

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

Université des Frères Mentouri Constantine 1

Faculté des sciences de la Technologie

Département d'électronique

Thèse

Présenté pour obtenir le diplôme de Doctorat 3^o cycle

en Automatique

Option: Automatique et Traitement du signal

**Contribution à l'Implémentation de Nouvelles Approches de
Synchronisation Hybride des Systèmes Chaotiques à Paramètres
Incertains**

Par : DRISS Zakaria

Devant le jury :

| | | | |
|----------------------------|-----------------------------|--------------|----------------------------------------|
| <u>Président :</u> | Hachouf Fella | Prof. | Université Mentouri Constantine |
| <u>Rapporteur :</u> | Mansouri Noura | Prof. | Université Mentouri Constantine |
| <u>Examineurs :</u> | Bennia Abdelhak | Prof. | Université Mentouri Constantine |
| | Boukrouche Abdelhani | Prof | Université de Guelma |
| | Boukabou Abdelkrim | Prof. | Université de Jijel |

Mai 2017

Résumé

La synchronisation des systèmes chaotiques avec des paramètres incertains est un problème attractif, en particulier dans le domaine des communications sécurisées. L'objectif de cette thèse vise principalement à vérifier les performances de certaines méthodes de contrôle utilisées pour la synchronisation des systèmes chaotiques incertains, de reléver les problèmes et d'y apporter des solutions. Le problème des incertitudes sur les paramètres, dans les systèmes de contrôle non linéaires, a été résolu en particulier par l'utilisation des méthodes de contrôle avancées. Cependant, l'effet de la synchronisation et le type de synchronisation hybride sur la méthode de contrôle est encore un sujet de recherche attractif.

La thèse traite de trois points principaux, chacun d'eux relève d'un aspect du problème étudié. Dans le premier point, nous avons examiné les performances de la synchronisation adaptative dans laquelle le contrôle self-tuning et le contrôle actif sont utilisés. De nombreux problèmes ont été rencontrés et l'un d'eux est la mauvaise estimation des paramètres incertains du système. Ce problème se produit lorsque deux systèmes chaotiques sont bien synchronisés mais une fois la synchronisation réalisée, les lois d'adaptation arrêtent d'estimer les valeurs des paramètres du système sans avoir atteint les vraies valeurs. Ce problème a été considéré et nous avons proposé une solution.

Dans le second point, la commande prédictive particulièrement la commande prédictive généralisée (GPC) a été traitée. C'est une méthode de contrôle complexe dans laquelle la précision du modèle joue un rôle essentiel au cours du processus de commande. Deux parties ont été étudiées ; La première partie porte sur la GPC avec contrôleur PID flou dans laquelle nous avons discuté le rôle du modèle linéaire. La deuxième partie étudie les performances de la GPC utilisant un réseau neurones (NNGPC).

Dans le troisième point, nous avons abordé le problème de la synchronisation hybride des systèmes chaotiques incertains dans lequel nous avons étudié la performance des méthodes de contrôle étudiées. En général, l'effet de la synchronisation complète ou la synchronisation hybride des systèmes chaotiques a paramètres incertains sur certaines méthodes de contrôle avancée est partielle.

Table des matières

| | |
|-----------------------------------------------------------------------------------------------------------------|------------|
| Table des matières | i |
| Table des figures | iii |
| Liste des tableaux | vi |
| Nomenclature | vi |
| Introduction générale | 1 |
| 1 Etat de l'art sur la synchronisation des systèmes chaotiques | 5 |
| 1.1 Introduction | 5 |
| 1.2 Systèmes chaotiques et communication | 7 |
| 1.3 Synchronisation à base de méthodes de contrôle | 9 |
| 1.4 Synchronisation des systèmes chaotiques incertains par les méthodes de contrôle | 10 |
| 1.5 Synchronisation hybride des systèmes chaotiques | 10 |
| 2 Contrôle adaptatif et synchronisation des systèmes Chaotiques incertains | 13 |
| 2.1 introduction | 13 |
| 2.2 Les principales bases du contrôle adaptatif | 14 |
| 2.2.1 Model-Reference Adaptive Control(MRAC) | 14 |
| 2.2.2 Contrôleur Self-Tuning(STC) | 15 |
| 2.3 Synchronisation adaptative des systèmes chaotiques incertains | 16 |
| 2.3.1 La synchronisation Q-S | 18 |
| 2.3.2 Synchronisation des systèmes chaotiques incertains et la condition d'indépendance linéaire LI. | 23 |
| 2.3.3 Synchronisation par éliminateur de perturbation (EOP) | 40 |

| | | |
|----------|---------------------------------------------------------------------|------------|
| 2.4 | Conclusion | 54 |
| 3 | GPC et la Synchronisation des Systèmes Chaotiques Incertains | 55 |
| 3.1 | Introduction | 55 |
| 3.2 | Algorithme GPC classique | 56 |
| 3.3 | Contrôleur PID flou Predictif | 60 |
| 3.3.1 | Régulation PID flou prédictive | 61 |
| 3.3.2 | Contrôleur PI+D flou | 65 |
| 3.3.3 | Contrôleur PD+I flou | 72 |
| 3.3.4 | Analyse de stabilité | 76 |
| 3.3.5 | Simulation numérique | 80 |
| 3.4 | Contrôle prédictif generalisé basé sur réseau neuronal | 85 |
| 3.4.1 | Une introduction générale de la NNGPC | 86 |
| 3.4.2 | Simulation numérique | 89 |
| 3.5 | Conclusion | 96 |
| 4 | Synchronisation hybride des systèmes chaotiques incertains | 97 |
| 4.1 | Introduction | 97 |
| 4.2 | La synchronisation hybride et la condition LI | 103 |
| 4.3 | La synchronisation hybride en utilisant la commande NNGPC | 111 |
| 4.4 | Conclusion | 115 |
| | Conclusion générale | 116 |
| | Bibliographie | 118 |

Table des figures

| | | |
|------|---------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | Principe de la méthode de synchronisation de Pecora et Carroll | 6 |
| 2.1 | Principe du Contrôle adaptatif basé sur un modèle de référence. | 15 |
| 2.2 | Principe du Contrôle adaptatif basé sur le self-tuning | 15 |
| 2.3 | Dynamique des erreurs de synchronisation pour les variables observables. . | 23 |
| 2.4 | Les valeurs estimées des paramètres a, c, d, b, r | 24 |
| 2.5 | Les valeurs estimées des paramètres a_1, c_1, b_1, r_1 | 25 |
| 2.6 | La réponse temporelle du premier commutateur($i = 1$). | 30 |
| 2.7 | Dynamique des erreurs de synchronisation du premier commutateur ($i = 1$). . | 31 |
| 2.8 | Les paramètres d'identification du premier commutateur($i = 1$). | 32 |
| 2.9 | La réponse temporelle du second commutateur($i = 2$). | 33 |
| 2.10 | Dynamique d'erreur de synchronisation du second commutateur ($i = 2$). . . | 34 |
| 2.11 | L'identification des paramètres du second commutateur($i = 2$). | 35 |
| 2.12 | Réponse de l'état du système (2.65) et du système (2.66). | 38 |
| 2.13 | Dynamique des erreurs de synchronisation du système (2.65) et le système (2.66). | 39 |
| 2.14 | Variation des paramètres q_1, q_2, q_3 | 40 |
| 2.15 | Variation des paramètres q_1, q_2, q_3 en modifiant la structure des termes de fonction. | 41 |
| 2.16 | Variation des paramètres q_1, q_2, q_3 en ajoutant de signal variant dans le temps $\omega_1 = -x_2$ | 42 |
| 2.17 | Variation des paramètres q_1, q_2, q_3 en ajoutant un signal constant $\omega_4 = 20$. | 43 |
| 2.18 | Schéma de l'EOP et le contrôleur principal, où $R(t) = 0$ est la référence du système d'erreur. | 44 |
| 2.19 | Le système de contrôle proposé. | 48 |
| 2.20 | Le comportement de synchronisation dans chaque cas. | 50 |
| 2.21 | Les actions de EOP et le contrôleur principal. | 51 |

| | | |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.22 | Les erreurs de paramètres à l'aide de (2.89). | 52 |
| 2.23 | L'identification des paramètres. | 53 |
| 3.1 | Structure de base de la GPC. | 56 |
| 3.2 | Principe de la commande prédictive à horizon étendu | 57 |
| 3.3 | Schéma de régulation PID floue prédictive. | 62 |
| 3.4 | Fonctions d'appartenance d'entrée PI. | 68 |
| 3.5 | Fonctions d'appartenance de sortie PI. | 68 |
| 3.6 | Fonctions d'appartenance d'entrée D. | 68 |
| 3.7 | Fonctions d'appartenance de sortie D. | 68 |
| 3.8 | Régions du régulateur PI floue. | 70 |
| 3.9 | Régions du contrôleur D floue. | 70 |
| 3.10 | Fonctions d'appartenance d'entrée PD. | 73 |
| 3.11 | Fonctions d'appartenance de sortie PD. | 73 |
| 3.12 | Fonctions d'appartenance d'entrée I. | 73 |
| 3.13 | Fonctions d'appartenance de sortie PD. | 73 |
| 3.14 | Régions du régulateur PD floue. | 75 |
| 3.15 | Régions du contrôleur I flou. | 75 |
| 3.16 | Prédicteurs des états x_2, y_2 en utilisant le contrôleur PI+D. | 82 |
| 3.17 | Prédicteurs des états x_2, y_2 en utilisant le contrôleur PD+I | 82 |
| 3.18 | Synchronisation des systèmes de Lorenz et les variations des fonctions coûts sans termes de prédiction à l'aide de commande floue PD+I. | 83 |
| 3.19 | Synchronisation des systèmes de Lorenz et les variations des fonctions coûts avec termes de prédiction à l'aide de commande floue PD+I. | 84 |
| 3.20 | Synchronisation des systèmes de Lorenz et les variations des fonctions coûts sans termes de prédiction à l'aide de commande floue PI+D. | 84 |
| 3.21 | Synchronisation des systèmes de Lorenz et les variations des fonctions coûts avec termes de prédiction à l'aide de commande floue PI+D. | 85 |
| 3.22 | La structure générale de la NNGPC. | 86 |
| 3.23 | Schéma de l'apprentissage du réseau neuronal off-line. | 87 |
| 3.24 | One-step prediction en mode série-parallèle. | 88 |
| 3.25 | Réseau de neurones multicouche , avec une structure en temps tardé. | 88 |
| 3.26 | Two-step prediction | 90 |
| 3.27 | La synchronisation, les variations d'erreur, et l'action de commande pour les systèmes Duffing. | 92 |

| | | |
|------|----------------------------------------------------------------------------------------------------------------------------------------|-----|
| 3.28 | La synchronisation et l'action de commande pour les systèmes de Lorenz. | 93 |
| 3.29 | Dynamique des états d'erreur de synchronisation pour les systèmes de Lorenz. | 94 |
| 3.30 | La synchronisation, et l'action de commande pour les systèmes Lu. | 95 |
| 3.31 | Dynamique des états d'erreur de synchronisation pour les systèmes Lu | 96 |
| 4.1 | Synchronisation hybride de deux systèmes de Rikitake identiques. | 101 |
| 4.2 | L'évolution des fonctions d'erreur e_1, e_2, e_3 durant la synchronisation hybride des deux systèmes de Rikitake identiques. | 102 |
| 4.3 | Synchronisation hybride des deux systèmes Lorenz. | 104 |
| 4.4 | L'évolution des fonctions d'erreur e_1, e_2, e_3 de deux systèmes de Lorenz. | 105 |
| 4.5 | Synchronisation hybride de deux systèmes de Lorenz incertains. | 108 |
| 4.6 | L'évolution des fonctions d'erreur e_1, e_2, e_3 de deux systèmes de Lorenz incertains | 109 |
| 4.7 | L'identification des paramètres en utilisant le contrôle active. | 110 |
| 4.8 | Synchronisation hybride de deux systèmes de Lorenz identiques en utilisant EOP. | 111 |
| 4.9 | L'évolution des fonctions d'erreur e_1, e_2, e_3 de deux systèmes de Lorenz identiques en utilisant EOP. | 112 |
| 4.10 | L'identification des paramètres en utilisant EOP. | 113 |
| 4.11 | Synchronisation hybride de deux systèmes de Lorenz identiques utilisant NNGPC. | 114 |
| 4.12 | L'évolution des fonctions d'erreur e_1, e_2, e_3 de deux identiques Lorenz systèmes NGPC. | 114 |

Liste des tableaux

| | | |
|-----|-------------------------------------------------------------------|----|
| 2.1 | Le tableau des états. | 49 |
| 3.1 | Les erreurs de synchronisation et de la fonction de coût. | 85 |

Introduction générale

les applications des systèmes dynamiques non linéaires sont aujourd'hui présentes dans de nombreuses disciplines. Le comportement qualitatif et quantitatif de ce type de systèmes peut être soudainement changé par une légère variation d'un des paramètres et pour certains d'entre eux, la variation d'un des paramètres peut donner naissance à un comportement complexe appelé chaos. Le terme chaos fait référence à des systèmes déterministes, apériodiques, présentant une dépendance aux conditions initiales et aux variations des paramètres.

En raison de ces propriétés, les signaux chaotiques sont utilisés dans diverses applications et dans divers domaines. Un intérêt particulier a été accordé par la communauté scientifique aux thèmes du contrôle et de la synchronisation des systèmes chaotiques ainsi qu'aux applications potentielles pouvant en découler. Cela s'est traduit par un très grand nombre de travaux sur ces sujets.

La synchronisation des systèmes chaotiques a été particulièrement étudiée par de nombreux chercheurs [1–4]. L'idée de la synchronisation a été proposée pour la première fois par Pecora et Carroll [5] en 1990 ; Ces derniers ont été les premiers à montrer que des systèmes chaotiques pouvaient être synchronisés par un couplage simple. Cela a suffi pour susciter beaucoup d'intérêt en raison des importantes applications possibles dans : les systèmes physiques [6], les systèmes écologiques[7], les systèmes chimiques[8], les communications sécurisées [9–11],...etc.

Différents types de schémas de synchronisation sont décrites dans la littérature, on peut citer à titre d'exemple : la synchronisation complète [12], la synchronisation avec décalage [13], la synchronisation de phase[14], la désynchronisation[15], la synchronisation projective [16], la synchronisation projective modifiée [17], la synchronisation hybride[18], etc.

La synchronisation projective (PS) est la plus notable. Elle a été rapportée pour la première fois par Mainieri et Reháček [19] pour être ensuite largement utilisée avec les sys-

tèmes chaotiques. C'est une synchronisation dans laquelle le système maître et le système esclave peuvent être synchronisés avec un facteur d'échelle . La synchronisation complète (CS) et l'anti-synchronisation (AS) sont des cas particuliers de synchronisation projective où le facteur d'échelle $K = 1$ et $K = -1$, respectivement.

Par la suite, la synchronisation projective modifiée (MPS) a été proposée dans [17]. Les systèmes chaotiques sont alors synchronisés à une matrice d'échelle constante. Récemment, Weng et al. [20] ont utilisé une synchronisation projective hybride entre les états de deux systèmes dynamiques. Dans la synchronisation hybride, la synchronisation complète (CS) et l'anti-synchronisation (AS) coexistent dans le système. Cette coexistence des deux types de synchronisation permet de nouvelles applications et l'amélioration de la sécurité dans certains systèmes de communication et de cryptage chaotique

Après les travaux de Pecora et Carroll, beaucoup de chercheurs se sont alors intéressés à la question de savoir s'il était possible de synchroniser des systèmes en utilisant des méthodes de contrôle. Presque toutes les méthodes de contrôle ont été appliquées pour la synchronisation des systèmes chaotiques : la commande adaptative [21–24], le contrôle actif [25], feedback avec temps retard [26], le backstepping [27], la commande par mode glissant [28]. La majorité de ces études concernent des systèmes avec des paramètres connus et constants alors que dans la réalité, les incertitudes sur les paramètres et les perturbations externes sont des éléments qui peuvent détruire la synchronisation, la rendre plus difficile à réaliser. Pour résoudre ce problème, de nombreux auteurs ont utilisé la synchronisation adaptative. En 2011, Li et al. [23] ont réalisé la synchronisation complète et l'anti-synchronisation des systèmes chaotiques avec des paramètres totalement incertains par la commande adaptative. En [24], Yu et al. ont étudié l'anti-synchronisation des systèmes hyper-chaotiques avec des paramètres incertains et une perturbation externe. Jawaadaa et al. [29] ont étudié l'anti-synchronisation des systèmes hyperchaotiques avec des incertitudes et des perturbations externes, en utilisant la méthode de commande par mode glissant.

L'objectif de cette thèse consiste principalement à vérifier les performances de certaines méthodes de contrôle utilisées pour la synchronisation des systèmes chaotiques incertains, de relever les problèmes et d'y apporter des solutions. Cependant, identifier les problèmes posés par les méthodes de contrôle qui sont très nombreuses n'a pas été une tâche facile d'autant plus que très peu d'auteurs insistent sur les problèmes posés par la méthode qu'ils proposent. Par conséquent, nous avons commencé notre travail en mettant l'accent sur les techniques d'adaptation pour faire face aux incertitudes de nombreuses méthodes

de contrôle telles que H_∞ [30] et Backstepping tout en cherchant à identifier les problèmes posés par certaines méthodes de contrôle robustes tels que TS-fuzzy control (Type-1, Type-2) [31] et la méthode de commande par mode glissant. Ce que nous avons constaté est que de nombreuses méthodes avancées atteignent la synchronisation sans problèmes apparents. Cependant, la méthode de contrôle actif, qui est considérée comme la méthode de contrôle la plus simple n'était pas performante pour un cas assez simple de synchronisation deux systèmes chaotiques identiques. Un intérêt particulier a été accordé au problème des conditions d'indépendance linéaires pour lequel nous avons proposé des solutions qui représentent notre première contribution dans cette thèse.

Comme il est sans doute connu, le problème de l'incertitude sur les paramètres peut aussi être résolu en utilisant une méthode de contrôle robuste. De nombreuses méthodes de contrôle robustes ont été utilisées avec succès, On peut citer à titre d'exemple : la commande floue Type-2 ou la commande par mode glissant. Par conséquent, la deuxième contribution de cette thèse réside dans la proposition d'une nouvelle approche de contrôle robuste basée sur l'utilisation d'un modèle de prédiction. Il est évident que la commande prédictive est la méthode la plus proche de nos critères.

Le travail que nous avons réalisé dans le cadre de cette thèse s'inscrit dans ce contexte. Il est organisé de la façon suivante :

Le chapitre 1 est consacré à un état de l'art sur la synchronisation des systèmes chaotiques. Un ensemble de travaux sur la synchronisation des systèmes chaotiques incertains par les méthodes de contrôle et la synchronisation hybride y sont passés en revue.

Dans le chapitre 2, nous avons donné un aperçu sur les défauts de certains schémas adaptatifs. Un intérêt particulier a été accordé au problème des conditions d'indépendance linéaires pour lequel nous avons proposé des solutions.

Le chapitre 3 est consacré à la présentation de certaines techniques de commande prédictive utilisées pour la synchronisation des systèmes chaotiques incertains et où nous proposons deux approches de commande prédictive basée sur le contrôle PID floue et commande prédictive généralisée basée sur le réseau neuronal.

Dans la synchronisation des systèmes chaotiques, le type de synchronisation peut être utilisé pour résoudre certains problèmes. Bien qu'il existe de nombreux types de synchroni-

sation, on s'est limité dans cette thèse à la synchronisation hybride des systèmes chaotiques. Dans le chapitre 4, on a vérifié l'effet de la commutation du type de synchronisation au cas de la synchronisation complète à la synchronisation hybride sur la méthode de contrôle étudiée au chapitre 2 et 3.

Enfin on terminera par une conclusion.

Chapitre 1

Etat de l'art sur la synchronisation des systèmes chaotiques

1.1 Introduction

Jusqu'à la publication des travaux de Pecora et Carroll en 1990 [5], la synchronisation des systèmes chaotiques a été considérée comme un phénomène contre-intuitif. Ce fut une surprise quand ces derniers ont révélé la possibilité de synchroniser deux systèmes chaotiques en couplant certains de leurs variables internes. En 1991, les mêmes auteurs ont étendu leur travail en l'appliquant à la synchronisation de circuits réels. Ils ont montré que la méthode de synchronisation proposée pouvait être utilisée avec efficacité pour synchroniser deux circuits électroniques présentant des comportement chaotiques [32].

La méthode proposée consiste à diviser le système maître en deux sous-systèmes dans lesquels le premier sous-système sert de signal de couplage pour synchroniser le deuxième sous-système avec le système esclave.

Si on considère le système chaotique de dimension n suivant, comme un système maître :

$$\dot{x} = f(x). \tag{1.1}$$

ce système est divisé en deux sous-systèmes, en posant :

$$x = (v, w)$$

où : $v = (x_1, \dots, x_m)$, $w = (x_{m+1}, \dots, x_n)$

et avec :

$$\dot{v} = g(v, w), \quad \text{et} \quad \dot{w} = h(v, w), \quad (1.2)$$

Ou :

$$g = (f_1(x), \dots, f_m(x)), \quad \text{et} \quad h = (f_{m+1}(x), \dots, f_n(x)).$$

Le système esclave w' est pris comme identique au sous-système w et il est soumis au signal de couplage v . Ce qui donne :

$$\dot{v} = g(v, w), \quad \dot{w} = h(v, w), \quad \dot{w}' = h(v, w'). \quad (1.3)$$

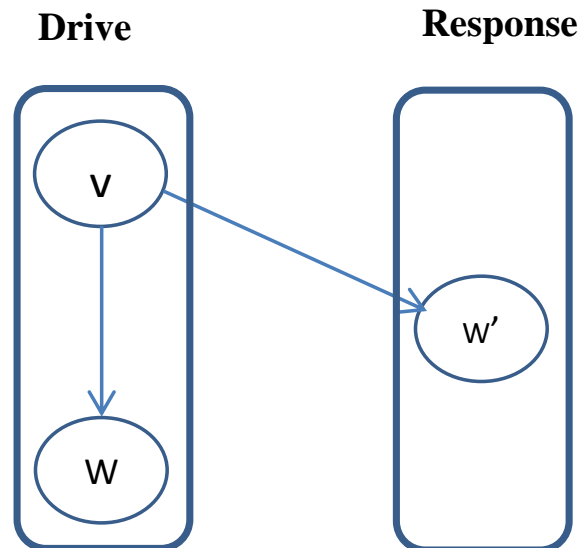


FIGURE 1.1 Principe de la méthode de synchronisation de Pecora et Carroll

Les systèmes sont synchronisés si la différence, $\Delta w = w' - w$ tend vers zéro lorsque le temps tend vers l'infini.

Depuis, les choses ont beaucoup évoluées et aujourd'hui, il existe de nombreux types de synchronisation : la synchronisation complète ou synchronisation identique (CS) [5], la synchronisation de phase (PS) [33] et de retard (LS) [13], la synchronisation généralisée (GS) [10], la synchronisation intermittente de retard (ILS) [34], la synchronisation de phase imparfaite (IPS) [35], la synchronisation projective [19], la synchronisation projective modifiée [17], et la synchronisation hybride [18].

Parmi tous ces types de synchronisation, la synchronisation projective est la plus notable.

C'est une synchronisation dans laquelle le système maître et le système esclave peuvent être synchronisés avec un facteur d'échelle. Elle a été d'abord rapportée par Mainieri et Řeháček [19] qui ont montré que deux systèmes identiques peuvent être synchronisés à un facteur d'échelle K , qui est une transformation constante entre les variables synchronisées des deux systèmes. Cette méthode a suscité par la suite, beaucoup d'attention de la part des chercheurs.

Qu et al.[36] ont étudié les conditions de synchronisation projective. Ghosh [37] a proposé une synchronisation projective à base d'observateur non linéaire dans les systèmes à retard modulés. En 2010, Ghosh et Bhattacharya [38] ont étudié la synchronisation projective d'un système hyper-chaotique de Newton-Leipnik avec des paramètres totalement inconnus.

Par la suite, un nouveau type de synchronisation appelé la synchronisation projective modifiée (MPS) a été proposée dans [17] (MPS). Les systèmes chaotiques sont synchronisés à une matrice d'échelle constante. Ce type de synchronisation permet la coexistence de deux types de synchronisation, Les principales applications sont dans le domaine des communications sécurisées [37]. Récemment, Hu et al. [39] ont utilisé une synchronisation projective hybride entre les états de deux systèmes dynamiques .

