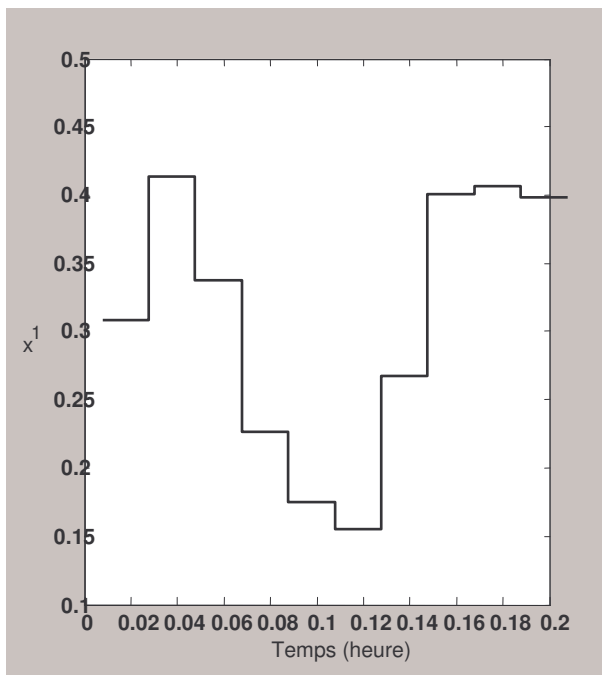


Fig.5.1. Variation de x_1 pour $m=10$

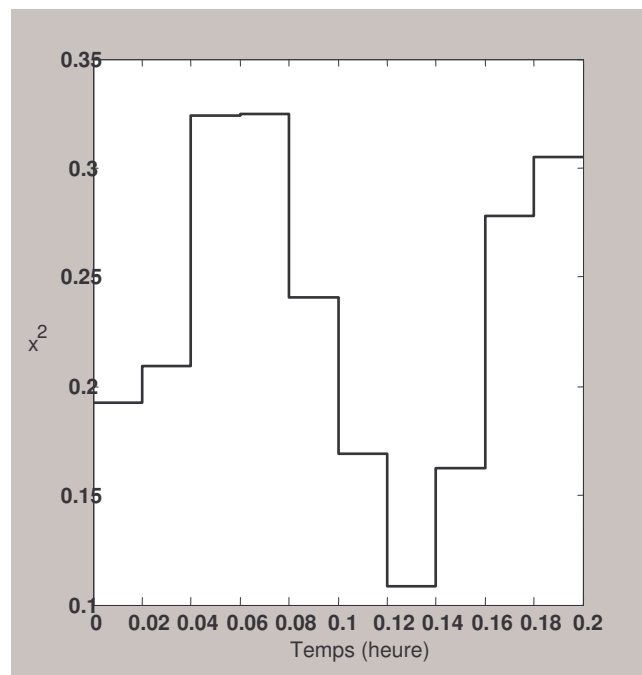


Fig.5.2. Variation de x_2 pour $m=10$

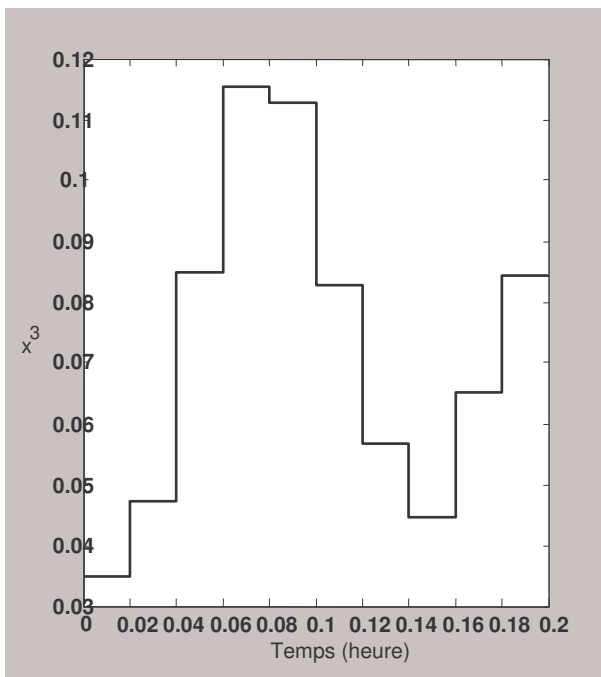


Fig.5.3. Variation de x_3 pour $m=10$

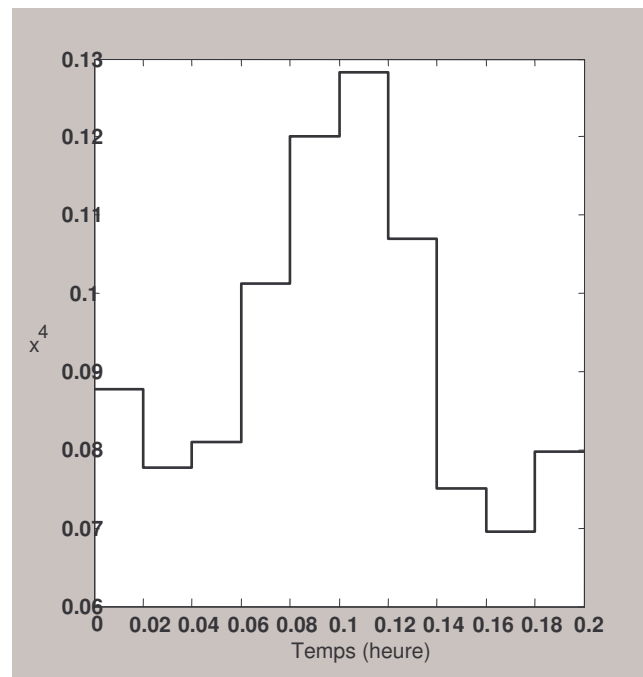


Fig.5.4. Variation de x_4 pour $m=10$

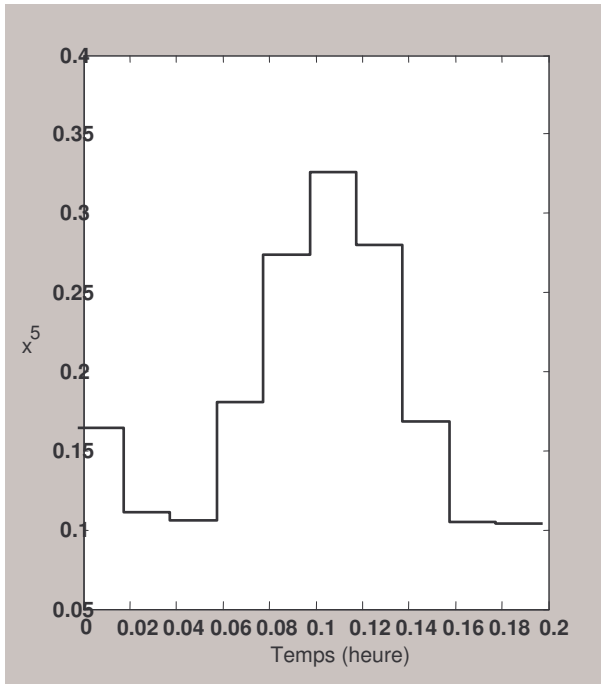


Fig.5.5. Variation de x_5 pour $m=10$

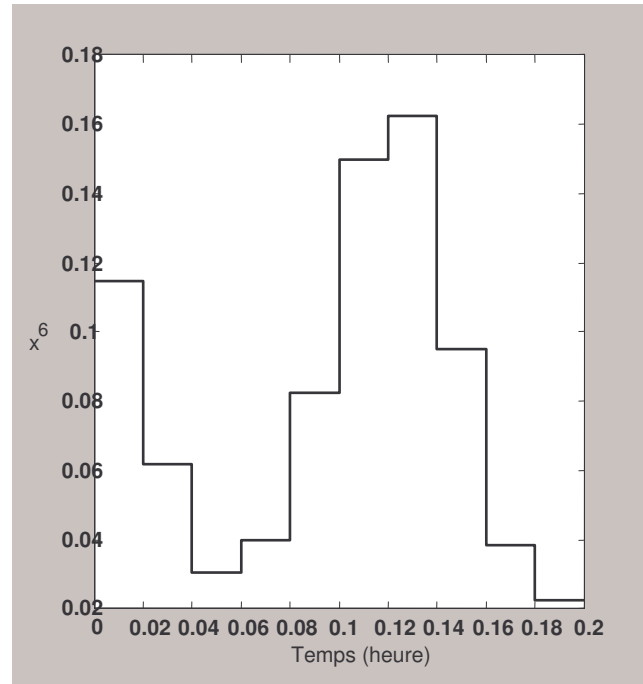


Fig.5.6. Variation de x_6 pour $m=10$

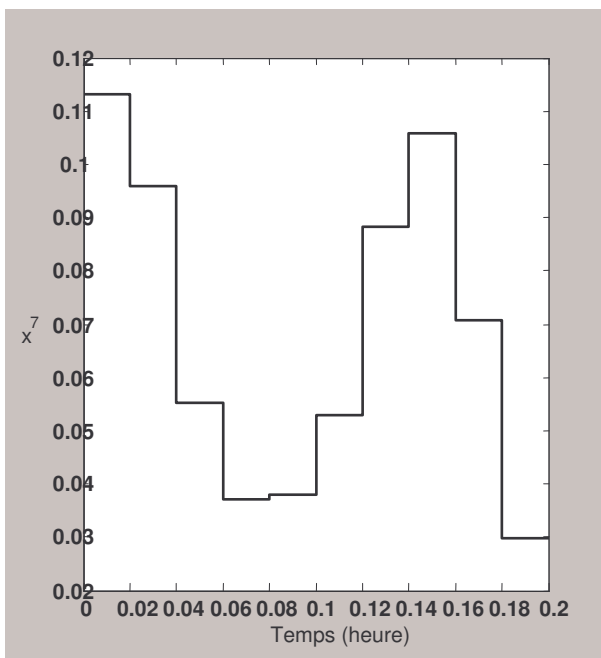


Fig.5.7. Variation de x_7 pour $m=10$

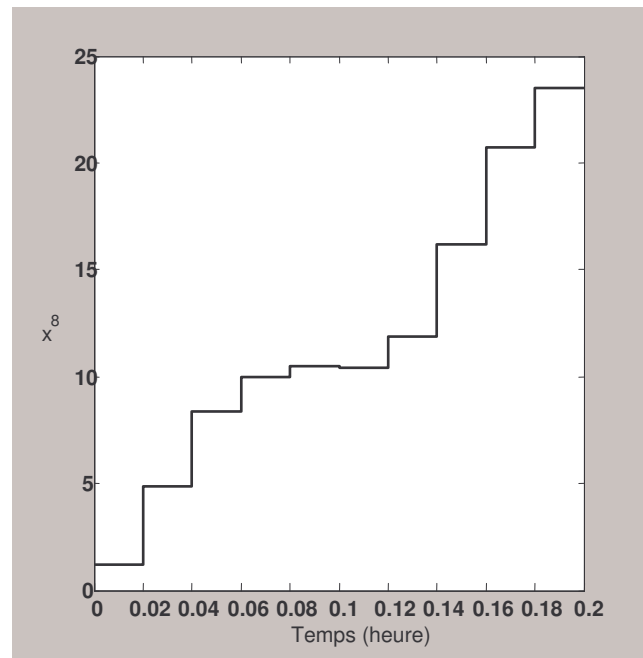


Fig.5.8. Variation de x_8 pour $m=10$