La synchronisation complète (CS) et l'anti-synchronisation (AS) sont des cas particuliers de synchronisation projective où le facteur d'échelle sont $K = 1$ et $K = -1$, respectivement. En raison de la relation étroite entre CS et AS et leur importante signification dans la recherche de la synchronisation des systèmes chaotiques, de nombreux articles ont abordé le problème de la coexistence de CS et AS ou ce qu'on appelle la synchronisation hybride pour une classe de systèmes chaotiques. Cette coexistence de CS et AS a été étudiée par Li et al. [40], ils ont montré que tous les états du système esclave étaient en concordance avec les états correspondants du système maître en fonction du type de synchronisation.

1.2 Systèmes chaotiques et communication

L'idée d'utiliser le chaos dans les systèmes de communication a été inspirée de la découverte de Pecora-Carroll. L'émergence de la combinaison de la synchronisation et de l'imprévisibilité a déclenché de nouvelles et intéressantes retombées en particulier dans le domaine des communications sécurisées. Le message entre deux points pouvait ainsi être encodé dans un signal chaotique d'où il était difficile voire impossible d'extraire le message. La conception d'un système de chiffrement par circuit électronique chaotique synchronisé a été réalisée en 1992 par Cuomo et Oppenheim [9].

Oppenheim et al. ont utilisé le concept des systèmes chaotiques synchronisés pour démontrer avec le circuit de Lorenz quelques approches pour sécuriser les communications. L'idée proposée est basée sur le masquage du signal transmis par un signal chaotique et de retirer ce masquage au niveau du récepteur. Le récepteur doit donc générer le signal de masquage et le soustraire au signal reçu. Le concept de masquage a été démontré dans [41, 42]. Dans [43], Murali et Lakshmanan ont utilisé la synchronisation chaotique pour transmettre des signaux dans un modèle intéressant nommé Van der Pol-Duffing oscillateur. Dans [44], Chua et al. ont étudié la synchronisation de circuits de Chua, ces circuits sont bien connus comme étant des circuits électroniques simples pouvant présenter un comportement chaotique.

Dans [45], Gabriel et Cerdeira ont montré comment il était possible d'extraire des messages masqués par un signal chaotique dans un système constitué de deux oscillateurs de Lorenz. Dans [46], Short a étudié le niveau de sécurité dans de tels systèmes, un examen a été effectué sur un ensemble de porteuses chaotiques et des signaux d'informations cachées par la dynamique non linéaire (NLD), ou le chaos. Kocarev et Parlitz [10] ont présenté une approche générale pour construire une grande classe de systèmes dynamiques synchronisés et discuté de son application dans un système de communication où l'information peut être récupérée sans erreurs. Dans un autre article, Kocarev et al. [47] décrit une procédure de synthèse des systèmes synchronisés de grandes dimensions. Outre l'idée d'ajouter un signal de masquage chaotique à l'émission du message, la transmission de signaux numériques par la synchronisation chaotique a également été abordé dans le travail de Parlitz et al. [48]. Inspiré de la méthode en boucle ouverte proposée par Pecora, le récepteur du signal de couplage en général, n'a pas accès aux signaux du système d'entraînement à comparer, et il n'est pas possible de savoir si le sous-système est synchronisé ou non. Pour résoudre ce problème, ils ont proposé d'utiliser un autre sous-système au niveau du récepteur.

Plus tard, Cuomo et al. [49], ont proposé un nouveau système de communication binaire chaotiques. La nouvelle méthode consistait à modéliser un coefficient d'émission avec la forme d'onde porteuse d'informations et l'utiliser pour transmettre le signal chaotique d'entraînement. L'erreur entre le signal de l'émetteur et le signal régénéré au niveau du récepteur a une amplitude qui dépend de la modulation. En utilisant l'erreur de synchronisation, la modulation peut être détectée.

1.3 Synchronisation à base de méthodes de contrôle

La stratégie établie par Pecora, qui est fondée sur la division d'un système en deux sous-systèmes a été vérifiée numériquement sur divers systèmes chaotiques, et a également été réalisée expérimentalement sur certains circuits électriques. Si l'un des sous-systèmes a un exposant de Lyapunov négatif, les trajectoires de deux sous-systèmes peuvent être synchronisées en les couplant avec l'autre sous-système. Cependant cette stratégie n'est pas la seule qu'on peut utiliser pour réaliser la synchronisation de deux systèmes chaotiques. Grebogi et Lai se sont alors intéressés à la question de savoir s'il était possible de synchroniser des systèmes chaotiques sans les diviser en sous-systèmes, en présence de bruit et avec des paramètres incertains en utilisant des méthodes de contrôle. Lai et Grebogi [50] ont proposé une méthode basée sur l'algorithme de Ott et al. [1]. La méthode OGY qui a été proposée au départ pour stabiliser des orbites périodiques instables noyées dans un attracteur chaotique, a été utilisée pour stabiliser la trajectoire d'un système autour de la trajectoire de l'autre système pour réaliser la synchronisation des deux systèmes. La stratégie proposée est basée dans le calcul d'une petite perturbation qui sera appliquée sur un paramètre réglable de l'extérieur. Plus tard, Chen [51] a montré que les systèmes de Lorenz et Duffing-Holmes peuvent être contrôlés sur une orbite désirée via un contrôle en boucle fermée. Un autre travail intéressant dans lequel la théorie du contrôle a été utilisée pour la synchronisation des systèmes chaotiques est apparu dans le travail de Bernardo [52]. Ce dernier a proposé un contrôleur adaptatif basé sur la matrice de Hurwitz dans laquelle l'amplitude de la loi de commande est estimé de manière adaptative.

Par conséquent, la synchronisation des systèmes chaotiques peut être abordée de deux points de vue différents :

- le premier est un problème de suivi, qui consiste à faire suivre les trajectoires du système maître par les trajectoires du système esclave, i.e. $x_{i,S} \longrightarrow x_{i,M}$.
- Le seconde peut être traitée comme un problème de stabilisation, c'est-à-dire que la référence pour la boucle fermée est le point zéro.

Les deux formulations sont similaires en ce sens que l'objectif du contrôle est de maintenir l'écart $\|X_M - X_S\|$ autour de zéro pour toutes les conditions initiales et tous les temps après que le contrôle ait été activé.

En transformant le problème de synchronisation en un problème de contrôle, les méthodes de contrôle peuvent donc être utilisées pour parvenir à la synchronisation. Les méthodes de contrôle qui ont été utilisées pour la synchronisation des systèmes chaotiques sont celles qui utilisent les termes non linéaires du modèle du système et la seconde mé-

thode de Lyapunov ; les plus connues sont les méthodes : contrôle actif [53], Feedback linearization[54], backstepping[55], et high-gain feedback [56].

1.4 Synchronisation des systèmes chaotiques incertains par les méthodes de contrôle

Au début de l'utilisation des méthodes de contrôle, presque tous les articles traitaient le problème sans prêter attention au problème des incertitudes sur les paramètres. Il s'est ainsi avéré qu'en impliquant des variations inconnues sur les paramètres, des techniques plus avancées doivent être utilisées. On peut par exemple, utiliser les méthodes mentionnées ci-dessus en les couplant avec des techniques d'adaptation [57, 58], ou des méthodes de commande robustes en utilisant sliding mode control [59] ou de contrôle TS-floue [31]. Toutes ces méthodes traitent le problème comme un problème de stabilité. Dans le cas d'un problème de suivi, le contrôle par logique floue [60] ou PID flou, sont les mieux indiqués pour atteindre la synchronisation des systèmes chaotiques à paramètres incertains. Les contrôleurs à réseau de neurones [61] sont également utilisés avec succès pour atteindre la synchronisation. Récemment, de nouveaux types de techniques de contrôle ont été émergé en combinant plusieurs méthodes ensemble. Par exemple, l'utilisation de FLC, que ce soit de type 1 ou de type 2, avec sliding mode control [62] ou backstepping [63].

En dehors, des méthodes basées sur la théorie de la stabilité de Lyapunov (première ou deuxième méthode), la synchronisation H_∞ des systèmes chaotiques via le contrôle de rétroaction linéaire et non linéaire ont également été mis en œuvre avec succès [30]. Une autre façon de résoudre le problème est de combiner la méthode H_∞ avec une méthode de commande robuste tel que le contrôle floue T-S [64].

La synchronisation des systèmes chaotiques d'ordre fractionnaire pour des applications théoriques et pratiques [65] a été également abordé. En général, la synchronisation des systèmes chaotiques, hyper chaotiques ou d'ordre fractionnaire concerne principalement le système de chiffrement numérique ou le protocole d'échange de clés dans une communication sécurisée [66] [67].

1.5 Synchronisation hybride des systèmes chaotiques

Sudheer et Sabir [18] ont étudié la synchronisation hybride du système hyperchaotique de Lu en utilisant une méthode de contrôle actif non linéaire. En 2009, chen et al. [68] ont

montré que la synchronisation et l'anti-synchronisation peuvent coexister dans le système chaotique de Chen-Lee en utilisant la méthode directe de Lyapunov et un contrôleur linéaire.

En utilisant le dispositif de commande à rétroaction linéaire, Sun al. [69] ont montré que deux réseaux complexes couplés peuvent réaliser la synchronisation hybride dans des conditions appropriées.

Basé sur la théorie de la passivité, Huang et al. [70] ont présenté la coexistence d'une anti-synchronisation et une synchronisation complète entre deux systèmes hyperchaotiques de Lu identiques avec des conditions initiales différentes.

Sundrapandian et al. [71] ont utilisé la commande par mode glissant (SMC) pour la synchronisation hybride de deux systèmes chaotiques de Qi 3D Four-Wing identiques et de deux systèmes chaotique de Liu 3D Four-Wing identiques. Encore une fois, Sundrapandian et al. [72] ont étudié la synchronisation hybride des systèmes de Wang-Chen identiques, les systèmes de Lorenz hyperchaotique identiques et Wang-Chen hyperchaotique non identiques et les systèmes de Lorenz en utilisant le contrôle non linéaire actif.

Le même auteur a étendu l'étude à de nombreux systèmes chaotiques tels que les systèmes Liu identiques, les systèmes Lü identiques, et les systèmes Liu et Lü non-identiques[73], La synchronisation hybride des systèmes hyperchaotiques de Qi identiques et la synchronisation hybride des systèmes hyperchaotiques de Qi et de Lü[74], les systèmes de Lorenz et Pehlivan identiques [75], les systèmes identiques hyperchaotique de Newton-Leipnik [76], les systèmes identiques hyperchaotique de Rössler, les systèmes identiques hyperchaotique de Lorenz et la synchronisation hybride des systèmes hyperchaotiques de Rössler et Lorenz [77], systèmes Liu identiques, systèmes Chen identiques et systèmes non identiques chaotiques Liu et Chen [78].

Rasappan et al [79] ont étudié la synchronisation hybride des systèmes chaotiques n-scroll Chua et Lur'e via backstepping contrôle. Wang et al. [80] ont étudié la synchronisation complète et la synchronisation hybride de système hyperchaotique de Lorenz avec un retard de temps en utilisant un seul dispositif de commande linéaire.

Zhou et al. ont pour leur part, étudié le problème des incertitudes sur les paramètres de synchronisation hybride de nouveaux systèmes hyperchaotique [81]. Ensuite, Z Zheng et al[82]. ont étudié la synchronisation hybride des deux systèmes chaotiques différents avec retard et des paramètres incertains. SK Agrawal et al.[83] ont proposé la synchronisation hybride entre deux systèmes hyperchaotique d'ordre fractionnaire différents : système Lorenz et système Newton-Leipnik.

Vaidyanathan et al. [84], ont examiné la possibilité de synchroniser 3-cellules de réseau de neurones par une méthode de commande adaptative. Plus tard, ils ont discuté ce type

de synchronisation avec les systèmes biologiques Lotka-Volterra trois espèces en utilisant la même méthode de contrôle [85]. En 2016, Sampath et al.[86] ont proposé une nouvelle méthode de commande par mode glissant pour la synchronisation hybride des systèmes chaotiques identiques avec application au système chaotique Sampath de quatre défilement. En général, le problème de synchronisation hybride a été étudié par de nombreux auteurs en termes de vérification de la capacité de la coexistence de AC et CS dans de nombreux systèmes chaotiques en présence d'incertitudes et de temps retardé.

Chapitre 2

Contrôle adaptatif et synchronisation des systèmes Chaotiques incertains

2.1 introduction

Les incertitudes sur les paramètres peuvent exister dans de nombreux systèmes. Par conséquent, si pour le contrôle de ces systèmes, on utilise un contrôleur ne faisant pas intervenir un mécanisme d'ajustement des paramètres, ce contrôleur peut ne pas être performant. Le contrôle adaptatif est une des méthodes proposées dans la littérature pour palier aux problèmes engendrés par variation des paramètres lors d'une opération de contrôle, dans le cas où le modèle est bien connu. Cette méthode consiste principalement à estimer en ligne les paramètres incertains du système et ensuite utiliser les paramètres estimés dans le calcul de la loi de commande. La commande adaptative peut donc être considérée comme un processus de commande avec un mécanisme d'estimation de paramètres en ligne.

La commande adaptative a été d'abord utilisée par Parlitz [87] comme méthode d'identification pour estimer les paramètres inconnus dans les systèmes de Lorenz incertains. Ensuite, cette méthode a été étendue pour assurer la synchronisation des systèmes chaotiques ou hyperchaotiques. La vaste utilisation de cette méthode a toutefois, fait apparaître de nombreux problèmes. On peut citer à titre d'exemple, les défauts cités dans [88] et qui se résument à cinq types :

- L'obtention de lois d'adaptation des paramètres irréalisables [89]
- La négligence de la condition d'indépendance linéaire [90]
- Les fonctions du contrôleur mal conçues [91]
- la synchronisation paramétrique [92]
- l'impossibilité d'un schéma de synchronisation adaptative pragmatique [93]

Dans la suite de ce chapitre, nous allons donner un aperçu sur les défauts de certains schémas adaptatifs rélevés dans [88]. Puis nous allons nous concentrer sur le problème des conditions d'indépendance linéaires pour lesquels nous allons proposer quelques solutions pour le résoudre

2.2 Les principales bases du contrôle adaptatif

En règle générale, le contrôle adaptatif a comme objectif fondamental le maintien de la performance d'un contrôleur en présence d'incertitude dans la dynamique des systèmes connus. Un système de commande adaptatif est alors constitué de deux boucles ; La première boucle est une boucle de retour d'état entre le système et le contrôleur, et la seconde est une boucle de réglage des paramètres. Il existe deux approches principales pour la construction de ce type de contrôleurs : la méthode de contrôle adaptatif basé sur un modèle de référence et la méthode d'auto-tuning.

2.2.1 Model-Reference Adaptive Control(MRAC)

Le contrôle adaptatif basé sur modèle de référence (MRAC) est un système d'asservissement adaptatif dans lequel la performance souhaitée est exprimée sous forme d'un modèle de référence. Il se compose de trois parties :

- Un modèle de référence pour spécifier la sortie souhaitée pour le système contrôlé.
- Une loi de commande de rétroaction contenant des paramètres réglables.
- Un mécanisme d'adaptation pour l'actualisation des paramètres réglables.

Le rôle du modèle de référence est de fournir la réponse idéale du système à une commande externe. Le contrôleur doit avoir une capacité de suivi parfaite et est généralement paramétriser par un certain nombre de paramètres ajustables. Autrement dit, lorsque les paramètres du système sont connus exactement, le dispositif de commande correspondant doit rendre la sortie du système identique à celle du modèle de référence. Lorsque les paramètres du système ne sont pas connus, le mécanisme d'adaptation ajuste les paramètres du contrôleur pour que le suivi se fasse parfaitement. Dans le contrôle adaptatif avec système de modèle de référence le système MRAC, la loi d'adaptation sert donc à estimer les paramètres de manière à ce que la réponse du système à la loi d'adaptation, devienne la même que celle du modèle de référence.

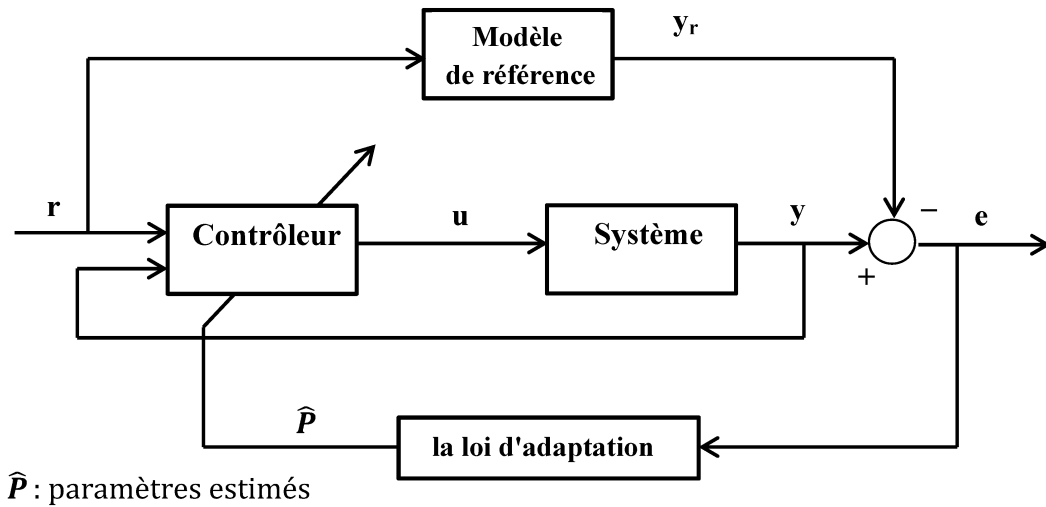


FIGURE 2.1 Principe du Contrôle adaptatif basé sur un modèle de référence.

2.2.2 Contrôleur Self-Tuning(STC)

Un contrôleur self-tuning est un contrôleur qui regroupe un contrôleur avec un estimateur des paramètres récursif en ligne. La Fig. 2.2 illustre la structure schématique d'un tel contrôleur. On remarque qu'il est constitué de deux composantes essentielles :

- Le contrôleur.
- L'estimateur des paramètres.

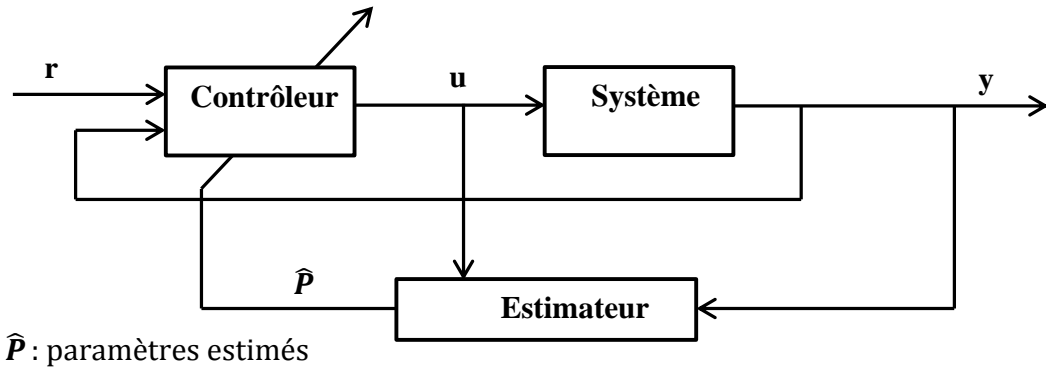


FIGURE 2.2 Principe du Contrôle adaptatif basé sur le self-tuning

L'estimation en ligne des paramètres est un élément clé dans cette méthode. Elle se fait sur la base des signaux des entrées et des sorties du processus et d'état mesurables. Un certain nombre de méthodes d'estimation des paramètres récursives sont employées pour l'auto-tuning. La méthode la plus connue est la méthode d'estimation par moindres carrés.

En général, la conception d'un régulateur adaptatif, tel que MRAC ou STC, implique donc les trois étapes suivantes :

- Choix d'une loi de commande contenant des paramètres variables.
- Choix d'une loi d'adaptation pour ajuster ces paramètres.
- Analyse des propriétés de convergence du contrôle résultant.

Il faut enfin, noter, que le contrôle robuste peut également être utilisé pour faire face à l'incertitude des paramètres. Cependant, dans le cas des constantes ou des paramètres variant lentement, performances de la commande adaptative sont supérieures à celle de la commande robuste. Ceci peut être expliqué par le comportement d'apprentissage des systèmes de contrôle adaptatifs, qui améliore leurs performances plus que l'adaptation continue. De plus, le contrôleur adaptatif nécessite peu ou pas d'information a priori sur les paramètres inconnus ; alors que un contrôleur robuste a besoin au moins des estimations a priori, des limites du paramètre. Enfin, le contrôleur robuste tente simplement de garder des performances constantes.

2.3 Synchronisation adaptative des systèmes chaotiques incertains

Comme mentionné au début, la méthode de contrôle adaptatif a été largement utilisé pour résoudre le problème de la synchronisation des systèmes chaotiques incertains. Dans cette section, nous présentons quelques approches de synchronisation adaptatives et ensuite présenter les défauts et quelques corrections rapportées dans [88].

le lemme 2.3.1 suivant précise les étapes fondamentales pour concevoir un contrôleur pour la synchronisation des systèmes chaotiques incertains.

Soient les systèmes maître et esclave suivants :

$$\begin{cases} \dot{x} = f(x) + F(x)P \\ \dot{y} = g(y) + G(y)Q + U \end{cases} \quad (2.1)$$

où $x \in \mathfrak{R}^n, y \in \mathfrak{R}^n$ sont les vecteurs d'état du maître et de l'esclave respectivement ; $f(x)$ est un vecteur et $F(x)$ est une matrice de $n \times p$ du système maître. $g(x)$ est un vecteur et $G(x)$ une matrice $n \times q$ du système esclave. $P \in \mathfrak{R}^p, Q \in \mathfrak{R}^q$ sont des vecteurs de paramètres incertains.

Lemma 2.3.1 ([88]) Soit $e(t) = y(t) - x(t)$, l'erreur de synchronisation et $\tilde{Q} = Q - \hat{Q}$ et $\tilde{P} = P - \hat{P}$ les erreurs d'estimation des paramètres, alors le système dynamique d'erreur est

obtenu sous la forme de :

$$\begin{cases} \dot{e} &= \dot{y} - \dot{x} = g(y) - f(x) + G(y)Q - F(x)P + U \\ &= G(y)\tilde{Q} - F(x)\tilde{P} + (U + g(y) - f(x) + G(y)\hat{Q} - F(x)\hat{P}), \\ &= G(y)\tilde{Q} - F(x)\tilde{P} + \bar{U} \end{cases} \quad (2.2)$$

où $\bar{U} = (U + g(y) - f(x) + G(y)\hat{Q} - F(x)\hat{P})$.

L'objectif de la synchronisation adaptative consiste à réaliser la synchronisation et l'identification des paramètres inconnus. Par conséquent, on choisit la fonction de Lyapunov suivante :

$$V = V_1(e) + V_2(P, Q), \quad (2.3)$$

où $V_1(e) = e^T e / 2$, and $V_2(P, Q) = (\tilde{P}^T \tilde{P} + \tilde{Q}^T \tilde{Q})$.

La dérivée de V le long de la trajectoire d'erreur est :

$$\begin{aligned} \dot{V} &= e^T (G(y)\tilde{Q} - F(x)\tilde{P} + \bar{U}) + \tilde{Q}^T \dot{\tilde{Q}} + \tilde{P}^T \dot{\tilde{P}} \\ &= e^T \bar{U} + e^T G(y)\tilde{Q} + \tilde{Q}^T \dot{\tilde{Q}} - e^T F(x)\tilde{P} + \tilde{P}^T \dot{\tilde{P}} \end{aligned} \quad (2.4)$$

De plus, si on choisit :

$$\begin{cases} \bar{U} &= -Ke \\ \dot{\tilde{Q}} &= -\dot{\tilde{Q}} = -G^T(y)e \\ \dot{\tilde{P}} &= -\dot{\tilde{P}} = F^T(x)e, \end{cases} \quad (2.5)$$

Alors on obtient $\dot{V} = -Ke^2 \leq 0$, où K représente les gains définis positifs. De plus si $V(0)$ est borné, $V(t)$ est également limité. Par conséquent, les trajectoires des états et les estimations des paramètres des systèmes dynamiques (2.1), sont également limitées.