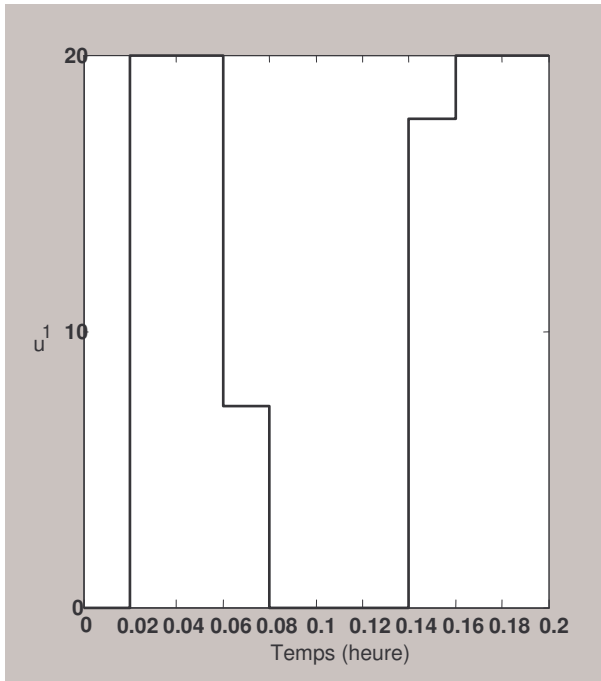


Fig.5.9. Variation de u_1 pour $m=10$

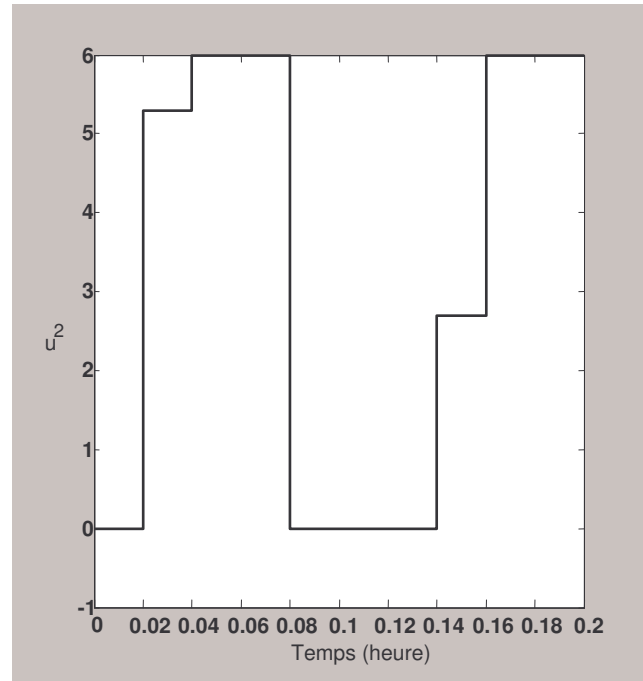


Fig.5.10. Variation de u_2 pour $m=10$

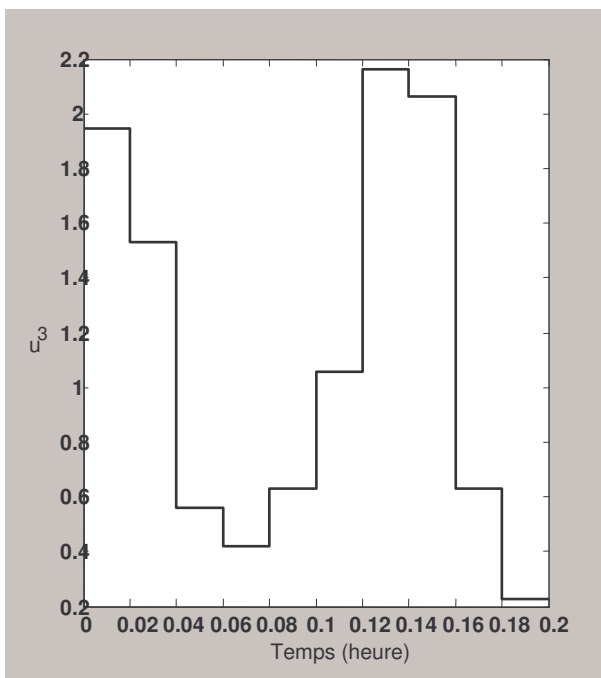


Fig.5.11. Variation de u_3 pour $m=10$

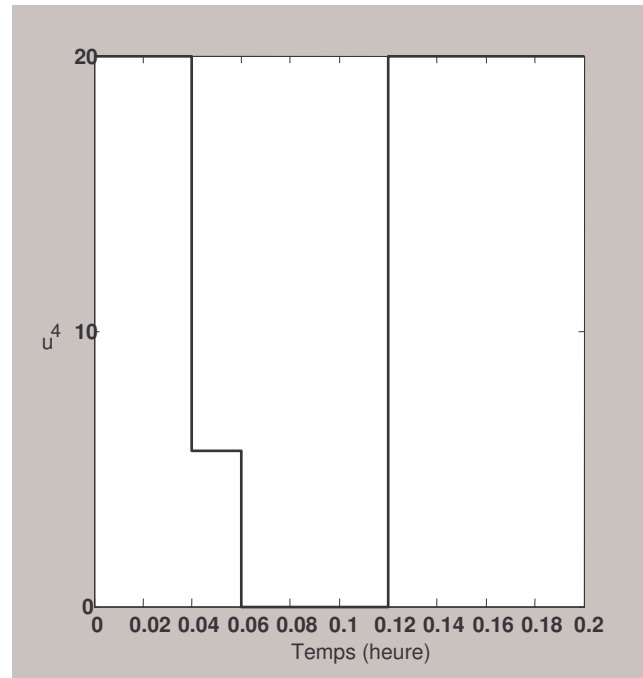


Fig.5.12. Variation de u_4 pour $m=10$

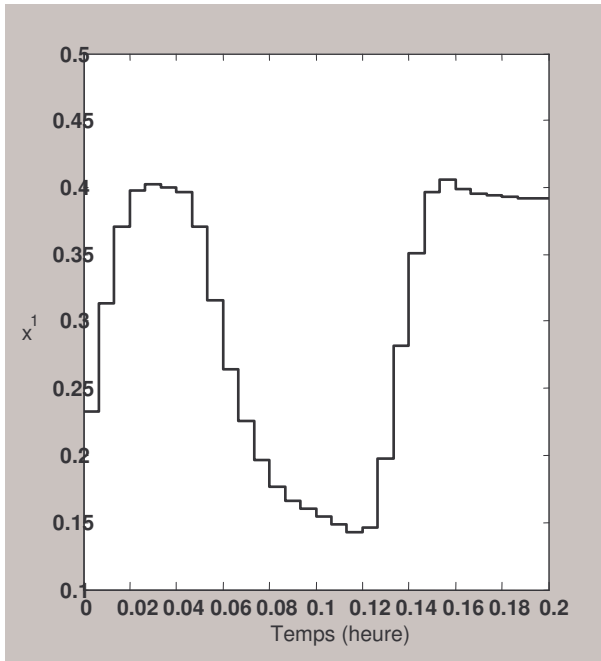


Fig.5.13. Variation de x_1 pour $m=30$

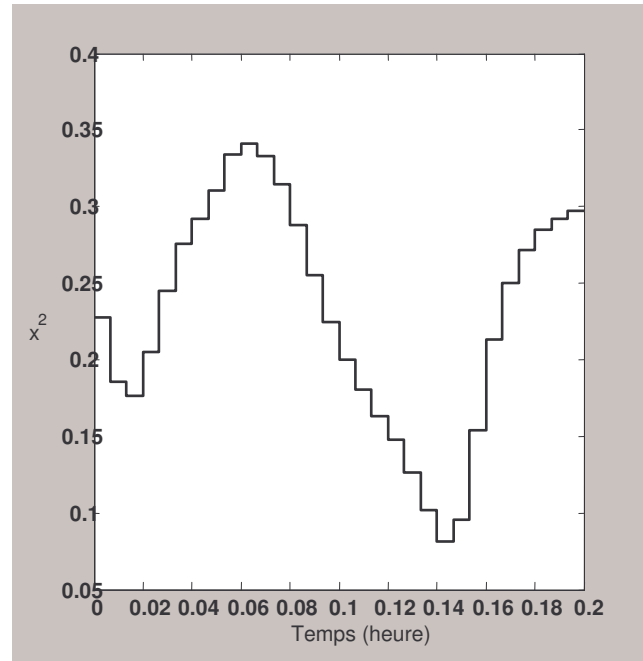


Fig.5.14. Variation de x_2 pour $m=30$

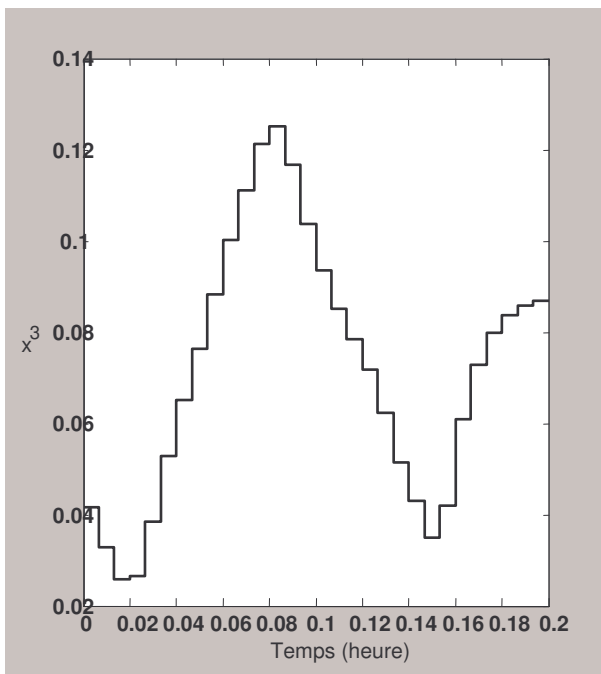


Fig.5.15. Variation de x_3 pour $m=30$

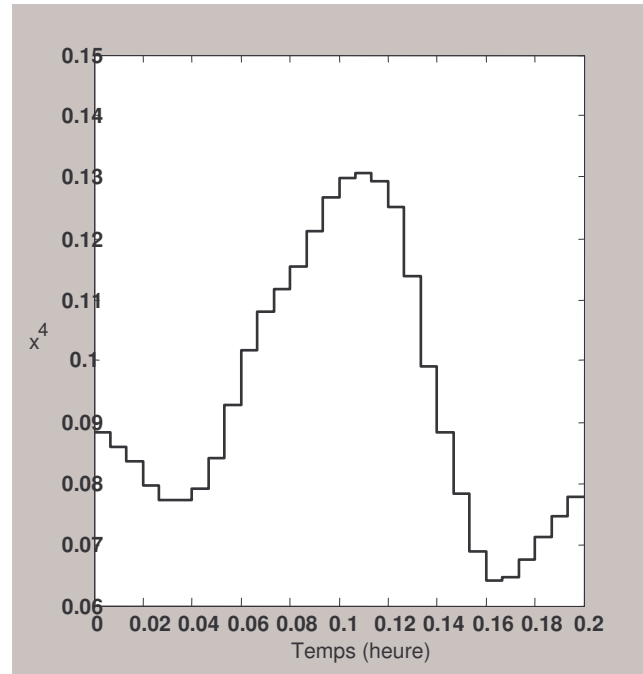


Fig.5.16. Variation de x_4 pour $m=30$