De (2.2), on déduit que $\dot{e}(t)$ existe et est bornée pour $t \in [0, +\infty)$.

De $\dot{V} = -Ke^2 \leq 0$, nous obtenons

$$\int_0^\infty e^2(\tau) d\tau = -\frac{1}{K} \int_0^\infty e^2 \dot{V} d\tau = \frac{1}{K} (V_0 - V_\infty), \quad (2.6)$$

où $V_0 = V(e(0), \tilde{P}(0), \tilde{Q}(0))$.

Selon le résultat du lemme 2.3.1, nous pouvons conclure que $e(t) \rightarrow 0$ quand $t \rightarrow \infty$, ce qui signifie que la synchronisation sera atteinte. De plus, le résultat obtenu implique que $\dot{\tilde{Q}}(t), \dot{\tilde{P}}(t) \rightarrow 0$ quand $t \rightarrow \infty$.

2.3.1 La synchronisation Q-S

La synchronisation Q-S est une version étendue de la synchronisation projective, caractérisée par des facteurs qui font que les deux systèmes se synchronisent proportionnellement. L'objectif dans l'utilisation de la synchronisation Q-S est l'amélioration de la capacité de communication sécurisée.

Soit le système chaotique de dimension n suivant, considéré comme système maître :

$$\dot{x} = f(x) + F(x)P, \quad (2.7)$$

où $x = (x_1, x_2, \dots, x_n)^T \in \mathfrak{R}^n$ est le vecteur d'état du système, $f \in C(\mathfrak{R}^n, \mathfrak{R}^n)$ est la fonction comprenant les termes non linéaires, $P \in \mathfrak{R}^k$ est le vecteur des paramètres du système et $F \in C(\mathfrak{R}^n, \mathfrak{R}^{n \times k})$.

Le système esclave est défini par :

$$\dot{y} = g(y) + G(y)\Theta + U, \quad (2.8)$$

où $y = (y_1, y_2, \dots, y_n)^T \in \mathfrak{R}^n$ est le vecteur d'état du système, $g \in C(\mathfrak{R}^n, \mathfrak{R}^n)$ est la fonction comprenant les termes non linéaires, $G \in C(\mathfrak{R}^n, \mathfrak{R}^{n \times l})$ et $\Theta \in \mathfrak{R}^l$ est le vecteur des paramètres du système.

Soit $Q(x) = (Q_1(x), Q_2(x), \dots, Q_h(x))^T$ et $S(x) = (S_1(x), S_2(x), \dots, S_h(x))^T$ les variables observables du maître et de l'esclave respectivement, Alors la double fonction $Q - S$ de synchronisation des deux systèmes chaotiques est :

$$\begin{aligned} e(t) &= \alpha(t)Q(x) - \beta(t)S(y) \\ &= (\alpha(t)Q_1(x) - \beta(t)S_1(y) \\ &\quad \alpha(t)Q_2(x) - \beta(t)S_2(y), \dots, \\ &\quad \alpha(t)Q_h(x) - \beta(t)S_h(y))^T, \end{aligned} \quad (2.9)$$

où $\alpha(t) \in C^1(\mathfrak{R}_+, \mathfrak{R}_+)$ ($0 < \alpha(t) \leq N_\alpha$, N_α est une constante positive) et $\beta(t) \in C^1(\mathfrak{R}_+, \mathfrak{R}_+)$ ($0 < \beta(t) \leq N_\beta$, N_β est une constante positives).

Le système d'erreur entre le maître et l'esclave s'écrit alors sous la forme :

$$\begin{aligned}
\dot{e}(t) &= \dot{\alpha}(t).Q(x) + \alpha(t).DQ.(f(x) + F(x)P) \\
&\quad - \dot{\beta}.S(y) - \beta(t)DS \\
&\quad .(g(y) + G(y)\Theta + U) \\
&= h(x, y, t) + \alpha(t).DQ.F(x)P \\
&\quad - \beta(t)DS.(g(y) + G(y)\Theta + U),
\end{aligned} \tag{2.10}$$

où

$$h(x, y, t) = \dot{\alpha}(t)Q(x) + \alpha(t)DQ.f(x) - \dot{\beta}(t)S(y), \tag{2.11}$$

et $DQ(x)$, $DS(y)$ sont les matrices jacobiennes de la fonction vectorielle $Q(x)$ et $S(y)$, respectivement.

$$DQ(x) = \begin{bmatrix} \frac{\partial Q_1(x)}{\partial x_1} & \frac{\partial Q_1(x)}{\partial x_2} & \cdots & \frac{\partial Q_1(x)}{\partial x_n} \\ \frac{\partial Q_2(x)}{\partial x_1} & \frac{\partial Q_2(x)}{\partial x_2} & \cdots & \frac{\partial Q_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Q_h(x)}{\partial x_1} & \frac{\partial Q_h(x)}{\partial x_2} & \cdots & \frac{\partial Q_h(x)}{\partial x_n} \end{bmatrix}, \tag{2.12}$$

$$DS(y) = \begin{bmatrix} \frac{\partial S_1(y)}{\partial y_1} & \frac{\partial S_1(y)}{\partial y_2} & \cdots & \frac{\partial S_1(y)}{\partial y_n} \\ \frac{\partial S_2(y)}{\partial y_1} & \frac{\partial S_2(y)}{\partial y_2} & \cdots & \frac{\partial S_2(y)}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial S_h(y)}{\partial y_1} & \frac{\partial S_h(y)}{\partial y_2} & \cdots & \frac{\partial S_h(y)}{\partial y_n} \end{bmatrix}, \tag{2.13}$$

Le contrôleur peut être réalisé sous la forme :

$$U = -g(y) - G(y)\hat{\Theta} + \frac{DS^{-1}}{\beta(t)}.(H(x, y, t) + \alpha(t).DQ.F(x)\hat{P}), \tag{2.14}$$

où $H \in C(\mathfrak{R}^n \times \mathfrak{R}^n \times \mathfrak{R} \rightarrow \mathfrak{R}^h)$, $\hat{\Theta}$ et \hat{P} sont le vecteur estimé de paramètres inconnus.

Les lois d'adaptation des paramètres estimés sont données par

$$\begin{cases} \dot{\hat{P}} &= (\alpha(t).DQ.F(x))^T . e - \tilde{P} \\ \dot{\hat{\Theta}} &= -(\beta(t).DS.G(y))^T . e - \tilde{\Theta}, \end{cases} \tag{2.15}$$

où $\tilde{P} = \hat{P} - P$ et $\tilde{\Theta} = \hat{\Theta} - \Theta$.

Dans (2.15), les lois de contrôle et d'adaptation font intervenir les vraies valeurs des paramètres alors que l'hypothèse posée indique que ces paramètres ne sont pas connus avec exactitude, ceci rend le processus infaisable dans la pratique (lois d'adaptation des paramètres infaisable). Le lemme 2.3.1 peut être utilisé comme outil de référence pour vérifier la faisabilité d'un système de synchronisation additif proposé. Par conséquent, le résultat de la synchronisation de QS doit être corrigée de la façon suivante

$$U = -g(y) - G(y)\hat{\Theta} + \frac{DS^{-1}}{\beta(t)} \cdot (H(x, y, t) + \alpha(t) \cdot DQ \cdot F(x)\hat{P}), \quad (2.16)$$

où

$$\begin{aligned} \dot{\hat{P}} &= (\alpha(t) \cdot DQ \cdot F(x))^T \cdot e \\ \dot{\hat{\Theta}} &= -(\beta(t) \cdot DS \cdot G(y))^T \cdot e \end{aligned}$$

Le système d'erreur s'écrit alors comme suit :

$$\dot{e} = h(x, y, t) - H(x, y, t) - \alpha(t) \cdot DQ \cdot F(x) \cdot \tilde{P} + \beta(t) DS \cdot G(y) \cdot \tilde{\Theta}. \quad (2.17)$$

Pour illustrer le principe de cette méthode, nous considérons la synchronisation de deux systèmes hyperchaotique de Chen ; le système maître est défini comme suit :

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) + x_4 \\ \dot{x}_2 = dx_1 - x_1x_3 + cx_2, \\ \dot{x}_3 = x_1x_2 - bx_3 \\ \dot{x}_4 = x_2x_3 + rx_4 \end{cases} \quad (2.18)$$

et le système esclave par :

$$\begin{cases} \dot{y}_1 = a_1(y_2 - y_1) + y_4 + u_1 \\ \dot{y}_2 = -y_1y_3 + c_1y_2 + u_2, \\ \dot{y}_3 = y_1y_2 - b_1y_3 + u_3 \\ \dot{y}_4 = y_1y_3 + r_1y_4 + u_4 \end{cases} \quad (2.19)$$

En supposant que les variables observables des systèmes maître et esclave sont respectivement :

$$\alpha(t)Q(x) = (1 - e^{-t}) \begin{bmatrix} x_1 - x_2 \\ x_2 + x_3, \\ x_3 + x_4 \end{bmatrix}$$

(2.20)

et

$$\beta(t)S(y) = (1 + e^{-t}) \begin{bmatrix} \frac{2}{3}y_1 - \frac{1}{3}y_2 + \frac{1}{3}y_4 \\ \frac{1}{3}y_1 - \frac{2}{3}y_3 + \frac{1}{3}y_4 \\ \frac{1}{3}y_2 - \frac{1}{3}y_3 + \frac{1}{3}y_4 \end{bmatrix}, \quad (2.21)$$

Alors,

$$DQ(x) = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad (2.22)$$

$$DS(y) = \frac{1}{3} \begin{bmatrix} 2 & -1 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}, \quad (2.23)$$

et

$$DS^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 2 \\ -1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix}. \quad (2.24)$$

Le contrôleur $U = (u_1, u_2, u_3, u_4)^T$ est obtenu en résolvant :

$$U = -g(y) - G(y)\hat{\Theta} + \frac{DS^{-1}}{1 + e^{-t}} \cdot (e^{-t}Q(x) + (1 - e^{-t}) \cdot DQ \cdot (f(x) + F(x)\hat{P}) + e^{-t}S(y) + e), \quad (2.25)$$

Soit :

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \frac{1}{1 + e^{-t}} \times \begin{pmatrix} (1 - e^{-t})(\lambda_1 - \lambda_2) + e^{-t}\varepsilon_1 + e_1 \\ (1 - e^{-t})(-\lambda_2 + \lambda_3 + 2\lambda_4) + e^{-t}(-\varepsilon_2 + 2\varepsilon_3) - e_2 + 2e_3 \\ (1 - e^{-t})(-\lambda_1 + 3\lambda_2 + \lambda_3 - \lambda_4) + e^{-t}(-\varepsilon_1 + 2\varepsilon_2 - \varepsilon_3) - e_1 + 2e_2 - e_3 \\ (1 - e^{-t})(\lambda_1 - 2\lambda_2 + \lambda_3 + 2\lambda_4) + e^{-t}(\varepsilon_1 - \varepsilon_2 + 2\varepsilon_3) + e_1 - 2e_2 + 2e_3 \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{pmatrix} \quad (2.26)$$

Avec :

$$\begin{aligned}\lambda_1 &= \hat{a}(x_2 - x_1) + x_4, \lambda_2 = \hat{d}x_1 - x_1x_3 + \hat{c}x_2, \\ \lambda_3 &= x_1x_2 - \hat{b}x_3, \lambda_4 = x_2x_3 + \hat{r}x_4, \\ \mu_1 &= \hat{a}_1(y_2 - y_1) + y_4, \mu_2 = -y_1y_3 + \hat{c}_1y_2, \\ \mu_3 &= y_1y_2 - \hat{b}_1x_3, \mu_4 = y_1y_3 + \hat{r}_1y_4, \\ \varepsilon_1 &= x_1 - x_2 + \frac{2}{3}y_1 - \frac{1}{3}y_2 + \frac{1}{3}y_4, \\ \varepsilon_2 &= x_2 + x_3 + \frac{1}{3}y_1 - \frac{2}{3}y_3 + \frac{1}{3}y_4, \\ \varepsilon_3 &= x_3 + x_4 + \frac{1}{3}y_2 + \frac{1}{3}y_3 + \frac{1}{3}y_4.\end{aligned}$$

Les lois d'adaptation sont les suivantes :

$$\begin{pmatrix} \dot{\hat{a}} \\ \dot{\hat{c}} \\ \dot{\hat{d}} \\ \dot{\hat{b}} \\ \dot{\hat{r}} \end{pmatrix} = (1 - e^{-t}) \times \begin{pmatrix} e_1(x_2 - x_1) \\ (-e_1 + e_2)x_1 \\ (-e_1 + e_2)x_2 \\ (-e_2 - e_3)x_3 \\ e_3x_4 \end{pmatrix} - \begin{pmatrix} \tilde{a} \\ \tilde{c} \\ \tilde{d} \\ \tilde{b} \\ \tilde{r} \end{pmatrix} \quad (2.27)$$

et

$$\begin{pmatrix} \dot{\hat{a}}_1 \\ \dot{\hat{c}}_1 \\ \dot{\hat{b}}_1 \\ \dot{\hat{r}}_1 \end{pmatrix} = \left(-\frac{1 + e^{-t}}{3}\right) \times \begin{pmatrix} (y_2 - y_1)(2e_1 + e_2) \\ y_2(-e_1 + e_3) \\ y_3(-e_2 - e_3) \\ y_4(e_1 + e_2 + e_3) \end{pmatrix} - \begin{pmatrix} \tilde{a}_1 \\ \tilde{c}_1 \\ \tilde{d}_1 \\ \tilde{b}_1 \end{pmatrix} \quad (2.28)$$

Pour faire la simulation de cet exemple ; on choisit pour les systèmes maître et esclave les valeurs des paramètres comme suit :

$$a = 35, b = 3, c = 12, d = 7, r = 0.5,$$

$$a_1 = 36, b_1 = 3, c_1 = 20, r_1 = 1,$$

Et comme valeurs initiales des états :

$$x_1(0) = 5, x_2(0) = 8, x_3(0) = 0.1, x_4(0) = 0.3,$$

$$y_1(0) = 3, y_2(0) = 4, y_3(0) = 5, y_4(0) = 5.$$

Les erreurs de synchronisation Q-S des trois fonctions sont représentées sur la Fig. 2.3, alors que la dynamique de l'estimation des paramètres est représentée par Fig. 2.4 et Fig. 2.5

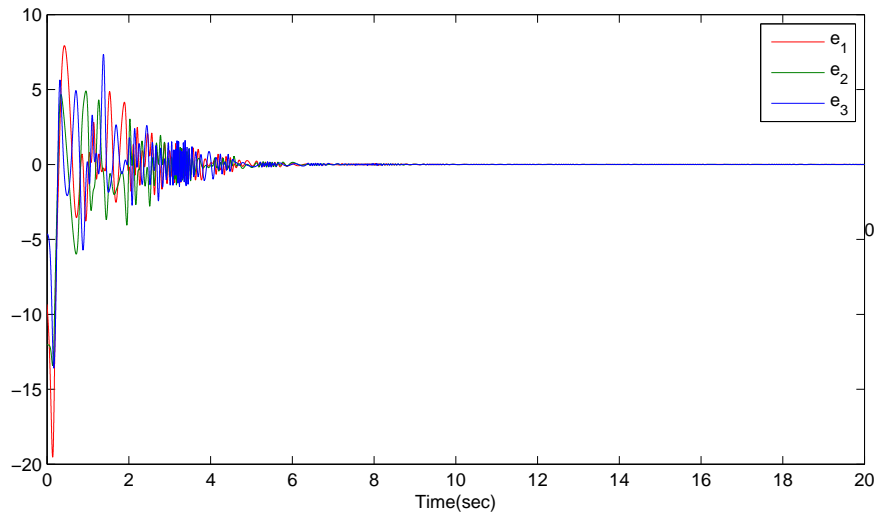


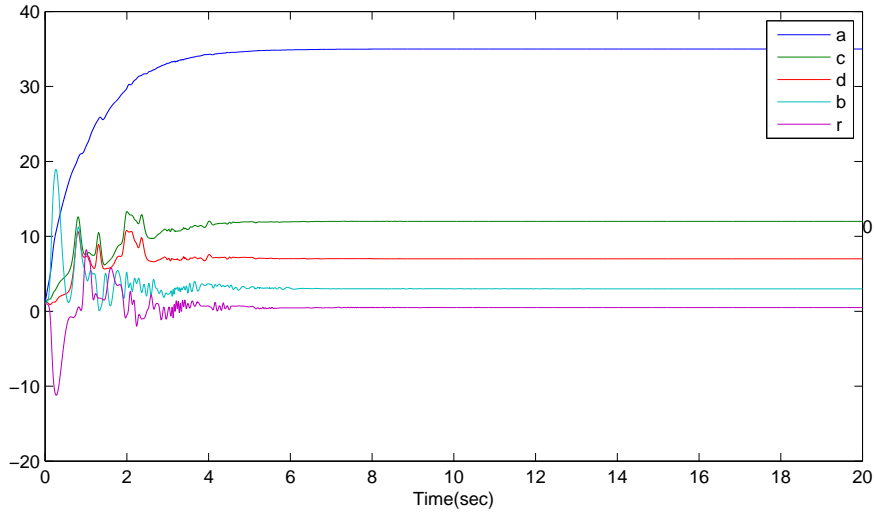
FIGURE 2.3 Dynamique des erreurs de synchronisation pour les variables observables.

Le même problème apparaît dans le cas du troisième défaut (fonctions de contrôleur mal conçus) dans lequel les vraies valeurs sont impliquées dans la mise à jour des lois de contrôle. Ce problème est résolu en utilisant le même Lemme. Le reste des régimes de défauts est bien expliqué dans les versions corrigées proposées dans [88].

Dans la section suivante nous allons nous concentrer uniquement sur le défaut de la négligence de la condition d'indépendance linéaire (LI).

2.3.2 Synchronisation des systèmes chaotiques incertains et la condition d'indépendance linéaire LI.

Pour rappel, la synchronisation multi commutation peut être considérée comme un type de synchronisation dans lequel la synchronisation se produit entre les deux systèmes, en combinant leurs états différemment. La différence entre la synchronisation complète (CS) et la synchronisation multi-commutation est que la CS est un cas particulier de la synchronisation multi-commutation, ce qui représente le premier commutateur. Le lemme 2.3.1 est utilisée pour concevoir le contrôleur non linéaire actif pour la synchronisation multi-commutation.

FIGURE 2.4 Les valeurs estimées des paramètres a, c, d, b, r .

Afin de présenter les bases de l'algorithme de synchronisation multi-commutation, on considère le système oscillant linéaire maître suivant :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -a_1 x_1, \end{cases} \quad (2.29)$$

où $\sqrt{a_1}$ est la fréquence de l'oscillation.

Et le système esclave :

$$\begin{cases} \dot{y}_1 = y_2 + u_{i1}(t) \\ \dot{y}_2 = -a_2 y_1 + u_{i2}(t), \end{cases} \quad (2.30)$$

où $\sqrt{a_2}$ est la fréquence de l'oscillation, u_{i1} , u_{i2} , sont les commandes et $i = 1, 2$ les indices du commutateur.

Dans le premier commutateur $i = 1$, les erreurs sont définies à partir des systèmes (2.29) et (2.30)

$$e_{11} = y_1 - x_1,$$

$$e_{12} = y_2 - x_2.$$

La dynamique du système d'erreur est déterminée comme suit :

$$\begin{cases} \dot{e}_{11} = y_2 - x_2 + u_{i1}(t) \\ \dot{e}_{12} = -a_2 y_1 + a_1 x_1 + u_{i2}(t), \end{cases} \quad (2.31)$$

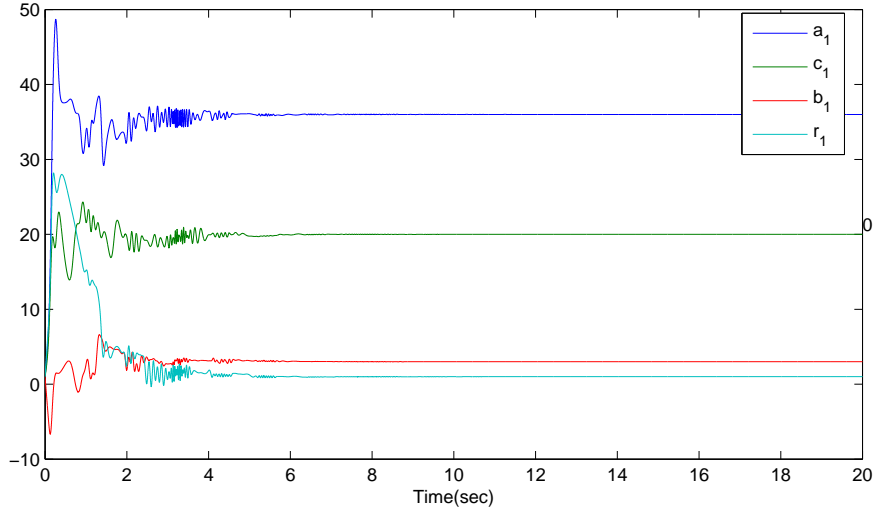


FIGURE 2.5 Les valeurs estimées des paramètres a_1, c_1, b_1, r_1 .

L'objectif est de concevoir un contrôleur et des mécanismes d'estimation des paramètres tels que la trajectoire du système (2.30) peut asymptotiquement approcher celle du système (2.29), i.e.

$$(2.32) \quad \lim_{t \rightarrow \infty} \|e(t)\| \rightarrow 0$$

Les commandes $u_{11}(t)$ et $u_{12}(t)$ sont les suivantes :

$$(2.33) \quad \begin{cases} u_{11}(t) = -e_{11} - e_{12}, \\ u_{12}(t) = -e_{12} + \hat{a}_2 y_1 - \hat{a}_1 x_1, \end{cases}$$

où \hat{a}_1, \hat{a}_2 sont les valeurs estimées des paramètres a_1, a_2 .

Et la dynamique du système d'erreur est donnée par :

$$(2.34) \quad \begin{cases} \dot{e}_{11}(t) = -e_{11}, \\ \dot{e}_{12}(t) = -e_{12} - \tilde{a}_2 y_1 + \tilde{a}_1 x_1, \end{cases}$$

En posant $\tilde{a}_1 = a_1 - \hat{a}_1$ et $\tilde{a}_2 = a_2 - \hat{a}_2$, et en choisissant la fonction de Lyapounov sui-

vante

$$(2.35) \quad V(e_{11}^2, e_{12}^2, a_1^2, a_2^2) = \frac{1}{2}(e_{11}^2 + e_{12}^2 + \tilde{a}_1^2 + \tilde{a}_2^2).$$

La dérivée de V le long de la trajectoire de la dynamique de la fonction d'erreur conduit à :

$$(2.36) \quad \begin{aligned} \dot{V} &= e_{11}\dot{e}_{11} + e_{12}\dot{e}_{12} + \tilde{a}_1\dot{\tilde{a}}_1 + \tilde{a}_2\dot{\tilde{a}}_2 \\ &= e_{11}\dot{e}_{11} + e_{12}\dot{e}_{12} - \tilde{a}_1\dot{\hat{a}}_1 - \tilde{a}_2\dot{\hat{a}}_2. \end{aligned}$$

Les paramètres estimés sont obtenus avec les lois d'adaptation suivantes :

$$(2.37) \quad \begin{aligned} \dot{\hat{a}}_1 &= x_2 e_{12}, \\ \dot{\hat{a}}_2 &= -y_1 e_{12}. \end{aligned}$$

A partir de (2.33) et (2.37), on obtient :

$$\dot{V} = -e_{11}^2 - e_{12}^2 \leq 0.$$

De plus, V est une fonction définie positive et \dot{V} est une fonction définie négative, i.e. (2.32) est satisfaite ce qui implique que les états du système esclave (2.30) et du système maître (2.31) sont synchronisés.

Dans le second commutateur, $i = 2$, les erreurs des systèmes (2.30) et (2.31) sont définies comme suit :

$$\begin{aligned} e_{21} &= y_1 - x_2, \\ e_{22} &= y_2 - x_2, \end{aligned}$$

et leur dynamique est déterminée par :

$$(2.38) \quad \begin{cases} \dot{e}_{21} = y_2 + a_1 x_1 + u_{21}(t) \\ \dot{e}_{22} = -a_2 y_1 - x_2 + u_{22}(t), \end{cases}$$

Avec la même méthode présentée précédemment, on construit les contrôleurs et les lois d'adaptation suivantes :

$$(2.39) \quad \begin{cases} u_{21} = -e_{21} - y_2 - \hat{a}_1 x_1, \\ u_{22} = -e_{22} + \hat{a}_2 y_1 + x_2. \end{cases}$$

$$\begin{cases} \dot{\hat{a}}_1 &= x_1 e_{21}, \\ \dot{\hat{a}}_2 &= -y_1 e_{22}. \end{cases} \quad (2.40)$$

En utilisant (2.39), la dynamique du système d'erreur s'écrit

$$\begin{cases} \dot{e}_{21} &= -e_{21} + \tilde{a}_1 x_1, \\ \dot{e}_{22} &= -e_{22} - \tilde{a}_2 y_1. \end{cases} \quad (2.41)$$

En Considérant la fonction de Lyapunov suivante :

$$V(e_{11}^2, e_{12}^2, a_1^2, a_2^2) = \frac{1}{2}(e_{11}^2 + e_{12}^2 + \tilde{a}_1^2 + \tilde{a}_2^2), \quad (2.42)$$

la dérivation de V conduit à

$$\begin{aligned} \dot{V} &= e_{21} \dot{e}_{21} + e_{22} \dot{e}_{22} + \tilde{a}_1 \dot{\tilde{a}}_1 + \tilde{a}_2 \dot{\tilde{a}}_2 \\ &= e_{21} \dot{e}_{21} + e_{22} \dot{e}_{22} - \tilde{a}_1 \dot{\tilde{a}}_1 - \tilde{a}_2 \dot{\tilde{a}}_2. \\ &= -e_{21}^2 - e_{22}^2 \leq 0. \end{aligned} \quad (2.43)$$

Par conséquent, l'esclave et le maître sont globalement synchronisés.

Synchronisation multi commutation du système de Lorenz

Pour illustrer cette méthode, nous considérons deux systèmes de Lorenz incertains. Le système maître est défini par :

$$\begin{cases} \dot{x}_1 &= a_1(x_2 - x_1), \\ \dot{x}_2 &= c_1 x_1 - x_1 x_3 - x_2, \\ \dot{x}_3 &= x_1 x_2 - b_1 x_3, \end{cases} \quad (2.44)$$

et le système esclave, par :

$$\begin{cases} \dot{y}_1 &= a_2(y_2 - y_1) + u_{i1}, \\ \dot{y}_2 &= c_2 y_1 - y_1 y_3 - y_2 + u_{i2}, \\ \dot{y}_3 &= y_1 y_2 - b_2 y_3 + u_{i3}, \end{cases} \quad (2.45)$$

où u_{i1}, u_{i2}, u_{i3} sont les commandess, $i = 1, 2, \dots$ représente la commutation.

Le premier commutateur, $i = 1$, représente la synchronisation complète dans laquelle les

signaux d'erreur sont choisis tels que :

$$e_{11} = y_1 - x_1, e_{12} = y_2 - x_2, e_{13} = y_3 - x_3.$$

La dynamique du système d'erreur est déterminée comme suit :

$$\begin{cases} \dot{e}_{11} = a_2(y_2 - y_1) - a_1(x_2 - x_1) + u_{11}, \\ \dot{e}_{12} = -y_1y_3 + c_2y_1 - y_2 + x_1x_3 - c_1x_1 + x_2 + u_{12}, \\ \dot{e}_{13} = y_1y_2 - b_2y_3 - x_1x_2 + b_1x_3 + u_{13}, \end{cases} \quad (2.46)$$

Les commandes peuvent être définies comme suit :

$$\begin{cases} u_{11} = -e_{11} - \hat{a}_2(y_2 - y_1) + \hat{a}_1(x_2 - x_1), \\ u_{12} = -e_{12} + y_1y_3 - \hat{c}_2y_1 + y_2 - x_1x_3 + \hat{c}_1x_1 - x_2, \\ u_{13} = -e_{13} - y_1y_2 + \hat{b}_2y_3 + x_1x_2 - \hat{b}_1x_3, \end{cases} \quad (2.47)$$

où $\hat{a}_1, \hat{a}_2, \hat{b}_1, \hat{b}_2, \hat{c}_1, \hat{c}_2$ sont les valeurs estimées des paramètres $a_1, a_2, b_1, b_2, c_1, c_2$. Si on pose $\tilde{a}_1 = a_1 - \hat{a}_1, \tilde{a}_2 = a_2 - \hat{a}_2, \tilde{b}_1 = b_1 - \hat{b}_1, \tilde{b}_2 = b_2 - \hat{b}_2, \tilde{c}_1 = c_1 - \hat{c}_1, \tilde{c}_2 = c_2 - \hat{c}_2$, alors la dynamique du système d'erreur est décrite comme :

$$\begin{cases} \dot{e}_{11} = -e_{11} + \tilde{a}_2(y_2 - y_1) - \tilde{a}_1(x_2 - x_1), \\ \dot{e}_{12} = -e_{12} + \tilde{c}_2y_1 - \tilde{c}_1x_1, \\ \dot{e}_{13} = -e_{13} - \tilde{b}_2y_3 + \tilde{b}_1x_3, \end{cases} \quad (2.48)$$

les lois d'adaptation sont choisies selon le modèle suivant :

$$\begin{cases} \dot{\hat{a}}_1 = -e_{11}(x_2 - x_1), \\ \dot{\hat{b}}_1 = e_{13}x_3, \\ \dot{\hat{c}}_1 = -e_{12}x_1, \\ \dot{\hat{a}}_2 = e_{11}(y_2 - y_1), \\ \dot{\hat{b}}_2 = -e_{13}y_3, \\ \dot{\hat{c}}_2 = e_{12}y_1. \end{cases} \quad (2.49)$$

Pour la fonction Lyapunov suivante

$$V = \frac{1}{2}(e^T e + \tilde{a}_1^2 + \tilde{a}_2^2 + \tilde{b}_1^2 + \tilde{b}_2^2 + \tilde{c}_1^2 + \tilde{c}_2^2), \quad (2.50)$$

la dérivation de V le long des trajectoires de la dynamique d'erreur conduit à

$$\begin{aligned}\dot{V} &= e_{11}\dot{e}_{11} + e_{12}\dot{e}_{12} + e_{13}\dot{e}_{13} - \tilde{a}_1\dot{\hat{a}}_1 - \tilde{b}_1\dot{\hat{b}}_1 - \tilde{c}_1\dot{\hat{c}}_1 - \tilde{a}_2\dot{\hat{a}}_2 - \tilde{b}_2\dot{\hat{b}}_2 - \tilde{c}_2\dot{\hat{c}}_2 \\ &= -e_{11}^2 - e_{12}^2 - e_{13}^2 < 0.\end{aligned}\quad (2.51)$$

Puisque V est une fonction définie positive et \dot{V} est une fonction définie négative, i.e. (2.32) est satisfaite. En d'autres termes, les états du système esclave vont asymptotiquement suivre les états du système maître.

Pour la simulation, on utilise la méthode d'Euler avec un temps d'échantillonnage $T = 0.001$ et

$$a_1 = 11, b_1 = 8/3 + 0.4, c_1 = 26$$

$$a_2 = 9, b_2 = 8/3 + 0.2, c_2 = 27.$$

Les valeurs initiales des états sont $x_1(0) = 2, x_2(0) = 10, x_3(0) = -6$ et $y_1(0) = -2, y_2(0) = 5, y_3(0) = 1$. Et ceux des paramètres sont $\hat{a}_1(0) = 2, \hat{b}_1(0) = 10, \hat{c}_1(0) = -6, \hat{a}_2(0) = -2, \hat{b}_2(0) = 5, \hat{c}_2(0) = 1$. Les résultats des simulations sont représentés par Fig. 2.6 Fig. 2.7. La variation des paramètres estimés pour le premier commutateur sont représentés dans la figure Fig. 2.8.

Dans le second commutateur, lorsque $i = 2$, l'erreur entre les systèmes sont définis comme par :

$$e_{21} = y_1 - x_2, e_{22} = y_2 - x_3, e_{23} = y_3 - x_1,$$

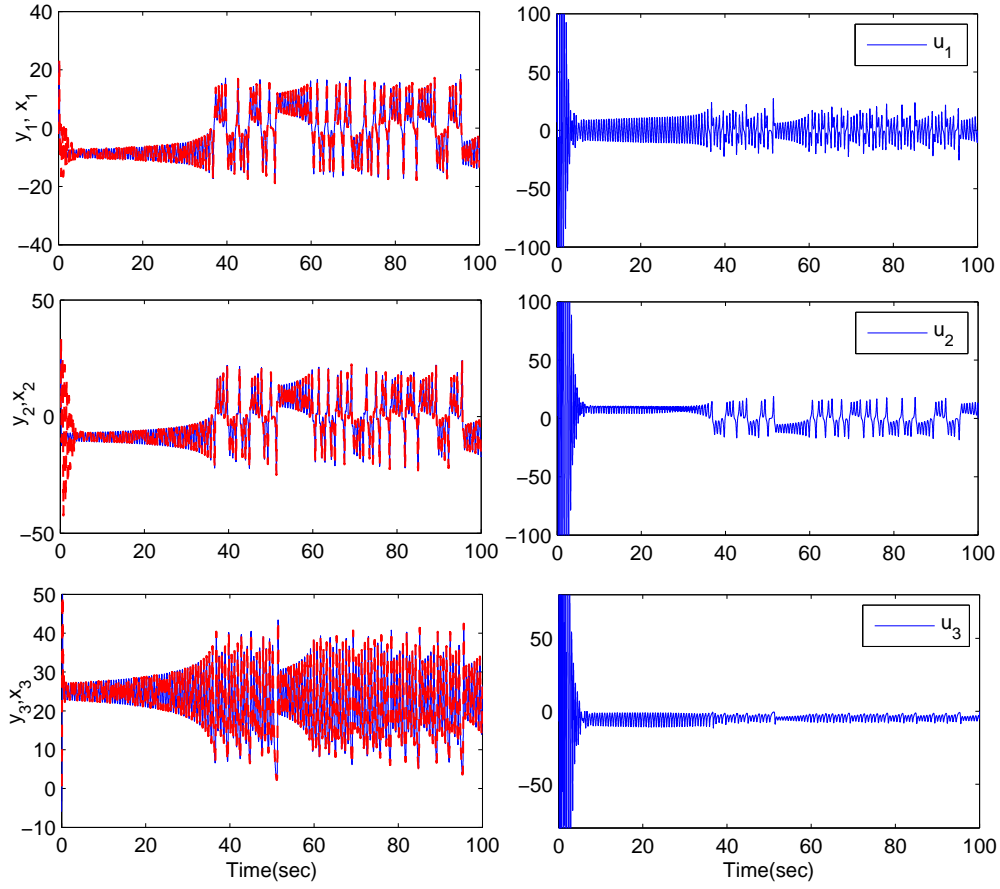
de sorte que la dynamique du système d'erreur est déterminé suit :

$$\begin{cases} \dot{e}_{21} &= a_2(y_2 - y_1) + x_1x_3 - c_1x_1 + x_2 + u_{21}, \\ \dot{e}_{22} &= -y_1y_3 + c_2y_1 - y_2 - x_1x_2 + b_1x_3 + u_{22}, \\ \dot{e}_{23} &= y_1y_2 - b_2y_3 - a_1(x_2 - x_1) + u_{23}, \end{cases}\quad (2.52)$$

Nous pouvons définir les fonctions de contrôle comme :

$$\begin{cases} u_{21} &= -e_{21} - \hat{a}_2(y_2 - y_1) - x_1x_3 + \hat{c}_1x_1 - x_2, \\ u_{22} &= -e_{22} + y_1y_3 - \hat{c}_2y_1 + y_2 + x_1x_2 - \hat{b}_1x_3, \\ u_{23} &= -e_{23} - y_1y_2 + \hat{b}_2y_3 + \hat{a}_1(x_2 - x_1), \end{cases}\quad (2.53)$$

où $\hat{a}_1, \hat{a}_2, \hat{b}_1, \hat{b}_2, \hat{c}_1, \hat{c}_2$ sont les valeurs estimées des paramètres $a_1, a_2, b_1, b_2, c_1, c_2$. Si on pose : $\tilde{a}_1 = a_1 - \hat{a}_1, \tilde{a}_2 = a_2 - \hat{a}_2, \tilde{b}_1 = b_1 - \hat{b}_1, \tilde{b}_2 = b_2 - \hat{b}_2, \tilde{c}_1 = c_1 - \hat{c}_1, \tilde{c}_2 = c_2 - \hat{c}_2$, alors

FIGURE 2.6 La réponse temporelle du premier commutateur ($i = 1$).

la dynamique du système d'erreur est décrit comme :

$$\begin{cases} e_{21} = -e_{21} + \tilde{a}_2(y_2 - y_1) - \tilde{c}_1 x_1, \\ e_{22} = -e_{22} + \tilde{c}_2 y_1 + \tilde{b}_1 x_3, \\ e_{23} = -e_{23} - \tilde{b}_2 y_3 - \tilde{a}_1(x_2 - x_1). \end{cases} \quad (2.54)$$

les lois d'adaptation sont comme suit :

$$\begin{cases} \dot{\hat{a}}_1 = -e_{23}(x_2 - x_1), \\ \dot{\hat{b}}_1 = e_{22}x_3, \\ \dot{\hat{c}}_1 = -e_{21}x_1, \\ \dot{\hat{a}}_2 = e_{21}(y_2 - y_1), \\ \dot{\hat{b}}_2 = -e_{23}y_3, \\ \dot{\hat{c}}_2 = e_{22}y_1. \end{cases} \quad (2.55)$$

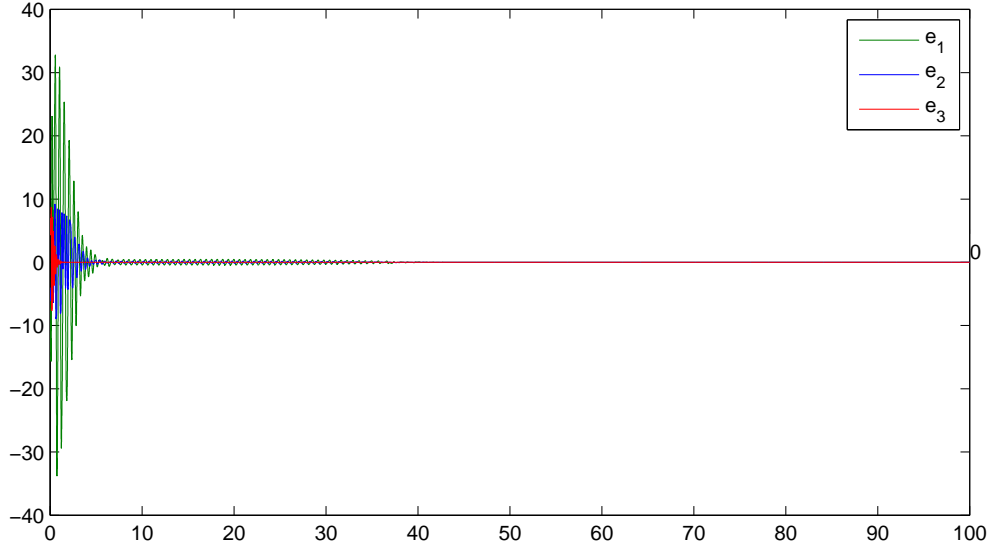


FIGURE 2.7 Dynamique des erreurs de synchronisation du premier commutateur ($i = 1$).

Et pour la fonction de Lyapunov :

$$V = \frac{1}{2}(e^T e + \tilde{a}_1^2 + \tilde{a}_2^2 + \tilde{b}_1^2 + \tilde{b}_2^2 + \tilde{c}_1^2 + \tilde{c}_2^2). \quad (2.56)$$

La dérivée de V le long de la trajectoire de la dynamique du système d'erreur

$$\begin{aligned} \dot{V} &= e_{21}\dot{e}_{21} + e_{22}\dot{e}_{22} + e_{23}\dot{e}_{23} - \tilde{a}_1\dot{\hat{a}}_1 - \tilde{b}_1\dot{\hat{b}}_1 - \tilde{c}_1\dot{\hat{c}}_1 - \tilde{a}_2\dot{\hat{a}}_2 - \tilde{b}_2\dot{\hat{b}}_2 - \tilde{c}_2\dot{\hat{c}}_2 \\ &= -e_{21}^2 - e_{22}^2 - e_{23}^2 < 0. \end{aligned} \quad (2.57)$$

V est une fonction définie positive et \dot{V} est une fonction définie négative, i.e. (2.32), donc le système esclave et le système maître sont synchronisés.

Fig. 2.9 indique l'état de la synchronisation entre les deux systèmes ainsi que la variation du signal de contrôle. Alors que Fig. 2.10 montre l'évolution des erreurs. La variation des paramètres estimés pour $i = 2$ sont représentés dans la figure Fig. 2.11.

Des résultats ci-dessus, nous pouvons observer que la synchronisation dans le premier et le second commutateur a été obtenue en utilisant l'équation des entrées de commande et les lois d'adaptation. Cependant, les paramètres estimés du premier commutateur se stabilisent tous à des valeurs différentes et ne convergent pas vers les valeurs réelles. D'autre part, dans le second commutateur, les paramètres estimés se stabilisent tous sur les valeurs réelles. La

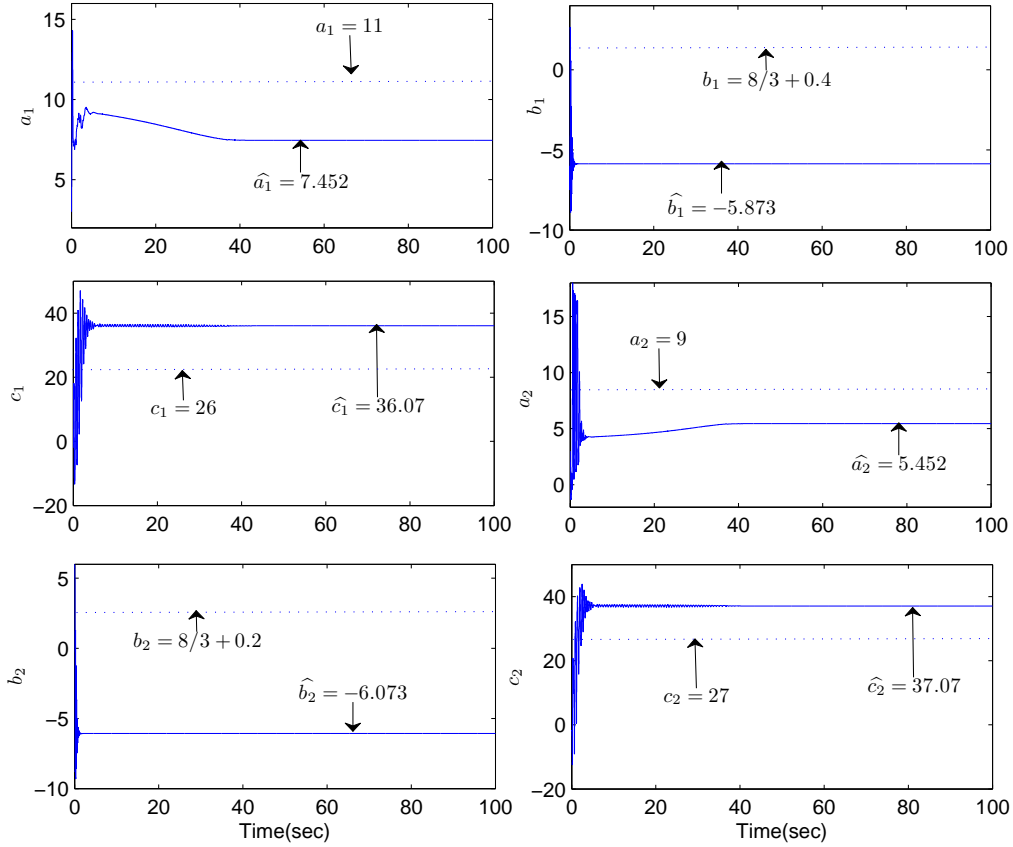


FIGURE 2.8 Les paramètres d'identification du premier commutateur ($i = 1$).

question est de savoir quelle est la raison qui est derrière le comportement de la méthode de commande adaptative proposée dans le cas du premier commutateur, bien que le lemme (2.3.1) soit satisfait. En d'autres termes, pourquoi dans le second commutateur l'estimation a été performante alors que dans le premier cas non. Afin de voir où se situe le problème, nous allons faire une comparaison entre le premier et le second commutateur quand $t \rightarrow \infty$.

A partir des résultats de simulation, on relève les conditions suivantes :

Pour $i = 1$

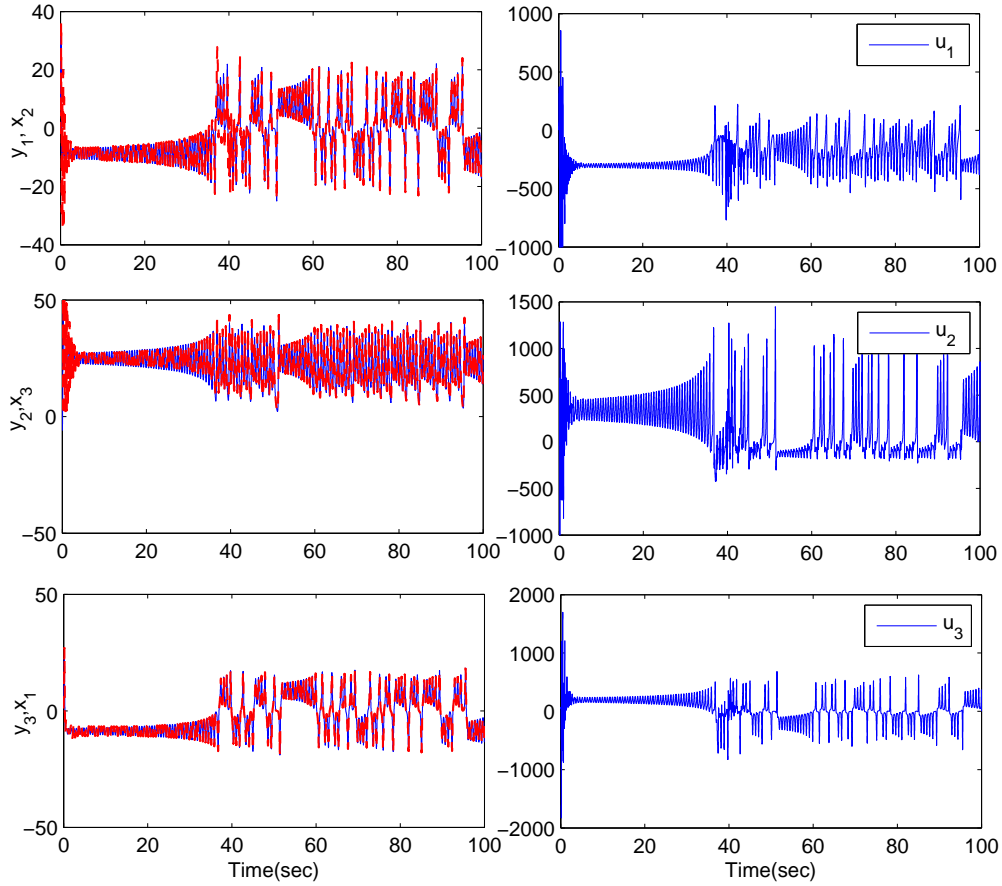
$$e_{11} \rightarrow 0, e_{12} \rightarrow 0, e_{13} \rightarrow 0$$

$$a_1 - \hat{a}_1 \neq 0, a_2 - \hat{a}_2 \neq 0, c_1 - \hat{c}_1 \neq 0, c_2 - \hat{c}_2 \neq 0, b_1 - \hat{b}_1 \neq 0, b_2 - \hat{b}_2 \neq 0$$

$$x_1 = y_1|_{t \rightarrow \infty}, x_2 = y_2|_{t \rightarrow \infty}, x_3 = y_3|_{t \rightarrow \infty}$$

$$x_2 - x_1 = y_2 - y_1|_{t \rightarrow \infty}$$

Pour $i = 2$

FIGURE 2.9 La réponse temporelle du second commutateur($i = 2$).

$$e_{21} \rightarrow 0, e_{22} \rightarrow 0, e_{23} \rightarrow 0$$

$$a_1 - \hat{a}_1 = 0, a_2 - \hat{a}_2 = 0, c_1 - \hat{c}_1 = 0, c_2 - \hat{c}_2 = 0, b_1 - \hat{b}_1 = 0, b_2 - \hat{b}_2 = 0$$

$$x_1 = y_3|_{t \rightarrow \infty}, x_2 = y_1|_{t \rightarrow \infty}, x_3 = y_2|_{t \rightarrow \infty}$$