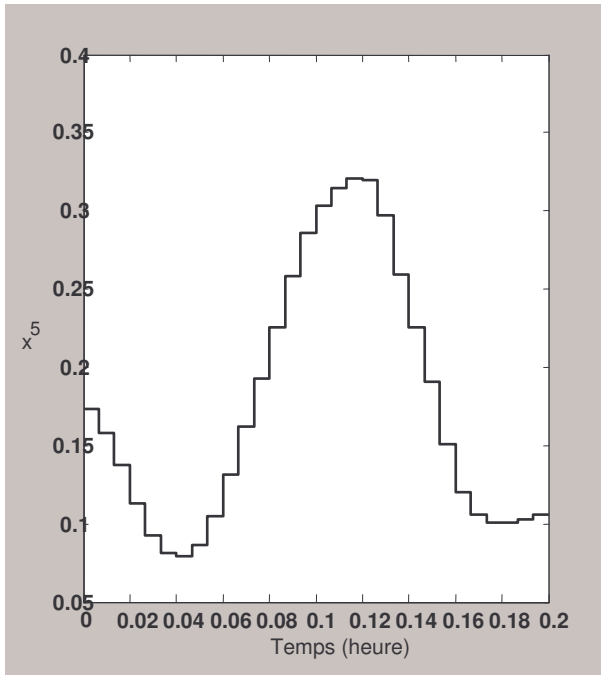


Fig.5.17. Variation de x_5 pour $m=30$

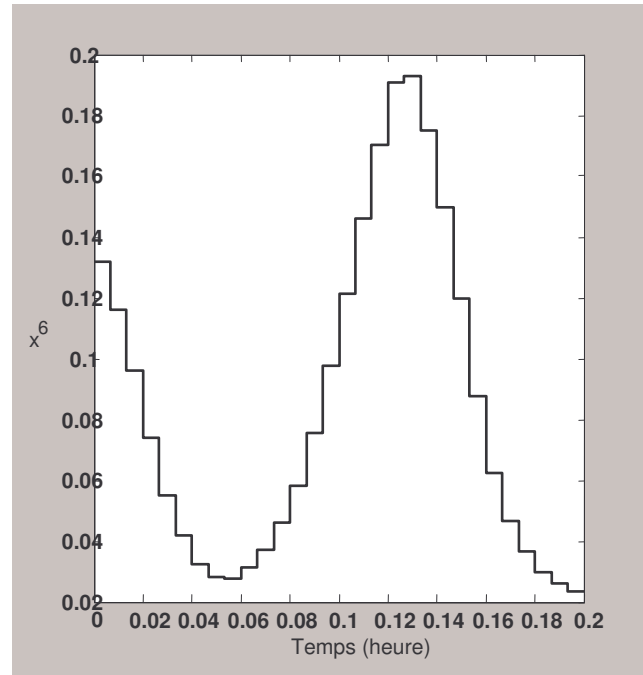


Fig.5.18. Variation de x_6 pour $m=30$

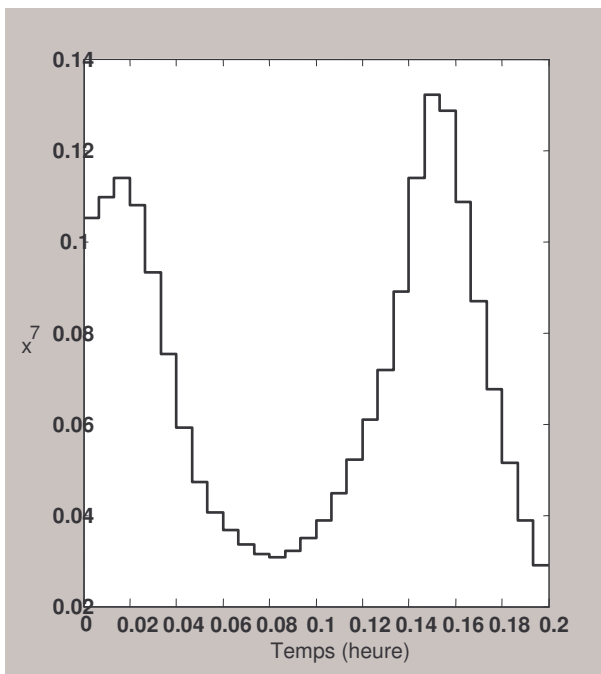


Fig.5.19. Variation de x_7 pour $m=30$

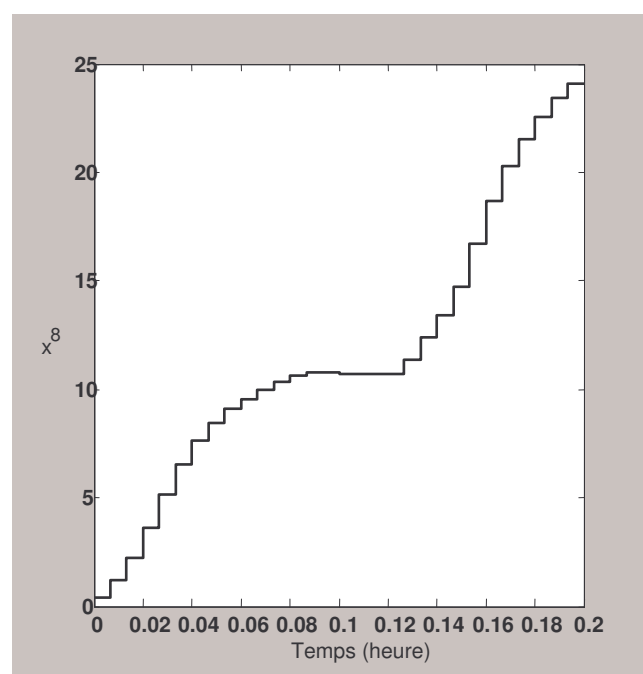


Fig.5.20. Variation de x_8 pour $m=30$

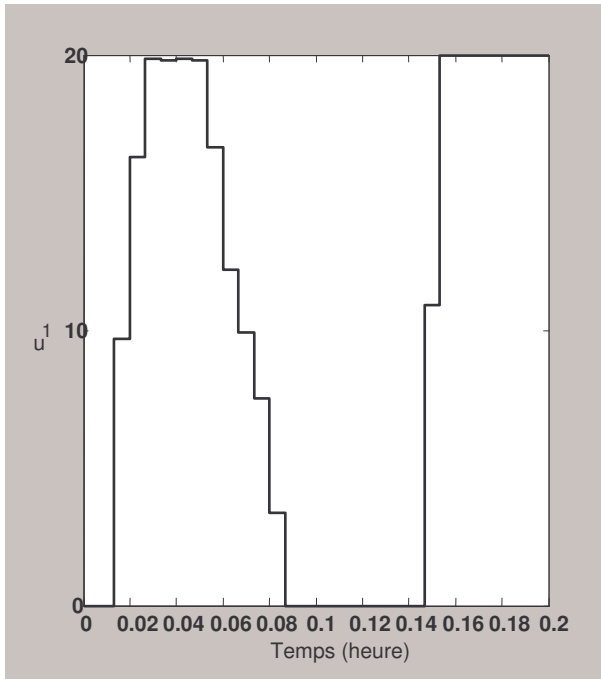


Fig.5.21. Variation de u_1 pour $m=30$

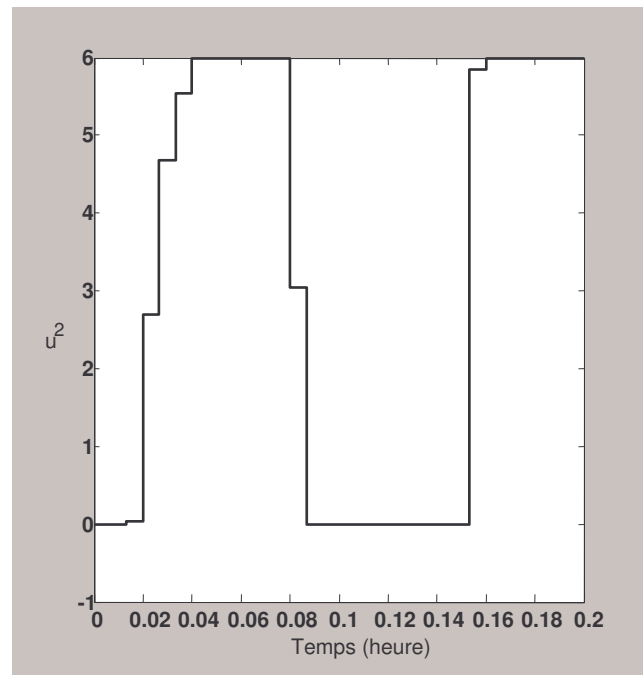


Fig.5.22. Variation de u_2 pour $m=30$

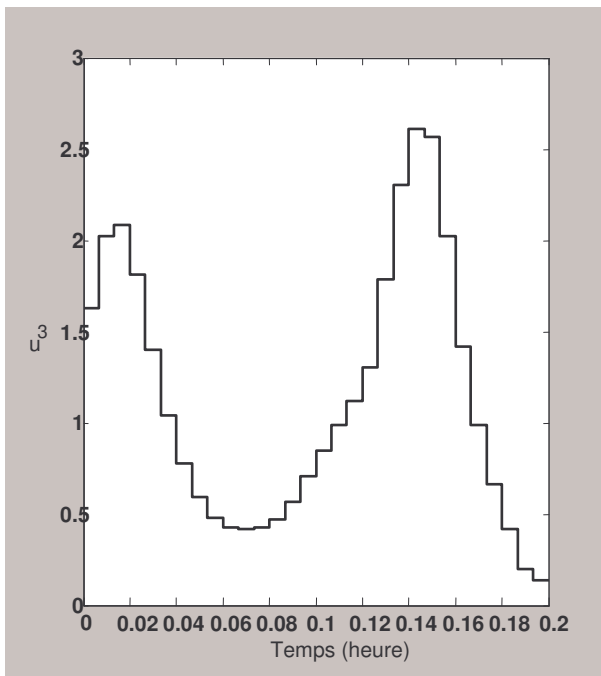


Fig.5.23. Variation de u_3 pour $m=30$

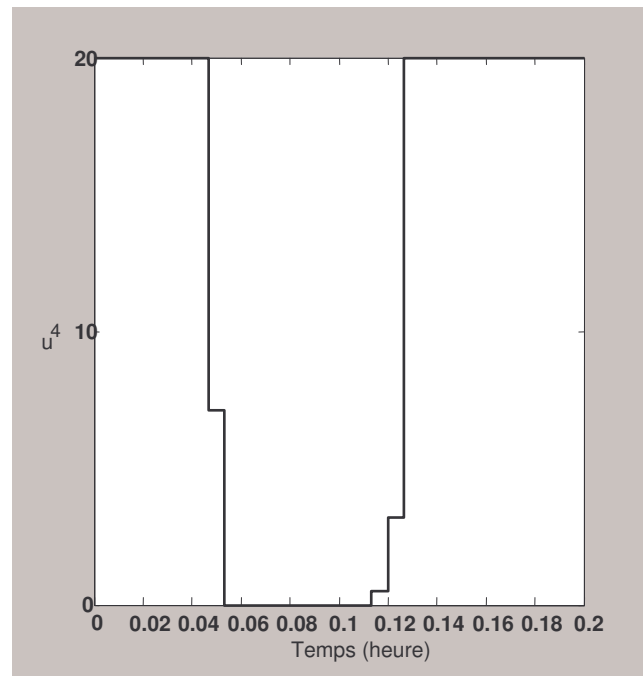