En insérant les conditions ci-dessus dans les équations (2.46) et (2.52), on obtient :

$$\begin{cases} (a_2 - \hat{a}_2)(y_2 - y_1) - (a_1 - \hat{a}_1)(x_2 - x_1) = 0, \\ (c_2 - \hat{c}_2)y_1 - (c_1 - \hat{c}_1)x_1 = 0, \\ -(b_2 - \hat{b}_2)y_3 + (b_1 - \hat{b}_1)x_3 = 0, \end{cases} \quad (2.58)$$

$$\begin{cases} (a_2 - \hat{a}_2)(y_2 - y_1) - (c_1 - \hat{c}_1)x_1 = 0, \\ (c_2 - \hat{c}_2)y_1 + (b_1 - \hat{b}_1)x_3 = 0, \\ -(b_2 - \hat{b}_2)y_3 - (a_1 - \hat{a}_1)(x_2 - x_1) = 0. \end{cases} \quad (2.59)$$

Ensuite, on compare les différences entre les équations (4.28) et (2.59) lorsque la syn-

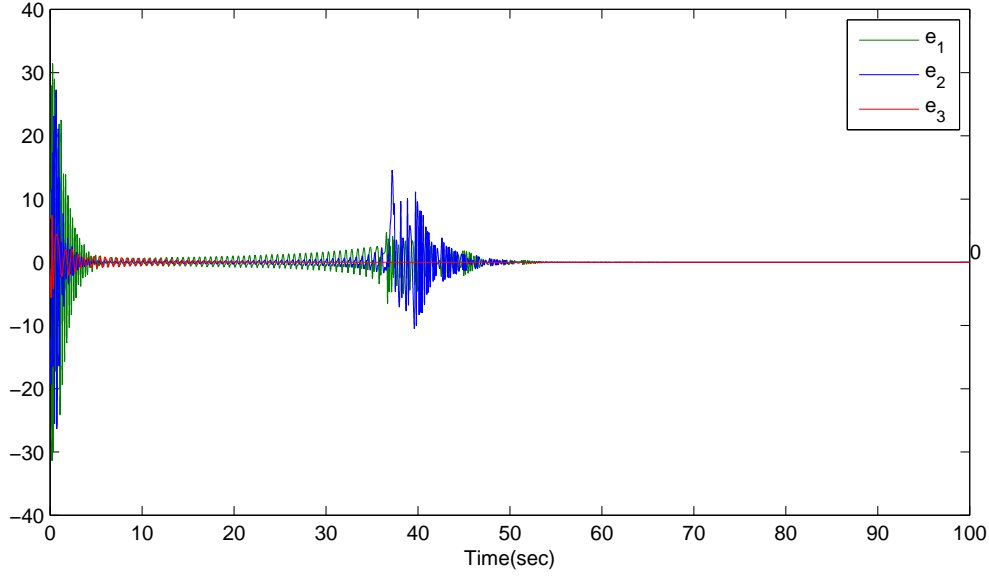


FIGURE 2.10 Dynamique d'erreur de synchronisation du second commutateur ($i = 2$).

chronisation est réalisée

Pour $i = 1$

$$\begin{cases} (a_2 - \hat{a}_2)(y_2 - y_1) - (a_1 - \hat{a}_1)(x_2 - x_1) = (x_2 - x_1)((a_2 - \hat{a}_2) - (a_1 - \hat{a}_1))|_{t \rightarrow \infty}, \\ (c_2 - \hat{c}_2)y_1 - (c_1 - \hat{c}_1)x_1 = x_1((c_2 - \hat{c}_2) - (c_1 - \hat{c}_1))|_{t \rightarrow \infty}, \\ -(b_2 - \hat{b}_2)y_3 + (b_1 - \hat{b}_1)x_3 = x_3((b_1 - \hat{b}_1) - (b_2 - \hat{b}_2))|_{t \rightarrow \infty}, \end{cases} \quad (2.60)$$

Pour $i = 2$

$$\begin{cases} (a_2 - \hat{a}_2)(y_2 - y_1) - (c_1 - \hat{c}_1)x_1 = (a_2 - \hat{a}_2)(x_3 - x_2) - (c_1 - \hat{c}_1)x_1|_{t \rightarrow \infty}, \\ (c_2 - \hat{c}_2)y_1 + (b_1 - \hat{b}_1)x_3 = (c_2 - \hat{c}_2)x_2 + (b_1 - \hat{b}_1)x_3|_{t \rightarrow \infty}, \\ -(b_2 - \hat{b}_2)y_3 - (a_1 - \hat{a}_1)(x_2 - x_1) = -(b_2 - \hat{b}_2)x_1 - (a_1 - \hat{a}_1)(x_2 - x_1)|_{t \rightarrow \infty}. \end{cases} \quad (2.61)$$

Nous pouvons remarquer que la seule différence entre les deux, c'est les termes non linéaires qui sont associés aux paramètres. Si les termes sont linéairement dépendants avec d'autres termes, par exemple, $y_2 - y_1$ avec $x_2 - x_1$ et y_1 avec x_1 et y_3 avec x_3 , les paramètres estimés ne peuvent pas stabiliser vers les vraies valeurs. En revanche, pour $i = 2$, nous remarquons que les termes non linéaires des paramètres sont linéairement indépendants : $(x_3 - x_2)$ avec

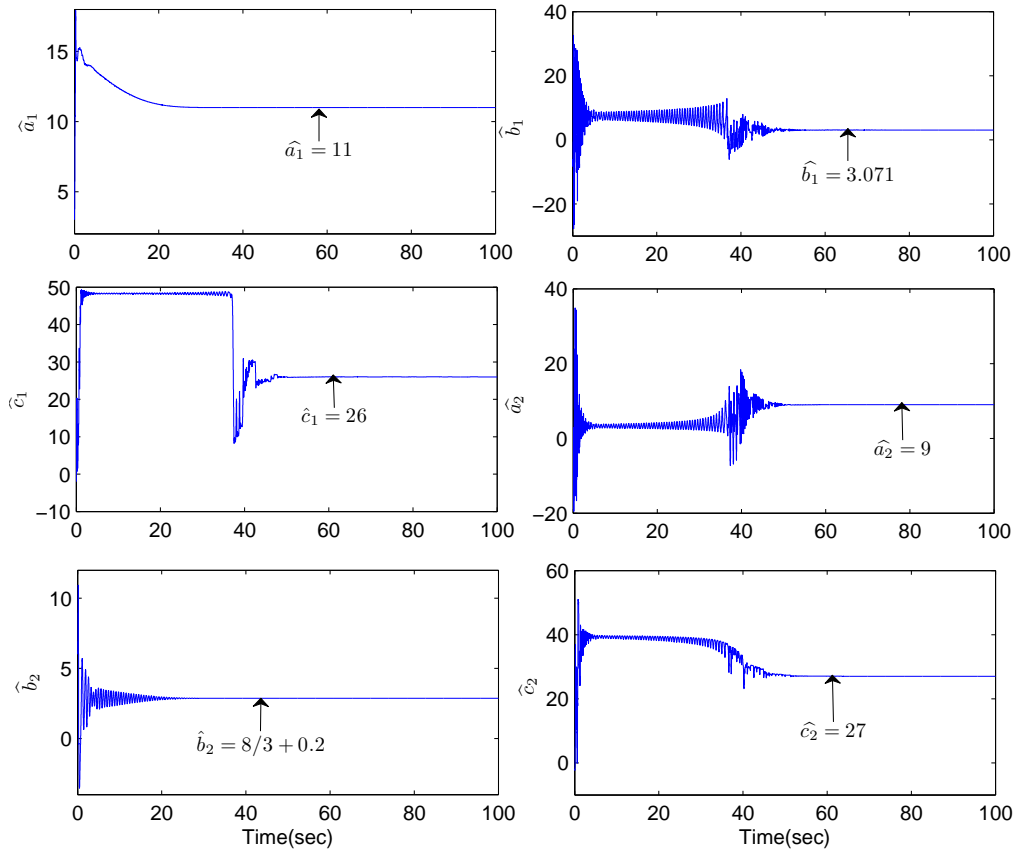


FIGURE 2.11 L'identification des paramètres du second commutateur ($i = 2$).

x_1, x_2 avec x_3 , $(x_2 - x_1)$ avec x_1 . Par conséquent, la raison de l'échec de l'estimation des paramètres du premier commutateur est la dépendance linéaire des termes. En d'autres termes, la synchronisation peut être considérée comme un obstacle pour l'identification des paramètres.

Selon [88], trois cas ont été signalés lorsque la condition de LI n'est pas satisfaite. Nous pouvons les résumer en deux cas : la synchronisation interne et la synchronisation externe. La synchronisation externe est le cas dans lequel la synchronisation entre le maître et l'esclave conduit à la dépendance linéaire des termes $F_i^T(x(t)) \rightarrow G_i^T(y(t))$ lorsque $t \rightarrow \infty$. Par conséquent, l'identification des paramètres inconnus P_i and Q_i ne sera pas obtenue.

Pour la synchronisation interne, il y a deux cas :

1. Les termes $F_i^T(x(t))$ et $G_i^T(y(t))$ sont linéairement dépendants parce que les termes convergent vers zéro en très peu de temps ce qui conduit à une mauvaise identification des paramètres P_i ou Q_i .

2. la synchronisation interne se produit lorsque certains états dans le système maître ou esclave atteignent la synchronisation entre eux, i.e. $x_i(t) \rightarrow x_{j(j \neq i)}$ ou $y_i(t) \rightarrow y_{j(j \neq i)}$ $t \rightarrow \infty$ ce qui implique que $F_i^T(x(t)) \rightarrow F_j^T(x(t))$ ou $G_i^T(y(t)) \rightarrow G_j^T(y(t))$. Si cela se produit dans un court laps de temps, l'identification des paramètres inconnus P_i et P_j ou Q_i et Q_j ne sera probablement pas obtenue.

Afin de remédier au problème de synchronisation interne, nous proposons de réaliser la synchronisation en utilisant des techniques de couplage adaptatifs.

Considérons un système chaotique de dimension n , décrit par

$$\dot{x} = F(x, p), \quad (2.62)$$

où $x = (x_1, x_2, \dots, x_n)^T \in \mathfrak{R}$, $F(x, p) = (F_1(x, p), F_2(x, p), \dots, F_n(x, p))^T$, et

$$F_i(x, p) = c_i + \sum_{j=1}^m p_{ij} f_{ij}(x), i = 1, 2, \dots, n. \quad (2.63)$$

$c_i(x)$ et $f_{ij}(x)$ sont supposés être des fonctions à valeurs réelles, et $p = p_{ij} \in U \cup \mathfrak{R}^n$ sont les paramètres à identifier.

Le système de réponse pour la donnée bornée conduite de signal $x(t)$ est donné par :

$$\begin{aligned} \dot{y} &= F(y, q) + \varepsilon \cdot e, \\ \dot{\varepsilon}_i &= -r_i \varepsilon_i^2, \\ \dot{q}_{ij} &= -\delta_{ij} \varepsilon_i f_{ij}(y), \\ i &= 1, 2, \dots, n, j = 1, 2, \dots, m, \end{aligned} \quad (2.64)$$

où le couplage $\varepsilon \cdot e$ est sous la forme de $(\varepsilon_1 \cdot e_1, \varepsilon_2 \cdot e_2, \dots, \varepsilon_n \cdot e_n)^T$, $e_i = (y_i - x_i)$, $q = q_{ij}$, et r_i et δ_{ij} sont des constantes positives arbitrairement choisies.

Pour la synchronisation interne, nous considérons le système de Lorenz suivant en tant que maître :

$$\begin{cases} \dot{x}_1 = p_1(x_2 - x_1) + p_2(x_4 - x_1) \\ \dot{x}_2 = cx_1 - x_2 - x_1x_3, \\ \dot{x}_3 = -bx_3 + x_1x_2 \\ \dot{x}_4 = -(x_4 - x_1) + p_3(x_2 - x_1) \end{cases} \quad (2.65)$$

où $p_1 = 10, b = 8/3, c = 28$ et $p_2 = 6, p_3 = 10$.

la synchronisation complète entre le système (2.65) et son système de réaction peut être

numériquement réalisé aussi longtemps que le système esclave est réalisé sous la forme

$$\begin{cases} \dot{y}_1 = q_1(y_2 - y_1) + q_2(y_4 - y_1) + \varepsilon_1(y_1 - x_1) \\ \dot{y}_2 = cy_1 - y_2 - y_1y_3 + \varepsilon_2(y_2 - x_2), \\ \dot{y}_3 = -by_3 + y_1y_2 + \varepsilon_3(y_3 - x_3) \\ \dot{y}_4 = -(y_4 - y_1) + q_3(y_2 - y_1) + \varepsilon_4(y_4 - x_4) \end{cases} \quad (2.66)$$

dans lequel les lois d'adaptation des paramètres sont pris telles que :

$$\begin{cases} \dot{q}_1 = -\delta_1(y_1 - x_1)(y_2 - y_1) \\ \dot{q}_2 = -\delta_2(y_1 - x_1)(y_4 - y_1), \\ \dot{q}_3 = -\delta_3(y_4 - x_4)(y_2 - y_1) \end{cases} \quad (2.67)$$

et les forces de couplage adaptatifs sont considérés comme suit :

$$\begin{cases} \dot{\varepsilon}_1 = -r_1(y_1 - x_1) \\ \dot{\varepsilon}_2 = -r_2(y_2 - x_1), \\ \dot{\varepsilon}_3 = -r_3(y_3 - x_3) \\ \dot{\varepsilon}_4 = -r_4(y_4 - x_4) \end{cases} \quad (2.68)$$

Pour la simulation, on utilise la méthode d'Euler avec $T = 0.01$. Les valeurs initiales du système maître et esclave respectivement sont $x_1(0) = 1, x_2(0) = 1, x_3(0) = 1, x_4 = 1$ et $y_1(0) = 6, y_2(0) = 15, y_3(0) = 10, y_4(0) = 2$.

Fig. 2.12 illustre la synchronisation entre les deux systèmes et Fig. 2.13 montre la variation de l'erreur. Les valeurs initiales des paramètres estimés sont $\hat{q}_1(0) = 0, \hat{q}_2(0) = 10, \hat{q}_3(0) = 0$. $r_j = 15, \delta_j = 2$ and $\varepsilon_j^0 = 1, (j = 1, 2, 3, 4)$. La variation des paramètres estimés sont représentés par Fig. 2.14.

Sur Fig. 2.14, on remarque que les paramètres p_1 et p_3 sont identifiés avec succès mais pas p_2 . En fait, à partir de la Fig. 2.12, nous pouvons observer que $x_4(t) \rightarrow x_1(t)$ quand $t \rightarrow \infty$. Par conséquent, la synchronisation interne entre l'état x_4 et l'état x_1 conduit à $(x_4 - x_1) \rightarrow 0$, ce qui signifie que l'élément de fonction $F_1(x) = x_4 - x_1$ est linéairement dépendant d'autres fonctions.

Une des méthodes pour résoudre ce problème est de changer la structure des termes de fonction liés à p_2 comme $(x_4 - x_1 + x_2)$. Ce qui fait que les lois d'adaptation du paramètre p_2 deviennent

$$\dot{q}_2 = -\delta_2(y_1 - x_1)(y_4 - y_1 + y_2).$$

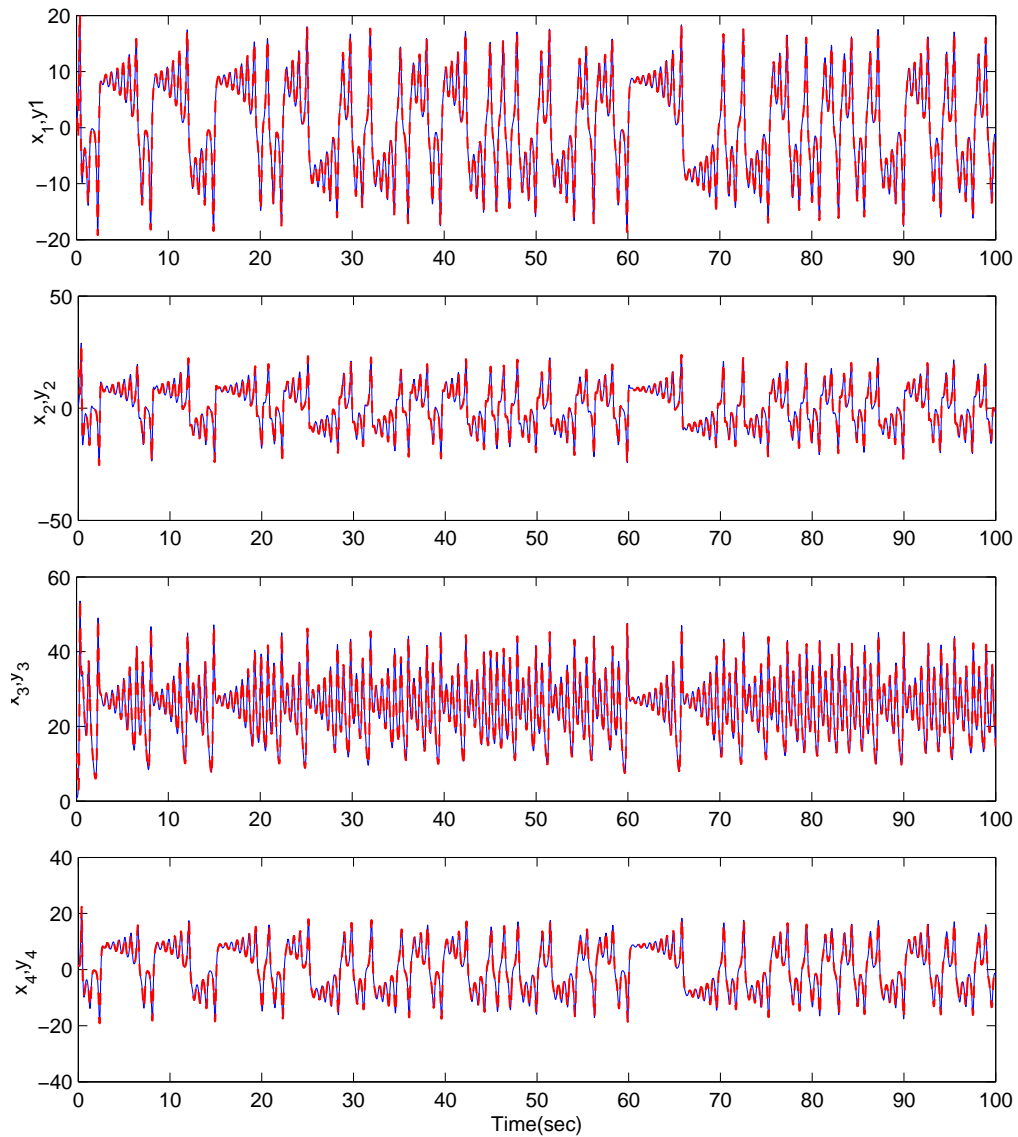


FIGURE 2.12 Réponse de l'état du système (2.65) et du système (2.66).

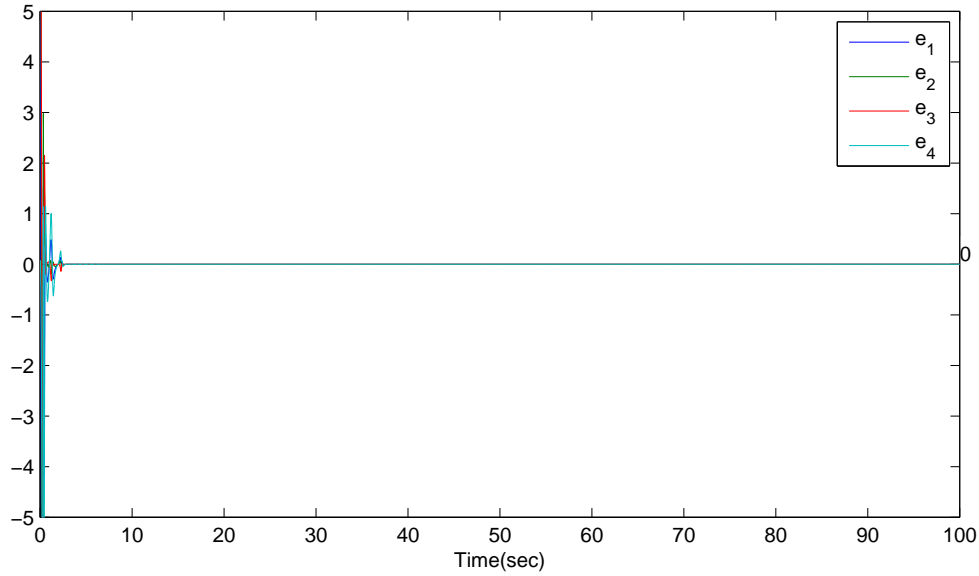


FIGURE 2.13 Dynamique des erreurs de synchronisation du système (2.65) et le système (2.66).

De toute évidence, la nouvelle fonction est LI à d'autres fonctions. Avec cette modification, l'identification de q_2 est réussie, comme indiqué dans la Fig. 2.15.

Le procédé ci-dessus montre que le problème de la synchronisation interne peut être résolu en modifiant l'élément fonctionnel. Cependant, dans de nombreux cas, en changeant la structure du système est peu pratique ou non autorisé. Ainsi, le problème de synchronisation interne peut être résolu en ajoutant ou en injectant un signal ω_i externe tout en gardant la même structure des fonctions.

Pour illustrer cette idée, nous reprenons le même exemple que précédemment. Le signal externe peut être une constante ou d'un signal variant dans le temps

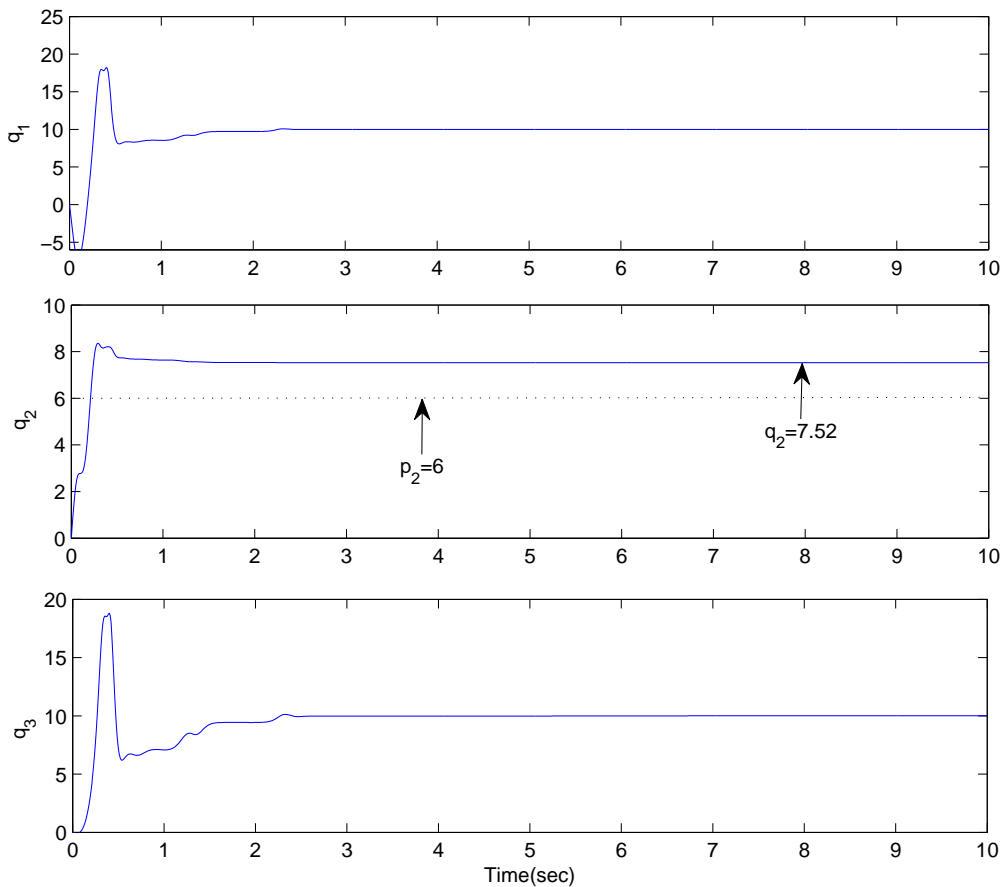
On rappelle que la raison de l'échec de l'identification est du à la dépendance linéaire des états x_4 et x_1 , ce qui a conduit $F(x(t)) = x_4 - x_1 \rightarrow 0$ que $t \rightarrow \infty$. Afin de perturber la synchronisation entre les états, nous pouvons ajouter le signal à x_1 ou x_4 :

$$\dot{x}_1 = p_1(x_2 - x_1) + p_2(x_4 - x_1) + \omega_1,$$

$$\dot{x}_4 = -(x_4 - x_1) + p_3(x_2 - x_1) + \omega_4.$$

En faisant cela, la synchronisation interne entre x_1 et x_4 pourrait être éliminée.

Pour la simulation numérique, deux cas sont considérés pour le signal externe ; dans le

FIGURE 2.14 Variation des paramètres q_1, q_2, q_3 .

premier cas il est choisi comme un signal variable dans le temps : $\omega_1 = -x_2$ et est ajouté au premier état x_1 . Dans le second cas, le signal est ajouté à x_4 , comme un signal constant $\omega_4 = 20$.

2.3.3 Synchronisation par éliminateur de perturbation (EOP)

Pour le problème de la synchronisation externe, nous avons remarqué que les défaillances de l'identification sont causés par la dépendance linéaire des fonctions des systèmes. Il est évident que pour le premier commutateur, $i = 1$, les termes $-x_1$, x_3 et $-(x_2 - x_1)$ sont toujours linéairement dépendants à $(y_2 - y_1)$, $-y_3$, y_1 , donc les six paramètres inconnus ne seront pas identifiés. Dans le second commutateur, le terme x_1 est toujours linéairement indépendant avec les termes $(y_2 - y_1)$; par conséquent, si nous laissons la synchronisation se produire entre les états y_1 et x_2 cela conduirait à la création de l'orbite de synchronisation

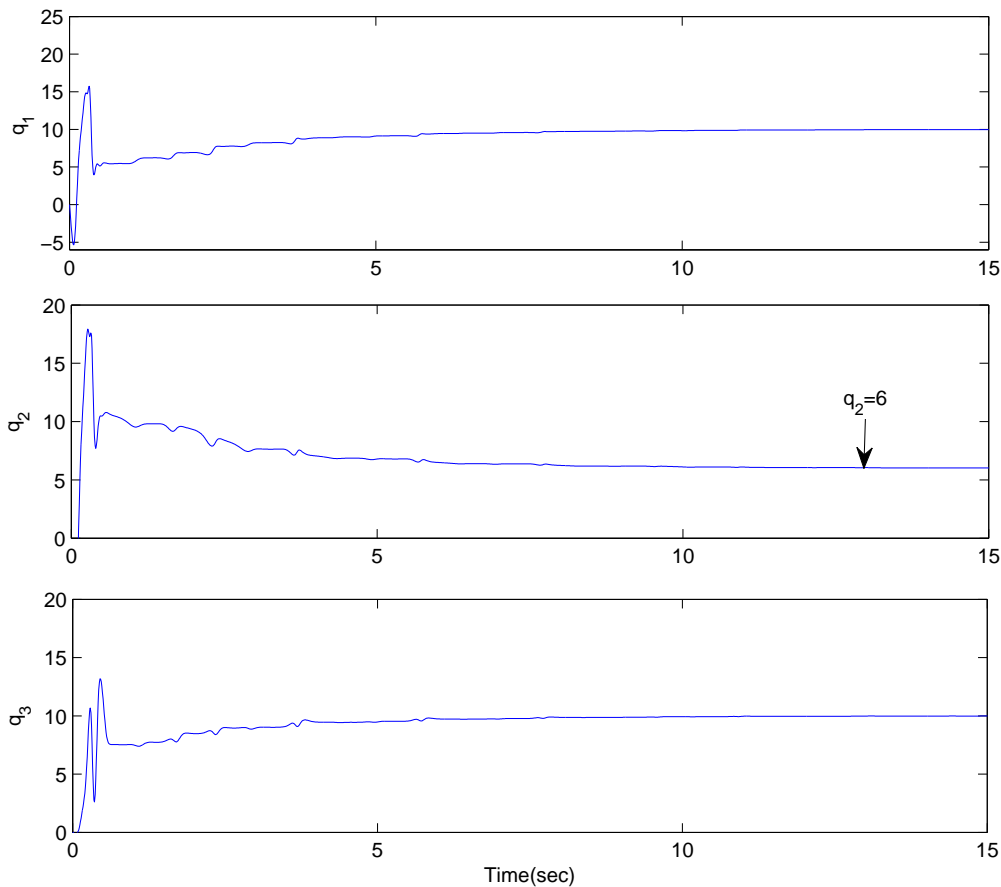


FIGURE 2.15 Variation des paramètres q_1, q_2, q_3 en modifiant la structure des termes de fonction.

$y_1(t) = x_2(t)$; par conséquent, les deux paramètres inconnus a_2 et c_1 seraient identifiés. De même, si y_3 et x_1 sont choisis comme les couples de synchronisation; les termes de fonction $-y_3$ et $(x_2 - x_1)$ sont linéaires indépendants sur l'orbite de synchronisation $y_3 = x_1$. Par conséquent, les paramètres inconnus b_2 et a_1 peuvent également être identifiés. La même chose pour les deux états y_2 et x_3 , s'ils sont synchronisés, les termes de fonction y_1 et $-x_3$ sont linéairement indépendants sur l'orbite créé $y_2 = x_3$, ce qui indique qu'il est possible d'identifier les deux derniers paramètres inconnus c_2 et b_1 .

Par conséquent, l'une des solutions du problème de synchronisation externe est inspiré de la synchronisation multi-commutation et est basé sur la modification des orbites de synchronisation. Cependant, en changeant les états, nous résolvons le problème de la condition LI et en même temps, nous perdons la synchronisation complète entre les systèmes.

Dans ce qui suit nous proposons une nouvelle approche pour résoudre ce problème [94].

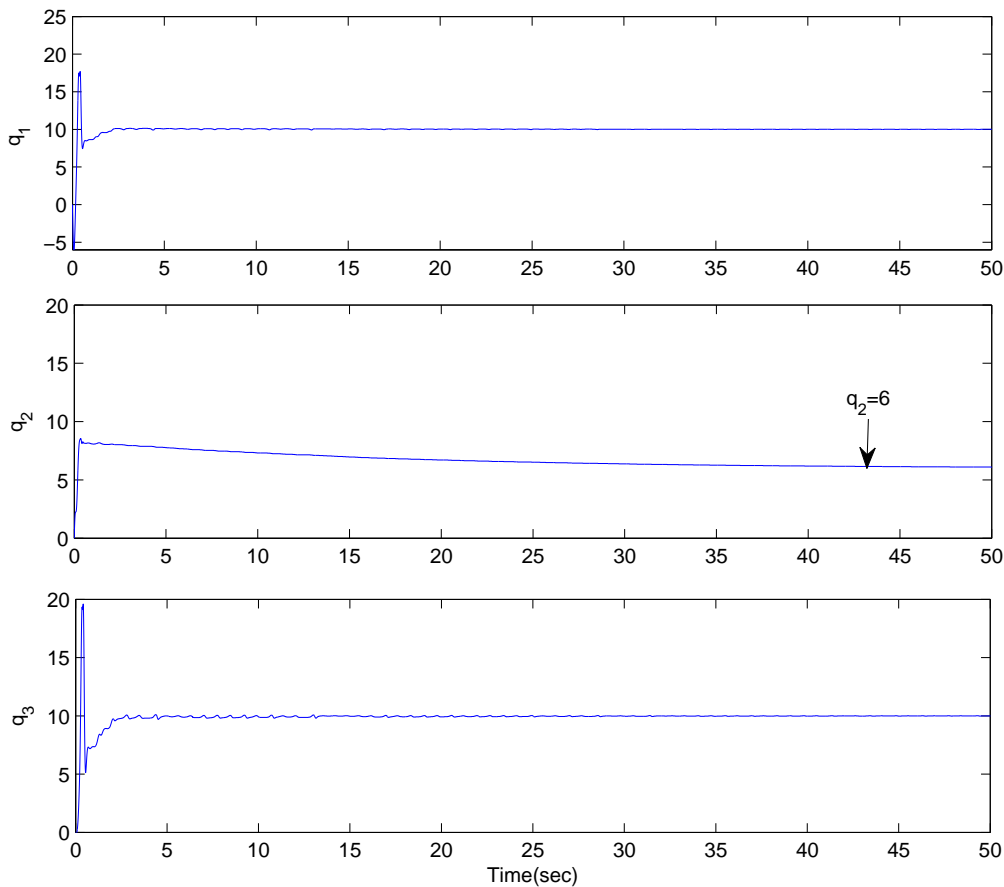


FIGURE 2.16 Variation des paramètres q_1, q_2, q_3 en ajoutant de signal variant dans le temps $\omega_1 = -x_2$.

Si l'on suppose que les incertitudes sur les paramètres sont considérés comme des termes ajoutés aux systèmes. Ces termes vont affecter le comportement du système et le contrôleur principal ne peut pas gérer toute nouvelle dynamique en dehors de son environnement. Toutefois, si un autre système fonctionne pour éliminer ces termes, la conception du contrôleur principal devient très simple. L'objectif principal d'éliminer les termes de perturbation est d'éviter le problème de la dépendance linéaire des fonctions. Par conséquent, le contrôleur est conçu en deux parties. La première partie est l'éliminateur de perturbation (EOP) dont la tâche est d'éliminer les termes de perturbation en raison des incertitudes sur les paramètres, et le second sert à synchroniser les deux systèmes Fig. 2.18.

Le choix de la méthode utilisée pour concevoir le EOP est important. La méthode choisie doit avoir la capacité de traiter avec le système sans connaître son modèle. Par conséquent, le contrôle par la logique floue est choisie comme EOP en raison de sa structure spéciale

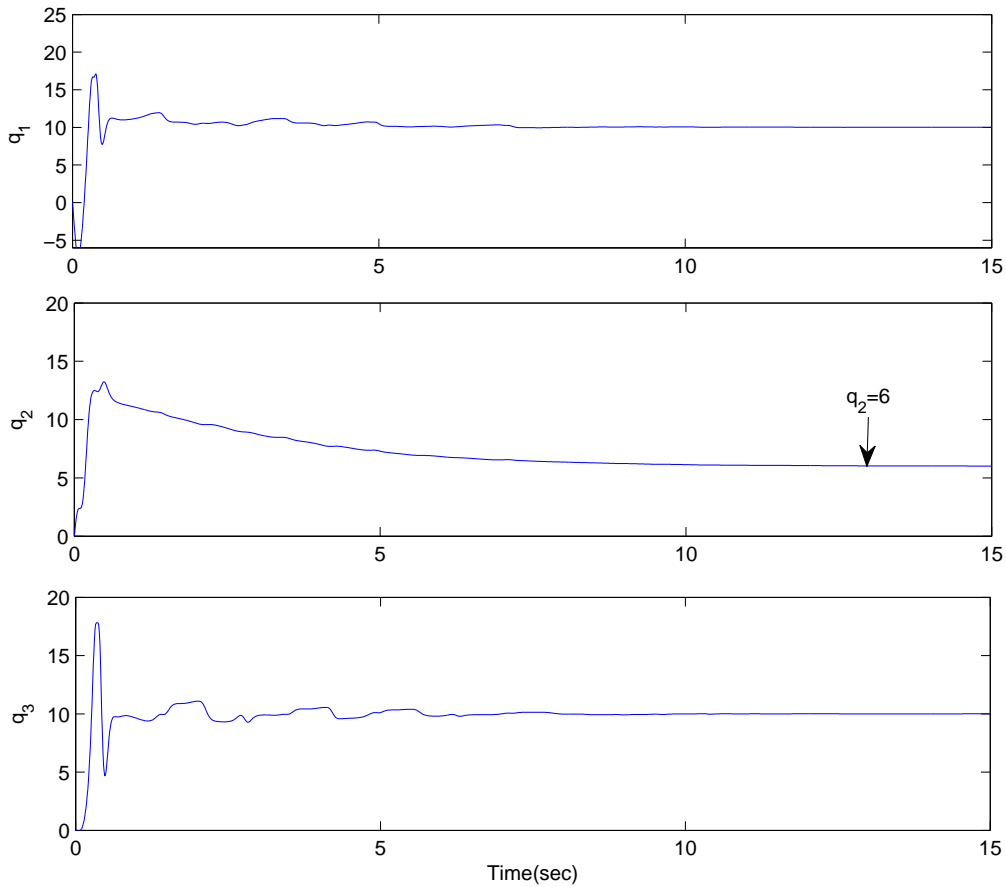


FIGURE 2.17 Variation des paramètres q_1, q_2, q_3 en ajoutant un signal constant $\omega_4 = 20$

qui aide à concevoir un contrôleur efficace pour les systèmes non linéaires, même si les modèles ne sont pas exacts ou non disponibles et que les paramètres sont incertains. Il est facile d'assurer la stabilité par la compréhension du comportement du système contrôlé en ajustant les règles au lieu d'utiliser les termes non linéaires des modèles qui déclenchent certains problèmes tels que les incertitudes sur les paramètres [95–97].

Le FLC a été utilisé pour la synchronisation des systèmes chaotiques incertains et a montré une bonne performance par rapport à de nombreuses approches de contrôle classiques [62, 98–101].

Dans notre proposition, un contrôleur PI flou est donc utilisé pour faire face aux incertitudes et sert comme éliminateur de perturbations. La conception du contrôleur PI flou sera détaillé dans le prochain chapitre.

Le contrôleur doit donc éliminer les termes de perturbation en fonction des entrées qui représentent l'erreur et le changement du taux d'erreur entre les deux systèmes chaotiques.

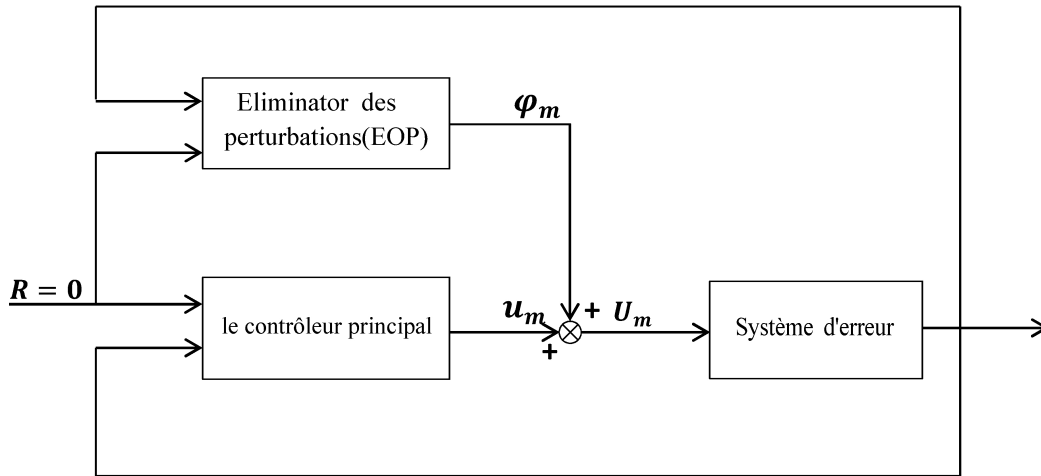


FIGURE 2.18 Schéma de l'EOP et le contrôleur principal, où $R(t) = 0$ est la référence du système d'erreur.

La question est de savoir comment faire pour que le contrôleur agisse sur la variation de l'erreur qui provient uniquement de l'effet des incertitudes sur les paramètres? Afin de résoudre ce problème, un autre contrôleur doit être conçu. Le rôle du nouveau contrôleur est d'assurer la synchronisation lorsque les paramètres utilisés pour la conception de celui-ci sont les mêmes que les paramètres du maître et le système esclave sans incertitude. La capacité du second dispositif de commande se limite à assurer la synchronisation quand il n'y a pas d'incertitudes.

Nous allons définir à nouveau le problème de synchronisation. Considérons deux systèmes chaotiques qui doivent être synchronisés. En tant que maître, le système suivant est défini :

$$\begin{cases} \dot{x}_1 = f_1(x, t) \\ \dot{x}_2 = f_2(x, t), \\ \vdots \\ \dot{x}_n = f_n(x, t) \end{cases} \quad (2.69)$$

$$x = [x_1, x_2, \dots, x_n] \in \mathfrak{R}^n,$$

et en tant que système esclave

$$\begin{cases} \dot{y}_1 = h_1(y,t) + g_1(y,t) + u_1(y,x,t) \\ \dot{y}_2 = h_2(y,t) + g_2(y,t) + u_2(y,x,t), \\ \vdots \\ \dot{y}_n = h_n(y,t) + g_n(y,t) + u_n(y,x,t) \end{cases} \quad (2.70)$$

$$y = [y_1, y_2, \dots, y_n] \in \mathfrak{R}^n,$$

où $x(t), y(t) \in \mathfrak{R}^n$ représentent les vecteurs d'état du maître et le système esclave respectivement. $f, h : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$ sont des fonctions vectorielles non linéaires continues, $g(t, y) : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$ sont les termes de perturbation résultant des incertitudes sur les paramètres ; ils sont censés être bornés. $u_1(t, x, y), u_2(t, x, y), \dots, u_n(t, x, y)$ sont les entrées de commande soient conçues.

Les erreurs de synchronisation sont définies par

$$e_m = y_m - x_m, m = 1, 2, \dots, n, \quad (2.71)$$

et leur dynamique par

$$\begin{cases} \dot{e}_1 = h_1(y,t) - f_1(x,t) + g_1(y,t) + u_1(y,x,t) \\ \dot{e}_2 = h_2(y,t) - f_2(x,t) + g_2(y,t) + u_2(y,x,t). \\ \vdots \\ \dot{e}_n = h_n(y,t) - f_n(x,t) + g_n(y,t) + u_n(y,x,t) \end{cases} \quad (2.72)$$

La sortie EOP agit sur la variation de l'erreur. L'apparition d'erreurs entre le maître et l'esclave indique que le contrôleur principal perd la synchronisation en raison de l'existence des termes de perturbation. L'EOP réagit en ajoutant de nouveaux termes ($\varphi_m = u_{PI_m}$) à (2.72) de manière à les éliminer. Ainsi, (2.72) devient

$$\begin{cases} \dot{e}_1 = h_1(y,t) - f_1(x,t) + g_1(y,t) + U_1(t) \\ \dot{e}_2 = h_2(y,t) - f_2(x,t) + g_2(y,t) + U_2(t), \\ \vdots \\ \dot{e}_n = h_n(y,t) - f_n(x,t) + g_n(y,t) + U_3(t) \end{cases} \quad (2.73)$$

avec

$$\begin{cases} U_1(t) = u_1(y, x, t) + \varphi_1(t) \\ U_2(t) = u_2(y, x, t) + \varphi_2(t) \\ \vdots \\ U_n(t) = u_n(y, x, t) + \varphi_n(t) \end{cases} \quad (2.74)$$

Si EOP élimine les termes de perturbation pendant le processus de synchronisation, la condition suivante est maintenue :

$$g_m(t) + \varphi_m(t) = 0, \quad (2.75)$$

et (2.73) devient

$$\begin{cases} \dot{e}_1 = h_1(y, t) - f_1(x, t) + u_1(y, x, t) \\ \dot{e}_2 = h_2(y, t) - f_2(x, t) + u_2(y, x, t) \\ \vdots \\ \dot{e}_n = h_n(y, t) - f_n(x, t) + u_n(y, x, t) \end{cases} \quad (2.76)$$

Cet ensemble d'équations représente un système simple, sans perturbations, et il est facile de concevoir un contrôleur en utilisant la théorie de Lyapunov.

Pour la conception du dispositif de commande principal, la méthode de contrôle non linéaire actif est choisi [102] [25] [103].

Pour illustrer l'efficacité de l'approche proposée, on l'applique pour la synchronisation des deux systèmes de Lorenz incertains. Le système maître est

$$\begin{cases} \dot{x}_1 = a_1(x_2 - x_1) \\ \dot{x}_2 = (-x_1x_3 + c_1x_1 - x_2), \\ \dot{x}_3 = x_1x_2 - b_1x_3 \end{cases} \quad (2.77)$$

où x_1, x_2, x_3 sont les variables d'état, a_1, c_1 et b_1 sont des paramètres incertains positifs.

Le système esclave est

$$\begin{cases} \dot{y}_1 = a_2(y_2 - y_1) + u_1 \\ \dot{y}_2 = (-y_1y_3 + c_2y_1 - y_2) + u_2, \\ \dot{y}_3 = y_1y_2 - b_2y_3 + u_3 \end{cases} \quad (2.78)$$

où y_1, y_2, y_3 sont les variables d'état, a_2, c_2 et b_2 sont des paramètres incertains positifs, et u_1, u_2, u_3 sont les contrôleurs soient conçus.

La dynamique du système d'erreur est définie comme

$$\begin{cases} e_1 = y_1 - x_1 \\ e_2 = y_2 - x_2, \\ e_3 = y_3 - x_3 \end{cases} \quad (2.79)$$

tandis que les erreurs des paramètres des systèmes sont donnés par

$$\begin{cases} e_a = a_2 - a_1 \\ e_c = c_2 - c_1. \\ e_b = b_2 - b_1 \end{cases} \quad (2.80)$$

Après un calcul simple, les états d'erreur sont donnés par

$$\begin{cases} \dot{e}_1 = g_1(y) + a_1(e_2 - e_1) + u_1 \\ \dot{e}_2 = g_2(y) - y_2 - y_1 y_3 + c_1 e_1 + x_2 + x_1 x_3 + u_2, \\ \dot{e}_3 = g_3(y) - b_1 e_3 + y_1 y_2 - x_1 x_2 + u_3 \end{cases} \quad (2.81)$$

où

$$\begin{cases} g_1(y) = e_a(y_2 - y_1) \\ g_2(y) = e_c y_1, \\ g_3(y) = -e_b y_3 \end{cases} \quad (2.82)$$

représente les termes de perturbation. Tous les termes de perturbation peuvent être négligés en raison des actions de l'EOP, et (3.102) devient

$$\begin{cases} \dot{e}_1 = a_1(e_2 - e_1) + u_1 \\ \dot{e}_2 = -y_2 - y_1 y_3 + c_1 e_1 + x_2 + x_1 x_3 + u_2. \\ \dot{e}_3 = -b_1 e_3 + y_1 y_2 - x_1 x_2 + u_3 \end{cases} \quad (2.83)$$

Par conséquent, les contrôleurs sont définis comme suit :

$$\begin{cases} u_1 = -e_1 - a_1(e_2 - e_1) \\ u_2 = -e_2 + y_2 + y_1 y_3 - c_1 e_1 - x_2 - x_1 x_3. \\ u_3 = -e_3 - y_1 y_2 + x_1 x_2 + b_1 e_3 \end{cases} \quad (2.84)$$

Les paramètres utilisés dans la conception des contrôleurs actifs sont fixés à $a_1 = a_2 = 10$, $b_1 = b_2 = 8/3$ et $c_1 = c_2 = 28$, ce qui limite le contrôleur pour atteindre la synchronisation lorsque les paramètres de les deux systèmes sont les mêmes que les paramètres fixes des contrôleurs actifs. La substitution de (2.84) dans (2.83) donne un système linéaire de la

forme

$$\begin{cases} \dot{e}_1 = -e_1 \\ \dot{e}_2 = -e_2 \\ \dot{e}_3 = -e_3 \end{cases} \quad (2.85)$$

On choisit une fonction de Lyapunov de la forme

$$V(e) = \frac{1}{2}(e_1^2 + e_2^2 + e_3^2). \quad (2.86)$$

Le dérivé de V est

$$\dot{V}(e) = -e_1^2 - e_2^2 - e_3^2, \quad (2.87)$$

ceci implique que le système contrôlé est globalement asymptotiquement stable.

L'estimation des paramètres a_2, b_2 et c_2 est déduit de (2.82) et (2.75) comme suit :

$$\begin{cases} \hat{a}_2(t) = \frac{-\varphi_1(t)}{y_2(t)-y_1(t)} + a_1 & y_2 - y_1 \neq 0 \\ \hat{c}_2(t) = \frac{-\varphi_2(t)}{y_1(t)} + c_1 & y_1 \neq 0 \\ \hat{b}_2(t) = \frac{\varphi_3(t)}{y_3(t)} + b_1 & y_3 \neq 0 \end{cases} \quad (2.88)$$

Fig. 2.19 montre le schéma de contrôle détaillé utilisé pour la synchronisation des systèmes chaotiques incertains.