Fig.5.24. Variation de u_4 pour $m=30$

Nous remarquons bien que les courbes obtenues avec $m=30$ sont meilleures par rapport à celles obtenues avec $m=10$, mais ceci s'est réalisé au détriment du temps de

calcul. Par conséquent, il faut toujours chercher un compromis entre précision et rapidité.

Le tableau 5.1 et La figure 5.25 donnent une idée sur l'évolution du critère de performance en fonction de m .

m	1	2	3	4	5	6	7	8
J	12.1781	18.1559	19.6876	21.2025	22.3639	22.8801	23.0543	23.2943
m	9	10	11	12	13	14	15	16
J	23.3868	23.4936	23.5794	23.6824	23.7387	23.7621	23.7019	23.8658
m	17	18	19	20	21	22	23	24
J	23.8934	23.9146	23.9550	23.9790	24.0004	24.0184	24.0384	24.0656
m	25	26	27	28	29	30		
J	24.0636	24.0768	24.0988	24.1090	24.1103	24.1223		

Tableau 5.1 Valeurs de J correspondantes à m

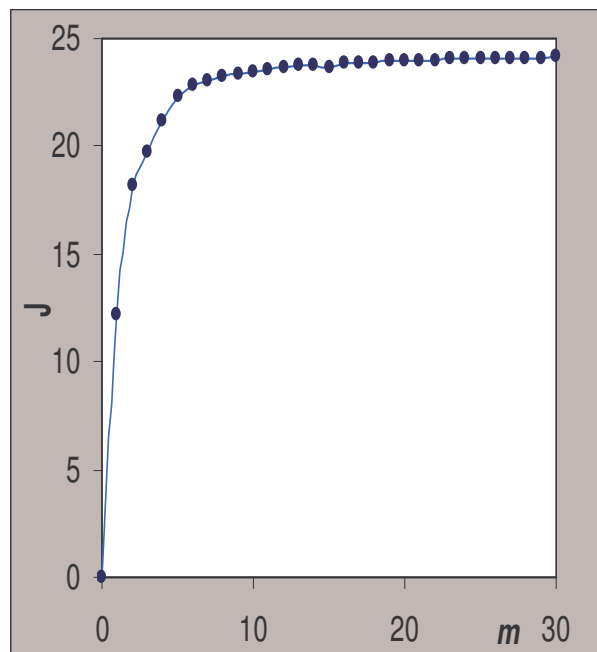


Fig. 5.25 Evolution de J en fonction de m

Nous pouvons enregistrer, à partir de la figure 5.25, une convergence à partir d'une valeur de m sensiblement égale à 10 à laquelle correspond une valeur de J égale à 23.4936 ce qui est, loin, meilleure que celle obtenue par R. Luus avec les mêmes données. Ceci est une conséquence de la méthode utilisée, d'une part, mais aussi des

performances des micro-ordinateurs qui ne cessent de s'améliorer chaque jour, d'autre part.