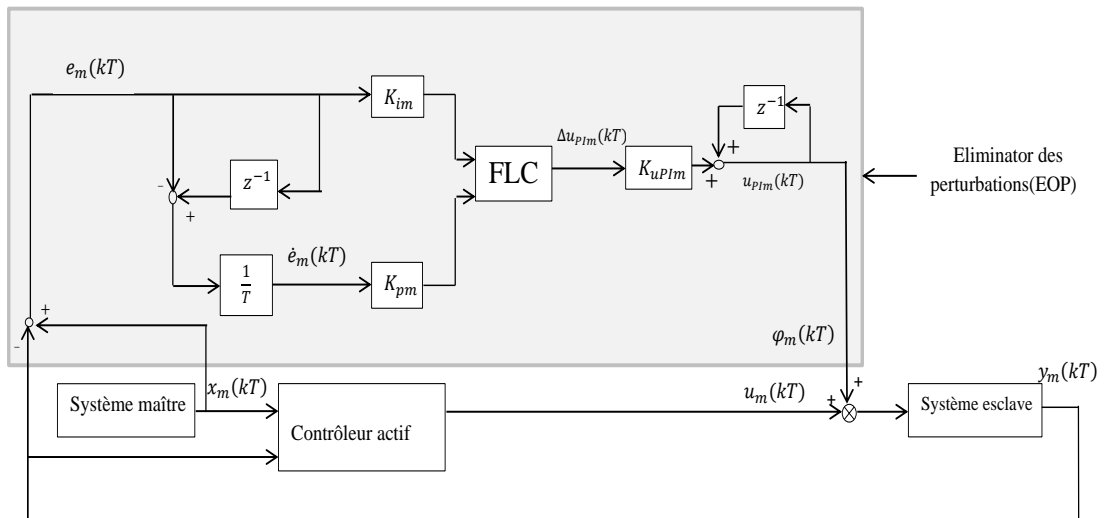


FIGURE 2.19 Le système de contrôle proposé.

L'approche décrite est appliquée pour la synchronisation des deux systèmes de Lorenz

incertains. Il y a deux objectifs à atteindre : la première est de rendre la dynamique d'erreur asymptotiquement stable, et la seconde est d'estimer les paramètres inconnus des systèmes.

Pour la simulation numérique, on utilise la méthode de Runge-Kutta d'ordre quatre avec $T = 0.001$ est utilisé. Les paramètres du maître et les systèmes d'esclaves sans perturbations sont choisis comme suit :

$$a_1 = 10, c_1 = 28, b_1 = 8/3,$$

$$a_2 = 10, c_2 = 8/3, b_2 = 8/3.$$

Les paramètres des systèmes sont aléatoirement perturbés en utilisant un générateur de perturbation (GP) sur la base de MATLAB fonction «rand». Les conditions initiales sont prises comme :

$$x_1(0) = -5, x_2(0) = 5, x_3(0) = 0,$$

$$y_1(0) = -6, y_2(0) = -8, y_3(0) = 12.$$

Les paramètres sont choisis comme suit : $L_1 = L_2 = L_3 = 1$,
 $K_{p1} = 0.9, K_{i1} = 2, K_{uPI1} = 2.43$,
 $K_{p2} = 1.4, K_{i2} = 0.2, K_{uPI2} = 1.3$,
 $K_{p3} = 1, K_{i3} = 1.2, K_{uPI3} = 2$.

Afin bien mettre en évidence les performances de l'EOP et du contrôleur principal, l'intervalle de temps de simulation est divisé en cinq parties. Chaque partie correspond à une situation ; ce qui permet de comprendre les actions de l'EOP. Les différentes situations considérées sont résumées dans le tableau 2.1.

Fig. 2.20 et Fig. 2.21 montrent le comportement du contrôleur principal et l'EOP concernant la synchronisation des deux systèmes. Les comportements suivants sont observés pour les différents cas rapportés dans le tableau 2.1 :

1. Pour $0 < t < 5$, les contrôleurs sont désactivés, de sorte que les trajectoires des deux

TABLE 2.1 Le tableau des états.

| Temps(sec) | Contrôleur | Générateur Perturbations | EOP |
|------------|------------|--------------------------|-----|
| 0 to 5 | Off | Off | Off |
| 5 to 15 | On | Off | Off |
| 15 to 25 | On | On | Off |
| 25 to 35 | On | On | On |
| 35 to 50 | On | Off | On |

systèmes ne soient pas les mêmes.

2. Pour $5 < t < 15$, il n'y a pas de termes de perturbation due à des incertitudes sur les paramètres, le contrôleur est activé, et les deux systèmes sont parfaitement synchronisés.

3. Pour $15 < t < 25$, le générateur de perturbation est mis sous tension, ce facteur détruit la synchronisation entre les deux systèmes. Le contrôleur ne peut pas gérer la nouvelle situation.

4. Pour $25 < t < 35$, EOP est activé, la dynamique d'erreur convergent vers zéro, et le contrôleur fonctionne bien comme s'il n'y avait aucune perturbation.

5. Pour $35 < t < 50$, EOP s'arrête exactement quand la perturbation des paramètres est désactivée.

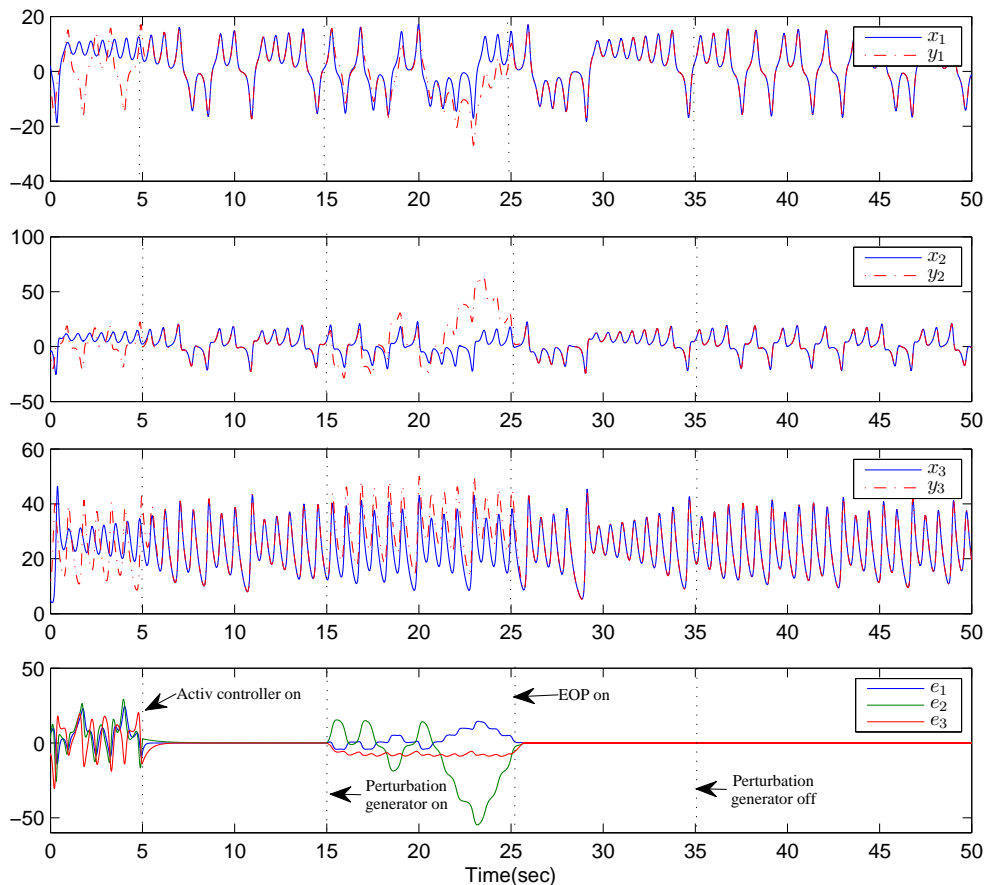


FIGURE 2.20 Le comportement de synchronisation dans chaque cas.

La deuxième partie de la simulation se concentre sur l'estimation des paramètres incon-

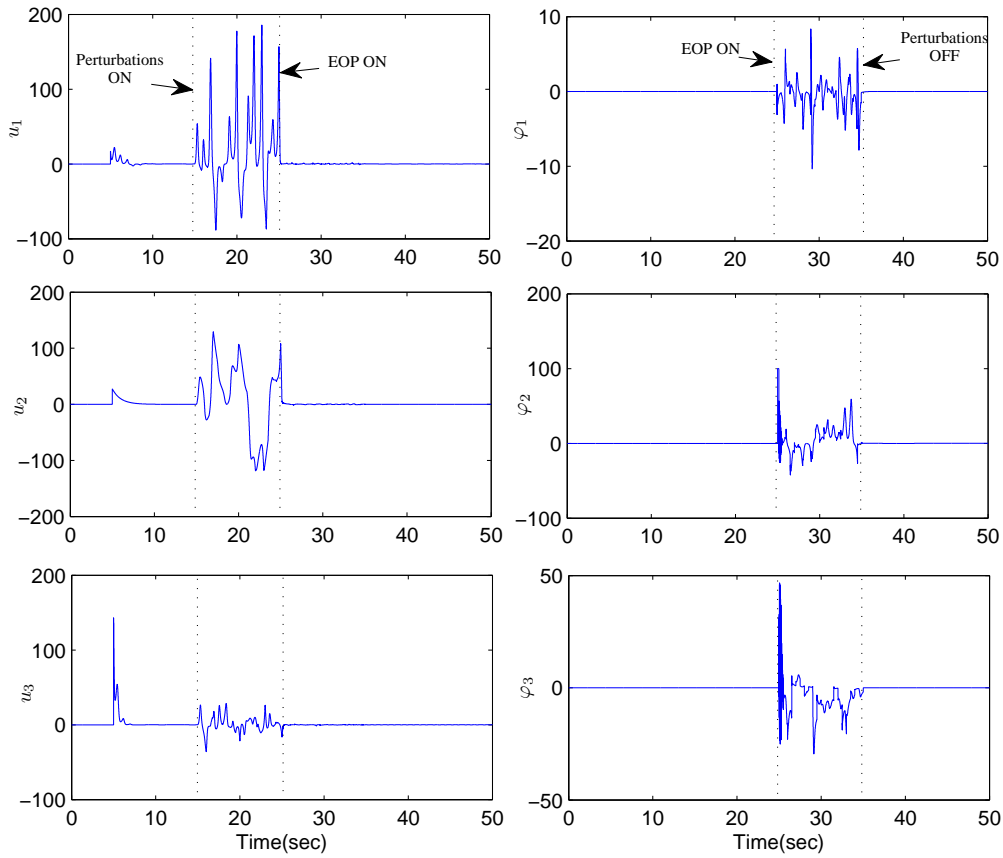


FIGURE 2.21 Les actions de EOP et le contrôleur principal.

nus des deux systèmes afin de vérifier les performances du système adaptatif. Il est possible d'identifier les paramètres inconnus en utilisant (2.88). Cependant, en présence des paramètres inconnus dans les deux systèmes, (2.88) ne donne que l'erreur entre les paramètres des systèmes et devient

$$\begin{cases} e_a(t) = \frac{-\varphi_1(t)}{y_2(t)-y_1(t)} & y_2 - y_1 \neq 0 \\ e_c(t) = \frac{-\varphi_2(t)}{y_1(t)} & y_1 \neq 0. \\ e_b(t) = \frac{\varphi_3(t)}{y_3(t)} & y_3 \neq 0 \end{cases} \quad (2.89)$$

Fig. 2.22 représente l'erreur entre les paramètres des deux systèmes dans le cas où les valeurs initiales des paramètres inconnus sont sélectionnés comme suit : $\hat{a}_1(0) = 2, \hat{b}_1(0) = 10, \hat{c}_1(0) = -6, \hat{a}_2(0) = -2, \hat{b}_2(0) = 5, \hat{c}_2(0) = 1$. Les vraies valeurs des paramètres du système maître sont $a_1 = 11, b_1 = 8/3 + 0.4, c_1 = 26$, tandis que les paramètres du système esclave sont $a_2 = 9, b_2 = 8/3 + 0.2, c_2 = 27$.

Pour trouver les vraies valeurs des paramètres, les paramètres inconnus du maître ou du système esclave doivent au moins être estimés. Pour résoudre ce problème, un système de référence avec des paramètres connus doit être conçu. Le système de référence permet d'estimer les paramètres du maître ou du système esclave. Par conséquent, les paramètres inconnus du système esclave peuvent être estimés à l'aide (2.88) comme suit :

$$\begin{cases} \hat{a}_2(t) = \frac{-\varphi_1(t)}{y_2(t)-y_1(t)} + a_r & y_2 - y_1 \neq 0 \\ \hat{c}_2(t) = \frac{-\varphi_2(t)}{y_1(t)} + c_r & y_1 \neq 0, \\ \hat{b}_2(t) = \frac{\varphi_3(t)}{y_3(t)} + b_r & y_3 \neq 0 \end{cases} \quad (2.90)$$

où $a_r = 10, b_r = 8/3, c_r = 28$ sont les paramètres du système de référence.

L'estimation des paramètres inconnus du système maître peut être dérivée de (2.89) et

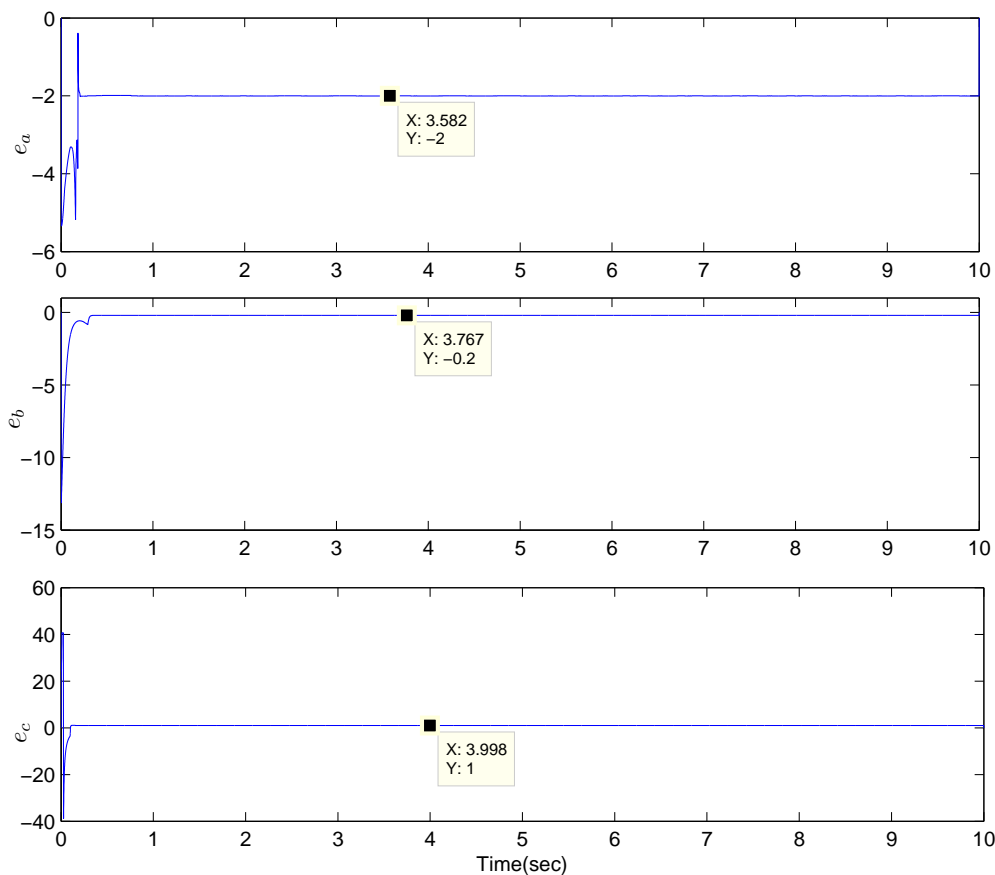


FIGURE 2.22 Les erreurs de paramètres à l'aide de (2.89).

(2.90) comme suit :

$$\begin{cases} \hat{a}_1(t) = \hat{a}_2(t) - e_a(t) \\ \hat{c}_1(t) = \hat{c}_2(t) - e_c(t) \\ \hat{b}_1(t) = \hat{b}_2(t) - e_b(t) \end{cases} \quad (2.91)$$

Fig. 2.23 montre l'estimation des paramètres du maître et le système esclave en utilisant (2.89), (2.90) et (2.91).

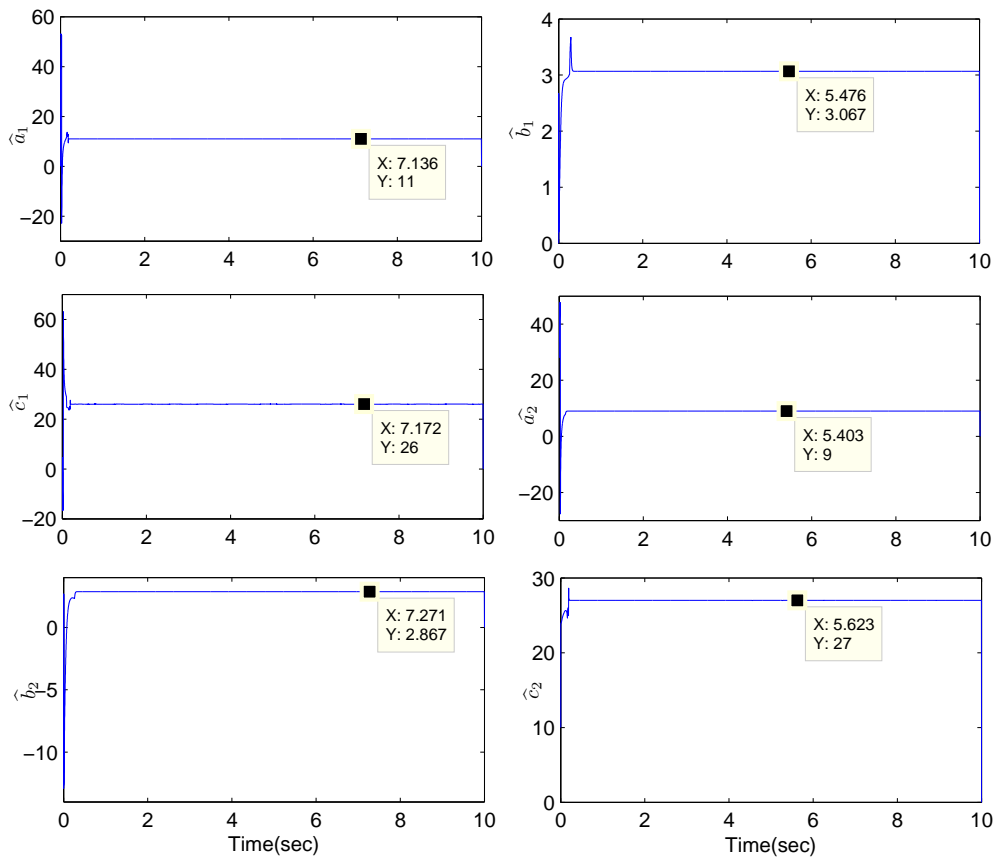


FIGURE 2.23 L'identification des paramètres.

2.4 Conclusion

Dans ce chapitre, nous avons étudié la performance de la méthode de contrôle adaptatif avec un contrôle non linéaire actif pour la synchronisation des systèmes chaotiques incertains. Cette étude révèle que de nombreux défauts ont été signalés en particulier le problème de l'estimation des paramètres inconnus. En outre, la synchronisation crée un obstacle pour estimer les paramètres inconnus en raison du problème de la dépendance linéaire des fonctions. De nombreuses améliorations ont été proposées pour résoudre ce problème en mettant l'accent sur l'importance de la condition LI. Cependant, dans la synchronisation des systèmes chaotiques incertains, La commande adaptative peut gérer le problème des incertitudes sur les paramètres même si les lois d'adaptation ne sont pas estimer les vraies valeurs des paramètres. En outre, les tentatives pour rendre les lois d'adaptation convergentes vers des valeurs réelles conduisent à la destruction de la synchronisation. Par conséquent, si le but du dispositif de commande adaptative est de parvenir à la synchronisation des systèmes chaotiques incertains, il n'y a pas besoin de résoudre le problème de l'estimation des paramètres de la manière qui conduit à la destruction de la synchronisation. Par contre, si l'objectif du problème étudié est d'atteindre l'estimation des paramètres, les améliorations ont besoin de quelques modifications du problème de synchronisation. Enfin, nous pouvons dire que la tâche de la synchronisation et de l'estimation des paramètres inconnus en utilisant le système non linéaire actif de contrôle est très difficile, surtout lorsque les paramètres des deux systèmes ne sont pas connus.

Chapitre 3

Commande Prédicative Généralisée et Synchronisation des Systèmes Chaotiques Incertains

3.1 Introduction

La commande prédictive est un terme général qui englobe un ensemble de méthodes différentes (PFC, DMC, GPC, EPSAC, NLPC...). Néanmoins, toutes ces techniques utilisent la même philosophie de contrôle et le principe de fonctionnement est le même .

La commande prédictive généralisée (GPC), une méthode à usage général, a été proposée par Clarke et al. en 1987 [104] comme algorithme particulier de la commande prédictive (MPC). Cette méthode est plus efficace, en particulier lorsque des contraintes sont impliquées. Dans ce cas, la plupart des algorithmes MPC se tournent vers l'inégalité de matrice linéaire (linear matrix inequality(LMI)) qui augmente le nombre d'opérations de la matrice, en particulier l'opération matrice inverse. Ces opérations rendent la méthode MPC plus complexe, comparée aux algorithmes GPC en termes de mise en œuvre.

Les approches de contrôle prédictif à base de modèles (MBPC) comporte trois parties principales : un modèle du système contrôlé, une fonction coût, et un algorithme chargé de calculer une séquence de futurs signaux de commande afin de minimiser la fonction coût sur un horizon de prédiction. D'une manière générale, ces algorithmes ont besoin d'un mode précis du système contrôlé pour être utilisé par l'algorithme d'optimisation pour obtenir les signaux de commande optimales.

Dans ce chapitre, deux approches différentes de la GPC ; l'une basée sur les contrôleurs

PID flous et l'autre basées sur les réseaux de neurones, sont utilisées pour la synchronisation des systèmes chaotiques incertains.

3.2 Algorithme GPC classique

La méthode GPC est une commande prédictive à longue portée qui génère un ensemble de commande futures qui minimise la fonction coût. Cependant, seul le premier vecteur de la séquence est envoyée au processus. Pour l'étape suivante, une nouvelle séquence optimale doit être déterminée [105, 106]. la Fig. 3.1 explique l'idée principale derrière le contrôle prédictif généralisée. La sortie du modèle utilisée par l'optimiseur, est divisée en deux parties distinctes : la réponse libre et la réponse forcée. Dans le cas d'une réponse libre, l'entrée précédente est maintenue constante et maintient sa dernière valeur dans le futur. Dans le cas de la réponse forcée ; l'entrée est prise égale à zéro dans le passé et à la variation de commande dans le futur.

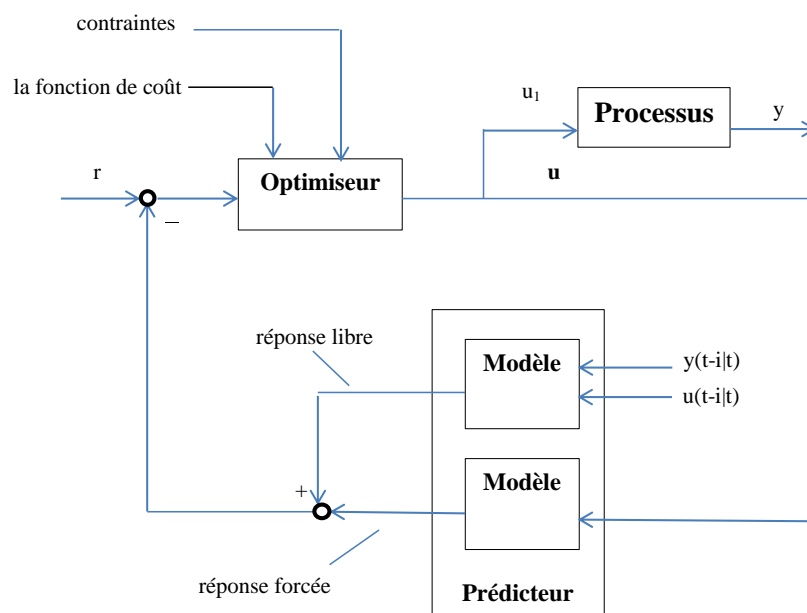


FIGURE 3.1 Structure de base de la GPC.