5.5 Conclusion

Nous avons présenté une méthode basée sur les BPFs comme moyen de réduction du problème de commande optimale à un problème de programmation non linéaire lequel est solutionné via l'utilisation des algorithmes génétiques en vue de l'obtention de l'optimum global. En fait notre algorithme est hybride dans le sens où il intègre une procédure de résolution d'un système d'équations non linéaires de façon récursive. Par conséquent le vecteur chromosome n'était constitué que des coefficients relatifs à la commande, donc de taille réduite, et le temps d'exécution s'était amélioré par rapport aux méthodes basées sur les AG simples.

Chapitre 6

Conclusions et Perspectives

6.1 Conclusions

Dans cette thèse nous avons proposé des méthodes numériques pour résoudre plusieurs types de problèmes de commande optimale. Ces méthodes sont basées sur la paramétrisation des variables d'état et de commande en utilisant les fonctions à blocs d'impulsion. L'utilisation des BPFs est motivée par différents avantages qu'elles offrent par comparaison avec d'autres fonctions d'approximation.

Les méthodes numériques proposées dans cette thèse ont les avantages suivants :

- Méthode simple d'approximation.
- Intégration facile de l'équation d'état.
- On n'a pas besoin d'approximer la fonction coût.
- Obtention de bons résultats même pour de faibles valeurs de m .

Nous avons appliquées notre méthode, dont la convergence est prouvée, sur plusieurs exemples test étudiés par d'autres chercheurs en utilisant différentes méthodes. Nous avons appliqué aussi la méthode (BPFs+AG) dans l'étude d'un problème de commande optimale pratique : problème de réacteur chimique. A partir des résultats obtenus, nous pouvons conclure que les algorithmes proposés donnent de meilleurs ou comparables résultats aux autres.

Les solutions des problèmes de commande optimale dans les chapitres 4 et 5 sont des solutions en boucles ouvertes, mais la solution en boucle fermée est désirable à obtenir pour ses avantages qu'elle peut offrir, par conséquent dans le chapitre 2, nous avons proposé une méthode qui donne la solution optimale en boucle fermée pour les systèmes linéaires à coefficients variables.

Les problèmes de commande optimale qui présentent un caractère fortement non linéaire ou multimodal sont nombreux surtout dans le domaine de la chimie

industrielle, c'est pourquoi nous avons élaboré une autre méthode basée sur les algorithmes génétiques qui permet de trouver l'optimum global.

6.2 Perspectives

Le travail de cette thèse peut être étendu comme suit :

- Au chapitre 2, nous avons proposé une méthode pour trouver la commande optimale en boucle fermée pour le problème de commande optimale des systèmes linéaires à coefficients variables. Celle-ci peut être étendue pour trouver la commande optimale en boucle fermée des problèmes non linéaires avec ou sans contraintes.
- L'algorithme du chapitre 5 peut être généralisé pour manipuler d'autres contraintes autres que les contraintes de saturation sur la commande.
- Les algorithmes élaborés dans le cadre de cette thèse peuvent être exploités pour l'étude de la commande prédictive des systèmes non linéaires.
- Etude de la commande optimale des systèmes à retard soumis à des contraintes.

Bibliographie

- [1] Lee E. B. et L. Markus, *Foundation of Optimal Control Theory*, John Wiley, USA, 1967
- [2] Anderson. B et J. Moore, *Optimal Control Linear Quadratic Method*, Englewood Cliffs, NJ, Prentice Hall, 1990.
- [3] Bryson A. et Y. C. Ho, *Applied Optimal Control*, Hemisphere publishing Corporation, Washington D.C, 1975.
- [4] Sage A. et Chelsea W. *Optimum Systems Control*, Prentice-Hall, NJ: Englewood Cliffs, 1977.
- [5] Kirk D. E., *Optimal Control Theory: An Introduction*, Prentice-Hall, Englewood Cliffs, 1970.
- [6] Dorato P. Chaouki A. et V. Cerone, *Linear Quadratic Control: An Introduction*, Prentice-Hall, 1995.
- [7] Lewis F.L. et V. Syrmos, *Optimal Control*, John Wiley, USA, 1995.
- [8] Athans M., The Status of Optimal Control Theory and Applications for Deterministic Systems, *IEEE Transactions on Automatic Control*, 11, 580-596, 1966
- [9] Bryson A., Optimal Control-1950 to 1985, *IEEE Control Systems Magazine*, 16, 26-33, 1996.
- [10] Sussmann H. J. et J. C. Willems, 300 Years of Optimal Control : From the Brachystochrone to the maximum principle, *IEEE Control Systems Magazine*, 17, 32-44, 1997.
- [11] Bellman R. E. et S. E. Dreyfus, *Applied Dynamic Programming*, Princeton NJ: Princeton University Press, 1962.

- [12] Pontryagin L.S., V. G. Boltyanskii, R. V. Gamkrelidze et E. F. Mischenko, *The Mathematical Theory of Optimal Process*, Wiley, New York, 1962.
- [13] Ahmed N. U., *Element of Finite Dimensional Systems and Control Theory*, Longman Scientific and Technical, UK, 1988.
- [14] Lukes D. L., Optimal Regulation of Nonlinear Dynamical System, *Automatica*, 7, 75-100, 1969.
- [15] Willemstein A. P., Optimal Regulation of Nonlinear Dynamical Systems on a Finite Interval, *SIAM J. Control*, 15, 1050-1069, 1977.
- [16] Garrard W. et J. Jordan, Design of Nonlinear Automatic Flight Control Systems, *Automatica*, 13, 497-505, 1977.
- [17] Nishikawa Y., N. Sannomiya et H. Itakura, A Method for Suboptimal Design of Nonlinear Feedback Systems, *Automatica*, 7, 703-712, 1971.
- [18] Yoshida T. et K. Loparo, Quadratic Regulatory Theory for Analytic Nonlinear System with Additive Controls, *Automatica*, 25-4, 531-544, 1989.
- [19] Kugelmann B. et H. J. Pesch, New General Guidance Method in Constrained Optimal Control, Part 1: Numerical Method, *Journal of Optimization Theory and Applications*, 67, 421-435, 1990.
- [20] Kugelmann B. et H. J. Pesch, New General Guidance Method in Constrained Optimal Control, Part 2: Application to Space Shuttle Guidance, *Journal of Optimization Theory and Applications*, 67, 437-446, 1990.
- [21] Pesch H. J., Real-time Computation of Feedback Controls for Constrained Optimal Control Problems, Part 1: Neighboring Extremals, *Optimal Control Applications and Methods*, 10, 129-145, 1990.
- [22] Roenneke A. et K. Well, Linear Optimal Control for Reentry Flight, *Computationnal Optimal Control*, Ed: R. Bulirsch et D. Kraft, Birkhauser, Germany pp. 339-348, 1994.
- [23] Burghart J. H., A Technique for Suboptimal Feedback Control of Nonlinear Systems, *IEEE Transactions on Automatic Control*, 14, 530-533, 1969.
- [24] Wernli A. et G. Cook, Suboptimal Control for the Nonlinear Quadratic Regulator Problem, *Automatica*, 11, 75-84, 1975.

- [25] Bass R. W. et R.F. Webber, Optimal Nonlinear Feedback Control Derived from Quadratic and Higher Order Performance Criteria, *IEEE Transactions on Automatic Control*, 11, 448-454, 1966.
- [26] Moylan P. J. et B. D. Anderson, Nonlinear Regulator Theory and an Inverse Optimal Control Problem, *IEEE Transactions on Automatic Control*, 18, 460-465, 1973.
- [27] Sandor J. et D. Williamson, Design of Nonlinear Regulator for Linear Plants, *IEEE Transactions on Automatic Control*, 22, 47-50, 1977.
- [28] Bernstein D., Non-quadratic Cost and Nonlinear Feedback Control, *Int. J. of Robust and Nonlinear Control*, 3, 211-229, 1993.
- [29] Nedeljkovic N. B., New Algorithms for Unconstrained Nonlinear Optimal Control Problems, *IEEE Trans. Automat. Cont.*, 26, 868-884, 1981.
- [30] Goh C. J., On the Nonlinear Optimal Regulator Problem, *Automatica*, 29, 751-756, 1993.
- [31] Longmuir A. G. et E. Bohn, Synthesis of Suboptimal Feedback Control Laws, *IEEE Trans. Automat. Cont.*, 12, 755-758, 1967.
- [32] Tabak D., Applications of Mathematical Programming Techniques in Optimal Control: A Survey, *IEEE Trans. Aut. Cont.*, AC-15-12, 688-690, 1970.
- [33] Troch I., F. Breiteneker et M. Graeff, Computing Optimal Controls for Systems with State and Control Constraints, *IFAC Control Applications of Nonlinear Programming and Optimization*, France, 39-44, 1989.
- [34] Teo K., C. Goh et K. Wong, *A Unified Computational Approach to Optimal Control Problems*, Longman Scientific & Technical, England, 1991.
- [35] Polak E., *Computational Method in Optimization*, Academic Press, New York, 1971.
- [36] Goh C. J. et K. L. Teo, Control Parameterization: A Unified Approach to Optimal Control Problems with General Constraints, *Automatica*, **24-1**, 3-18, 1988.
- [37] Sirisena H. R., Computation of Optimal Controls Using a Piecewise Polynomial Parameterization, *IEEE Trans. Aut. Cont.*, **AC-18**, 409-411, 1973.

- [38] Spangelo I., *Trajectory Optimization for Vehicles Using Control Vector Parameterization and Nonlinear Programming*, PhD thesis, 1994.
- [39] Vlassenbroeck J. et R. Van Doreen, A Chebyshev Technique for Solving Nonlinear Optimal Control Problems, *IEEE Trans. Aut. Cont.*, **33**, 333-340, 1988.
- [40] Vlassenbroeck J., A Chebyshev Polynomial Method for Optimal Control With State Constraints, *Automatica*, **24**, 499-506, 1988.
- [41] Frick P. A. et D. J. Stech, Solution of Optimal Control Problems on a Parallel Machine Using the Epsilon Method, *Optimal Control Applications and Methods*, **16**, 1-17, 1995.
- [42] Frick P. A. et D. J. Stech, Epsilon-Ritz Method for Solving Optimal Control Problems: Useful Parallel Solution Method, *Journal of Optimization Theory and Applications*, **79**, 31-58, 1993.
- [43] Sirisena H. R. et F. S. Chou, An Efficient Algorithm for Solving Optimal Control Problems with Linear Terminal Constraint, *IEEE Trans. Aut. Cont.*, **AC-21**, 275-277, 1976.
- [44] Fox L. et I. B. Parker, *Chebyshev Polynomials in Numerical Analysis*, Oxford University Press, England, 1968.
- [45] Bellman R. et R. Kalaba, *Quasi Linearization and Nonlinear Boundary Value Problems*, Elsevier, New York, 1965.
- [46] Vlach J., *Computerized Approximation and Synthesis of Linear Networks*, London, Wiley, 1969.
- [47] Lee J. et Z. Bien, A Computational Method for Infinite Time Optimal Regulation Problems, *IEEE Trans. Automatic. Cont.*, **AC-31**, 1146-1149, 1986.
- [48] Elnagar G., M. Kazemi et M. Razzaghi, The pseudospectral Legendre Method for Discretizing Optimal Control Problem, *IEEE Trans. Automatic Cont.* **AC-40-10**, 1793-1796, 1995.
- [49] Sirisena H. R. et F. S. Cou, State Parameterization Approach to the Solution of Optimal Control Problem, *Optimal Control Applications & Methods*, **2-3**, 289-298, 1981.
- [50] Sirisena H. R et K. S. Tan, Computation of Constrained Optimal Controls

Using Parameterization Techniques, *IEEE Trans. Automatic Cont.* **19-8**, 431-433, 1974.

- [51] Hsieh H. C., Synthesis of adaptive control systems by function space methods, *Advances in control systems*, Vol. 2, 117-208, 1965.
- [52] Neuman C. P. and A. Sen, A suboptimal control algorithm for constrained problems using cubic splines, *Automatica*, **9**, 601-613, 1973.
- [53] Mehar R. and R. Davis, A generalised gradient method for optimal control problems with inequality constraints and singular arcs, *IEEE Trans. Automatic Control*, **AC 17-1**, 69-79, 1973.
- [54] Fegley K., S. Blum, J. Bergholm, A. Calise, J. Marowitz, G. Porcelli and L. Sinha, stochastic and deterministic design and control via linear and quadratic programming, *IEEE Trans. Automatic Control*, **AC 16-6**, 759-766, 1971.
- [55] Bashein G. and M. Enns, Computation of optimal control by a method combining quasi-linearization and quadratic programming, *Int. Journal of Control*, vol.16, 177-187, 1972.
- [56] Ma B. and W. S. Levine, An algorithm for solving control constrained optimal control problem, *Proc. Of 32ed IEEE Conference on Decision and Control (CDC)*, San Antonio, 784-790, 1993.
- [57] Ma B. and W. S. Levine, An algorithm for solving optimal control problem with control and terminal state constraints *Proc. Of 32ed IEEE Conference on Decision and Control (CDC)*, San Antonio, 1374-1380, 1993.
- [58] Kelly H. J., Method of gradients, In *Optimization Techniques* Ed: G. Leitman, Academic Press, New York, 1962.
- [59] Bullock T. and G. Franklin, A second order feedback method for optimal control computations, *IEEE Trans. Automat. Cont.*, 12, 666-673, 1967.
- [60] Bellman R., *Introduction to mathematical theory of control process*, Academic Press, New York, 1971.
- [61] Chen C. F. and C. H. Hsiao, Design of piecewise constant gains for optimal control via Walsh functions, *IEEE Trans. Automat. Cont.*, **20**, 593-603, 1975.

- [62] Chou J. and I. Horng, Application of Chebyshev polynomials to optimal control of time varying linear systems, *Int. Journal of Cont.*, **41**, 135-144, 1985.
- [63] Glad S. T., Robustness of nonlinear state feedback- A survey, *Automatica*, **23**, 425-435, 1987.
- [64] Liu C. and Y. Shih, Analysis and optimal control of time varying systems via Chebyshev polynomials, *Int. Journal of Cont.*, **38**, 1003-1012, 1983.
- [65] Mitter S. K., Successive approximation methods for the solution of optimal control problems, *Automatica*, **3**, 135-149, 1966.
- [66] Paraskevopoulos P. N., Chebyshev series approach to systems identification, analysis and optimal control, *Journal of the Franklin Institute*, **316**, 135-157, 1983.
- [67] Rao V. P. and K. R. Rao, Optimal feedback control via block pulse functions, *IEEE Tran. Automat. Control*, **24**, 372-374, 1979.
- [68] Urabe M., Numerical solution of boundary value problems in chebyshev series- A method of computation and error estimation, *lecture Notes Math*, **109**, 40-86, 1969.
- [69] Wang S. K. and M. L. Nagurka, Designing linear optimal regulator via Chebyshev polynomials, *Proc. Of the American Control Conference*, San Francisco, 2685-2689, 1993.
- [70] Van Doreen R. and J. Vlassenbroeck, A computational method in optimal systems control with various applications, *Third IMA Conference on Control Theory*, Academic Press, London, 407-429, 1981.
- [71] Van Doreen R., A Chabyshev technique applied to a controlled nuclear reactor system, *Optimal Control Applications and Methods*, **10**, 285-291, 1989.
- [72] Cullum J., Discrete approximations to continuous optimal control problems, *SIAM J. Control*, **7-1**, 32-49, 1969.
- [73] Stryk O. and R. Bulirsch, Direct and indirect methods for trajectory optimization, *Annals of Operations Research*, **37**, 357-373, 1992.
- [74] Sakawa Y. and Y. Shindo, Optimal control of container cranes,

Automatica, **18-3**, 257-266, 1982.

- [75] Kraft D., On converting optimal control problems in nonlinear programming problems, *Computational Mathematical Programming*, Vol. F15, Ed Springer, Berlin, 261-280, 1985.
- [76] Schwartz A. L, *Theory and implementation of numerical methods based on Runge-Kutta integration for solving optimal control problem*, Phd Thesis, 1996.
- [77] Nair G. G., Suboptimal control of nonlinear systems, *Automatica*, **14**, 517-519, 1978.
- [78] Betts J., Issues in the direct transcription of optimal control problem to sparse nonlinear programs, *Computational Optimal Control*, 3-17, 1994.
- [79] Yen V. and M. Nagurka, Optimal control of linearly constrained linear systems via state parameterization, *Optimal Control Applications and Methods*, **13**, 155-167, 1992.
- [80] Jacobson D. H. and D. Q. Mayne, *Differential dynamic programming*, Elsevier, New York, 1990.
- [81] Han S. P., Globally convergent method for nonlinear programming, *Journal of Optimization Theory and Applications*, **22**, 297-309, 1977.
- [82] Han S. P., Superlinearly convergent variable metric algorithm for general nonlinear programming problems, *Mathematical Programming*, **11**, 263-283, 1976.
- [83] Lin S. Y., Complete decomposition algorithm for nonconvex separable optimization problems and applications, *Automatica*, **28**, 1249-1254, 1992.
- [84] Razzaghi M. and G. Elnagar, Linear quadratic optimal control problems via shifted Legendre state parameterization, *Int. J. Systems Science*, **25**, 393-399, 1994.
- [85] Yang C. Y. and C. K. Chen, Analysis and optimal control of time varying systems via Fourier series, *Int. J. Systems Science*, **25**, 1663-1678, 1994.
- [86] Elnagar G. and M. Razzaghi, A collocation type method for linear quadratic optimal control problems, *Optimal Control Applications and Methods*, **18**, 227-235, 1997.

- [87] Razzaghi M., Optimal control of linear time varying system via Fourier series, *Journal of Optimization Theory and Applications*, **65**, 375-384, 1990.
- [88] Hussein M. J, *Numerical methods for solving optimal control problems using Chebyshev polynomials*, Phd Thesis, 1999. www.jaist.ac.jp/thesis/is-doctor-1999/paper/jaddu/paper.ps
- [89] Sannuti P., Analysis and synthesis of dynamic system via block pulse functions, *Proc. IEE Control Application*, **124-6**, 569-571, 1977.
- [90] Shien- yu W., On the Block pulse operator method to solving system with time delay, *16th Annual Conference of IEEE-IECON 90*, vol.1, 120-125, 1990.
- [91] Lapin S. and N. D. Egupov, Analysis of continuous time-varying non Linear systems using block pulse expansions, *Int. Journal Systems Sci.*, **23-7**, 1201 – 1207, 1992.
- [92] Lapin S. and N. D. Egupov, Determinant and statistical analysis of linear time-varying systems via block pulse functions, *Int. Journal Systems Sci.*, **23-7**, 1213-1227, 1992.
- [93] Boussiala N. and A. H. Chaabi, Application of genetic algorithm to high dimensional nonlinear optimal control via the block pulse functions, *Wseas Transactions on Systems*, **5-4**, 886-891, 2006.
- [94] Boussiala N. and A. H. Chaabi, Numerical method for solving constrained nonlinear optimal control using the block pulse functions, *Int. Journal of Innovative Computing Information and Control*, **4-7**, 1733-1740, 2008.
- [95] Boussiala N., *Analyse et commande optimale des systèmes Linéaires à coefficients variable, via les BPFs et les séries de Taylor*, Thèse de Magistère, Université de Constantine(Algérie), 1991.
- [96] Kenneth. H., O. G. Anders, and M. E. Marcus, *User's guide for TOMLAB/SNOPT*, <http://tomlab.biz>.
- [97] Gronwall T. H., Note on the derivatives with respect to a parameter of the solutions of a system of differential equations, *The Annals of Mathematics*, **20-4**, 292-296, 1919.
- [98] Hartig F. and F. J. Keil, Large-scale spherical fixed bed reactor: modelling

- and optimization, *Industrial and Engineering Chemistry Research*, **32-3**, 1993.
- [99] Luus R., J. Dittrich and F.J. Keil, Multiplicity of solutions of a bifunctional catalyst blend in a Tubular Reactor, *Can. J. Chem. Eng.*, **70**, 780-785, 1 992.
- [100] Luus R., *Iterative Dynamic Programming*, Chapman and Hall/CRC, London. UK, 2000.
- [101] Storn R. and K. Price, Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, **11-4**, 341-359, 1997.
- [102] Chiou J. P. and F. S. Wang, A hybrid method of differential evolution with application to optimal control problems of a bioprocess system, *In Proceeding 1998 IEEE on evolutionary computation conference*, **1**, 627-632, 1998.
- [103] Luus R., Application of dynamic programming to high dimensional nonlinear optimal control problems, *Int. J. of Control*, **52-1**, 239-250, 1990.
- [104] Jean-Michel R., *Algorithmes génétiques et réseaux de neurons*, Hermès, Paris, 1995.