Le critère de performance est représenté par une fonction coût quadratique considérant l'erreur de poursuite et l'effort de commande sur un horizon glissant de la forme :

$$J(u, t) = \sum_{j=N_1}^{N_2} [y(t+j) - r(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2, \quad (3.1)$$

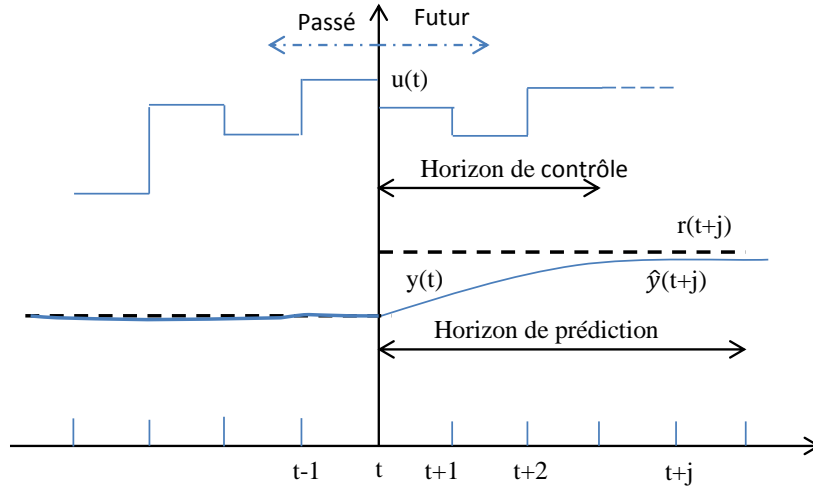


FIGURE 3.2 Principe de la commande prédictive à horizon étendu

où N_1 est l'horizon de prédiction minimum, N_2 est l'horizon de prédiction maximal, et N_u est l'horizon de contrôle. $\lambda(j)$ représente le facteur de pondération sur l'effort de commande. La méthode est basée sur un modèle ARIMAX (Auto Regressive Integrated Moving Average with eXternal inputs) dont l'équation est donnée par

$$\begin{aligned} A(q^{-1})y(t) &= B(q^{-1})u(t-1) + \frac{C(q^{-1})}{\Delta(q^{-1})}\zeta(t), \\ A(q^{-1})\Delta y(t) &= B(q^{-1})\Delta u(t-1) + C(q^{-1})\zeta(t), \end{aligned} \quad (3.2)$$

où q^{-1} représente l'opérateur de retard, et $\zeta(t)$ un bruit blanc de moyenne nulle et A, B et C des polynômes définis comme suit :

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{na}q^{-na} \\ B(q^{-1}) &= b_0 + b_1q^{-1} + \dots + b_{nb}q^{-nb} \\ C(q^{-1}) &= 1 + c_1q^{-1} + \dots + c_{nc}q^{-nc} \\ \Delta(q^{-1}) &= 1 - q^{-1} \end{aligned} \quad (3.3)$$

Le modèle ARIMAX aint intervenir les incréments de commande et non la commande effective. Cet aspect permet d'imposer au final une action intégrale au sein du régulateur et assure par conséquent une erreur statique nulle pour des consignes et perturbations constantes. L'aspect incrémental du modèle se retrouve aussi dans le critère par la présence de Δu .

Pour obtenir le prédicteur, l'équation Diophantine suivante est utilisée :

$$C(q^{-1}) = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}), \quad (3.4)$$

où les polynômes E_j et F_j sont de degré $j - 1$ ($\deg(E_j) = j - 1$) et na respectivement. En Multipliant (3.2) par $E_j \Delta q^j$ et son remplacement pour $E_j A \Delta$ de (3.4) nous obtenons

$$y(t+j) = \frac{F_j}{C} y(t) + \frac{E_j B}{C} \Delta u(t+j-1) + E_j \zeta(t+j). \quad (3.5)$$

On remarque que $\zeta(t+j)$ est indépendant de l'information passée au temps t et E_j est de degré $j - 1$. Par conséquent, le prédicteur devient

$$y(t+j) = \frac{F_j}{C} y(t) + \frac{E_j B}{C} \Delta u(t+j-1). \quad (3.6)$$

Afin de séparer l'entrée passée de l'entrée future, l'équation Diophantine suivante est introduite :

$$E_j(q^{-1})B(q^{-1}) = G_j(q^{-1})C(q^{-1})q^{-j}\Gamma_j(q^{-1}). \quad (3.7)$$

Cette étape permet d'obtenir séparément la réponse libre et la réponse forcée. Le prédicteur (3.6) devient

$$y(t+j) = F_j y^f(t) + \Gamma_j u^f(t-1) + G_j(q^{-1}) \Delta u(t+j-1), \quad (3.8)$$

où

$$\begin{aligned} u^f(t) &= 1/C(q^{-1}) \Delta u(t), \\ y^f(t) &= 1/C(q^{-1}) y(t). \end{aligned} \quad (3.9)$$

En supposant que toutes les variations futures de l'entrée seront nulles, on obtient la réponse libre comme suit :

$$\hat{y}(t+j|t) = F_j y^f(t) + \Gamma_j u^f(t-1). \quad (3.10)$$

Par conséquent, la prédiction pour les deux parties est donnée par :

$$\hat{y}(t+j) = \hat{y}(t+j|t) + G_j(q^{-1}) \Delta u(t+j-1). \quad (3.11)$$

Si on définit f comme la réponse libre

$$f = [\hat{y}(t+1|t), \dots, \hat{y}(t+N_2|t)]^T, \quad (3.12)$$

et \tilde{u} le vecteur des incréments futurs :

$$\tilde{u} = [\Delta u(t), \dots, \Delta u(t + N_u - 1)]^T, \quad (3.13)$$

Le vecteur de prédiction \hat{y} :

$$\hat{y} = [\hat{y}(t+1), \dots, \hat{y}(t+N_2)]^T, \quad (3.14)$$

Peut être écrit sous la forme :

$$\hat{y} = G\tilde{u} + f, \quad (3.15)$$

où G est de la dimension $N_2 \times N_u$ contenant les éléments des premiers j coefficients des polynômes.

$$G = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_u-1} & g_{N_u-2} & \dots & g_0 \\ \vdots & \vdots & \dots & \vdots \\ g_{N_2-1} & g_{N_2-2} & \dots & g_{N_2-N_u} \end{bmatrix}. \quad (3.16)$$

Le critère quadratique J peut être réécrit sous forme matricielle comme suit :

$$\begin{aligned} J(u, t) &= (y - r)^T (y - r) + \lambda \tilde{u}^T \tilde{u} \\ &= (G\tilde{u} + f - r)^T (G\tilde{u} + f - r) + \lambda \tilde{u}^T \tilde{u}, \end{aligned} \quad (3.17)$$

où

$$r = [r(t+1), \dots, r(t+N_2)]^T. \quad (3.18)$$

La séquence de contrôle optimal obtenu par la résolution du problème d'optimisation du critère quadratique, peut être écrit de la manière suivante :

$$\tilde{u} = [G^T G + \lambda I]^{-1} G^T (r - f). \quad (3.19)$$

On obtient ainsi, la meilleure entrée de commande, qui représente la première entrée du vecteur \tilde{u}

$$u(t) = u(t-1) + K(r - f), \quad (3.20)$$

où $K = [G^T G + \lambda I]^{-1} G^T$.

Les principales étapes de l'algorithme de GPC classique peuvent être résumées à ce qui suit :

1. Définir les polynômes $A(q^{-1})$, $B(q^{-1})$, $C(q^{-1})$ après la linéarisation du système contrôlé.
2. Lire la sortie du système $y(t)$ et le vecteur de la trajectoire de référence r .
3. calculer récursivement les polynômes $G_j(q^{-1})$, $\Gamma_j(q^{-1})$, et $F(q^{-1})$ en résolvant le Diophantine.
4. Former les matrices G , Γ et f
5. calculer le gain de feedback K
6. calculer l'incrément de sortie du régulateur $\tilde{u} = K(r - f)$
7. Sélectionnez le premier élément, $u(t) = u(t - 1) + \Delta u(t)$, $t = t + 1$, retour l'étape 2.

Nous pouvons observer à partir de cette introduction que GPC est une méthode analytique basée sur le modèle ARIMAX. Cette méthode se compose de plusieurs étapes mathématiques, qui en font une procédure complexe de calcul. Cependant, le seul inconvénient majeur de cet algorithme est la linéarisation du processus en utilisant la série de Taylor, puis la transformation de Laplace et Z-transformation pour obtenir les polynômes A et B. Cette étape limite la capacité du contrôleur à traiter des systèmes non linéaires complexes qui sont largement présents dans la réalité. Par conséquent, certains outils avancés ont été utilisés pour rendre cet algorithme plus simple en impliquant d'autres méthodes avancées en termes de précision du modèle et les algorithmes d'optimisation. Par exemple, l'utilisation des réseaux de neurones pour la modélisation, et des algorithmes métaheuristiques tels que le PSO pour résoudre le problème d'optimisation.

3.3 Contrôleur PID floue Predictif

Depuis l'émergence de la théorie des ensembles flous par Zadeh [107], beaucoup de travaux ont été publiés sur le sujet avec des applications dans de nombreux domaines : le traitement de l'image [108], la biologie [109], la robotique [110], ...etc.

Dans le domaine du contrôle des systèmes ; la stratégie dénommée Fuzzy Logic Control (FLC) a été largement utilisée pour le contrôle des systèmes non linéaires [111] [112] [113]. Elle est connue pour ses bonnes performances face aux problèmes engendrés par les incertitudes sur les paramètres et les erreurs de modélisation. Dans de nombreux cas, la fusion

de FLC avec un contrôleur formel génère un nouveau contrôleur efficace. Par exemple, la fusion de FLC avec un régulateur PI génère un régulateur puissant et non conventionnel appelé flou PI [114], qui est plus efficace et apte à commander des systèmes non linéaires.

D'autre part, la FLC a été utilisée avec des approches de contrôle classiques comme un assistant afin d'éviter de nombreux problèmes [115]. Par exemple, l'utilisation de la FLC avec La commande par mode glissant (FSMC) permet d'éviter le problème causé par la suppression de la fonction de signe dans la loi de commande dans (SMC) [116] [117] [118].

Les Contrôleurs PID flous sont classés parmi les approches de contrôle les plus efficaces dans les applications industrielles. Cet ensemble de contrôleurs n'a besoin que de la sortie du système contrôlé afin de trouver la bonne action de contrôle à appliquer, ce qui les place mieux que de nombreuses approches de contrôle classiques qui nécessitent l'utilisation d'un modèle mathématique précis du processus.

Un Contrôleur PID flou est un contrôleur PID classique, dans lequel un contrôleur flou a été ajouté afin d'augmenter sa capacité à gérer la complexité des systèmes non linéaires comportant des incertitudes. Jialiang et al. [119] ont utilisé l'avantage de la simplicité des contrôleurs PID flou afin de réduire la complexité de l'algorithme GPC classique. Ils ont proposé d'utiliser du contrôleur PID flou afin de résoudre le problème d'optimisation et d'atteindre la stabilité asymptotique. Cette approche n'a pas besoin d'un modèle précis du système contrôlé, ce qui la rend apte à faire face à l'hypersensibilité des systèmes chaotiques au cours du processus de synchronisation. L'idée d'utiliser un contrôleur PID flou prédictif permet de se servir des informations fournies par la fonction coût J_m et son taux de variation au lieu du signal d'erreur e_m et son taux de variation \dot{e}_m .

3.3.1 Régulation PID flou prédictive

La régulation PID flou prédictive peut être considérée comme un algorithme GPC basé sur des contrôleurs PID flous. L'algorithme GPC consiste principalement à minimiser une fonction coût qui contient les valeurs prédites. Dans le cas de la synchronisation, le critère à minimiser est le suivant :

$$J_m(k) = \sum_{i=-1}^N [x_m(k-i) - y_m(k-i)]^2 + \lambda \sum_{j=0}^{N_c} [\Delta u_m(k-j)]^2 \quad (3.21)$$

$$J_m(k) = \sum_{i=-1}^N [e_m(k-i)]^2 + \lambda \sum_{j=0}^{N_c} [\Delta u_m(k-j)]^2,$$

où N est l'horizon de prédiction, N_c l'horizon de l'incrément de commande, Δu_m est l'incrément de commande, $\lambda \geq 0$ est un poids de l'incrément de commande.

Pour éviter les étapes mathématiques fastidieuses, les contrôleurs PID flous peuvent être utilisés [119]. Fig. 3.3 représente la structure principale de la régulation PID floue prédictive que nous proposons pour la synchronisation des systèmes chaotiques incertains.

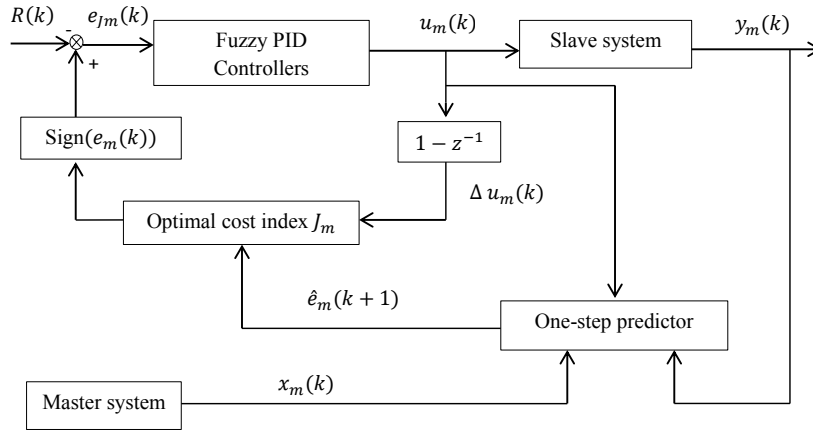


FIGURE 3.3 Schéma de régulation PID floue prédictive.

Pour obtenir les valeurs prédites des deux systèmes, nous utilisons le modèle ARX. Pour le système esclave, le modèle est donné par :

$$\begin{aligned} \hat{y}_m(k+1) = & a_1 y_m(k) + a_2 y_m(k-1) + a_3 y_m(k-2) \\ & + a_4 y_m(k-3) + b_1 u_m(k-1), \end{aligned} \quad (3.22)$$

tandis que le modèle du maître il est donnée par :

$$\begin{aligned} \hat{x}_m(k+1) = & a_1 x_m(k) + a_2 x_m(k-1) + a_3 x_m(k-2) \\ & + a_4 x_m(k-3), \end{aligned} \quad (3.23)$$

où $x_m(k)$, $y_m(k)$ et $u_m(k)$ sont les sorties du système maître, les sorties du système esclave et les entrées de commande respectivement ; a_1, a_2, a_3, a_4, b_1 sont des paramètres constants le prédicteur à un pas d'avance des états d'erreur est donné par :

$$\begin{aligned} \hat{e}_m(k+1) = & a_1 e_m(k) + a_2 e_m(k-1) + a_3 e_m(k-2) \\ & + a_4 e_m(k-3) + b_1 u_m(k-1). \end{aligned} \quad (3.24)$$

Pour l'optimisation, nous utilisons l'algorithme des essaims de particules ou PSO [120, 121], Dans cet algorithme, l'essaim est formé d'un certain nombre de particules. Chaque particule a une mémoire, et le mouvement d'une particule dans la zone de recherche est donné par une valeur de décalage, appelée vitesse qui est liée à la mémoire et la position de

la meilleure particule dans l'essaim.

L'essaim de particules H est définie comme

$$S = \{s_1, s_2, \dots, s_H\},$$

et chacun est donnée par :

$$s_w = (s_{w1}, s_{w2}, \dots, s_{wh})^T, \quad w = 1, 2, \dots, H, \quad q = 1, 2, \dots, h.$$

Les particules se déplacent à l'intérieur de l'espace de recherche en utilisant un changement de position, appelée vitesse :

$$v_w = (v_{w1}, v_{w2}, \dots, v_{wh})^T.$$

Pendant le processus de recherche, chaque particule stocke la meilleure position qu'il n'a jamais visité dans un mémoire :

$$P = \{p_1, p_2, \dots, p_H\},$$

où

$$p_w = (p_{w1}, p_{w2}, \dots, p_{wh})^T.$$

La vitesse de chaque particule est donnée comme suit :

$$v_{wq}(l+1) = v_{wq}(l) + c_1 R_1 (p_{wq} - s_{wq}(l)) + c_2 R_2 (p_{gq}(l) - s_{wq}(l)), \quad (3.25)$$

$$s_{wq}(l+1) = s_{wq}(l) + v_{wq}, \quad (3.26)$$

où R_1, R_2 sont des variables aléatoires entre $[0,1]$; c_1 et c_2 sont des facteurs de pondération; g est l'indice de la meilleure position dans la mémoire P . L'algorithme PSO peut être résumé aux étapes suivantes (1) :

L'algorithme PSO est utilisé pour ajuster hors ligne des paramètres du prédicteur, dans lequel les particules sont définies par :

$$s_w = (a_{w1}, a_{w2}, a_{w3}, a_{w4}, b_{w1})^T,$$

L'aptitude de chaque particule de l'essaim est obtenue en utilisant l'algorithme de remise en

Algorithm 1 PSO algorithm.

Step 1. mettre $l = 1$.

Step 2. Initial un essaim aléatoire S de particules dans l'espace de recherche, et définir $P = S$.

Step 3. Évaluer la fonction de coût de chaque particule de S et P , et obtenir l'indice g de la meilleure position.

Step 4. while $l < OI$ **do**

Step 5. Mise à jour de S en utilisant (3.25) et (3.26).

Step 6. Evaluate S .

Step 7. Mettre à jour P et redéfinir l'indice g .

Step 8. $l = l + 1$;

Step 9. end while.

Step 10. Print meilleure position trouvée.

forme suivant :

Algorithm 2 Fitness algorithm.

Step 1. Set $k = 1$.

Step 2. While($k \leq I$) **do**

Step 3. Calculer la sortie du prédicteur du système esclave en utilisant (3.22).

Step 4. Calculer la valeur de fonction de coût en utilisant (3.21).

Step 5. Obtenir $u(k)$ en utilisant un régulateur.

Step 6. Calculer la sortie du système asservi.

Step 7. Calculer l'erreur $E(k) = |y_m(k) - \hat{y}_m(k)|$.

Step 8. $k = k + 1$;

Step 9. end while.

I est le nombre d'itérations. L'aptitude d'une particule est obtenue en calculant la valeur maximale de E dans le cas permanent.

3.3.2 Contrôleur PI+D flou

Le régulateur PI+D flou c'est la somme de deux contrôleurs [96, 97] : un contrôleur PI flou et un contrôleur D flous. Le régulateur PI flou est défini comme suit :

$$\begin{cases} u_{PI_m}(t) &= K_{pm}e_{J_m}(t) + K_{im} \int e_{J_m}(t)dt \\ e_{J_m}(t) &= J_m(t) \times \text{Sign}(e_m(t)) - R_m(t), \end{cases} \quad (3.27)$$

où $R_m(t)$ est la référence pour la fonction de coût ; K_{pm} est le gain proportionnel ; K_{im} est le gain intégral. $e_{J_m}(t)$ et $e_m(t)$ sont le signal d'erreur de l'indice optimal J_m et l'erreur entre le maître et le système esclave, respectivement.

le régulateur PI analogique est donnée dans le domaine fréquentiel S de la façon suivante :

$$u_{PI_m}(s) = (K_{pm}^c + \frac{K_{im}^c}{s})E_{J_m}(s). \quad (3.28)$$

Pour obtenir la version numérique, la transformée bilinéaire est appliquée $s = (2/T)(z - 1)/(z + 1)$, où $T > 0$ est le temps d'échantillonnage, ce qui conduit à la forme suivante :

$$u_{PI_m}(z) = (K_{pm}^c - \frac{K_{im}^c T}{2} + \frac{K_{im}^c T}{1 - z^{-1}})E_{J_m}(z). \quad (3.29)$$

Si on pose :

$$K_{pm} = K_{pm}^c - \frac{K_{im}^c T}{2} \quad \text{et} \quad K_{im} = K_{im}^c T$$

et en utilisant la transformée z inverse , nous obtenons la forme numérique du contrôleur :

$$u_{PI_m}(kT) - u_{PI_m}(kT - T) = K_{pm}[e_{J_m}(kT) - e_{J_m}(kT - T)] + K_{im}e_{J_m}(kT). \quad (3.30)$$

En divisant (3.30) par T , nous obtenons

$$\Delta u_{PI_m}(kT) = K_{pm}e_{vm}(kT) + K_{im}e_{pm}(kT), \quad (3.31)$$

où

$$\Delta u_{PI_m}(kT) = \frac{u_{PI_m}(kT) - u_{PI_m}(kT - T)}{T}, \quad (3.32)$$

$$e_{vm}(kT) = \frac{e_{J_m}(kT) - e_{J_m}(kT - T)}{T}, \quad (3.33)$$

$$e_{pm}(kT) = e_{Jm}(kT), \quad (3.34)$$

$\Delta u_{PI_m}(kT)$ est l'incrément de commande du régulateur PI, $e_{pm}(kT)$ l'erreur entre le maître et l'esclave, et $e_{vm}(kT)$ est le taux d'erreur. L'équation (3.32) peut être écrit comme suit :

$$u_{PI_m}(kT) = u_{PI_m}(kT - T) + T\Delta u_{PI_m}(kT). \quad (3.35)$$

Pour obtenir le régulateur PI flou, l'incrément de commande $T\Delta u_{PI_m}(kT)$ est remplacé par un terme de commande floue $K_{uPI_m}\Delta u_{PI_m}(kT)$, de sorte que :

$$u_{PI_m}(kT) = u_{PI_m}(kT - T) + K_{uPI_m}\Delta u_{PI_m}(kT), \quad (3.36)$$

où K_{uPI_m} est un gain de commande flou.

La deuxième partie du contrôleur flou PI+D est dérivée du contrôleur classique D défini dans le domaine fréquentiel s , comme suit :

$$u_{Dm}(s) = sK_{dm}^c Y_m(s), \quad (3.37)$$

où $Y_m(s)$ sont les sorties du système esclave et K_{dm}^c est le gain de commande.

En utilisant la transformation bilinéaire on obtient : :

$$u_{Dm}(z) = K_{dm}^c \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} Y_m(z), \quad (3.38)$$

Ensuite, en prenant les transformées en z inverses, nous obtenons la version discrète du contrôleur D

$$u_{Dm}(kT) + u_{Dm}(kT - T) = \frac{2K_{dm}^c}{T} [y_m(kT) - y_m(kT - T)]. \quad (3.39)$$

Si on pose :

$$K_{dm} = \frac{2K_{dm}^c}{T},$$

puis en divisant (3.39) par T , on obtient :

$$\Delta u_{Dm}(kT) = K_{dm}\Delta y_m(kT), \quad (3.40)$$

où

$$\Delta u_{Dm}(kT) = \frac{u_{Dm}(kT) + u_{Dm}(kT - T)}{T}, \quad (3.41)$$

et

$$\Delta y_m(kT) = \frac{y_m(kT) - y_m(kT - T)}{T}, \quad (3.42)$$

On remarque que dans l'expression définie dans (3.40), il n'y a qu'une seule entrée du contrôleur D, $\Delta y_m(kT)$, ce qui ne suffit pas pour donner la bonne information sur la position de la sortie dans la conception des règles floues (en dessous de la valeur de la consigne ou au-dessus). Par conséquent, un autre signal doit être utilisé dans (3.40), qui devient

$$\Delta u_{Dm}(kT) = K_{dm}\Delta y_m(kT) + K_m y_{dm}(kT), \quad (3.43)$$

où $K_m = 1$ et $y_{dm}(kT) = -e_{Jm}(kT)$.

Ainsi,

$$u_{Dm}(kT) = -u_{Dm}(kT - T) + T\Delta u_{Dm}(kT). \quad (3.44)$$

Afin d'obtenir le contrôleur D floue, le terme $T\Delta u_{Dm}(kT)$ est remplacé par $K_{uDm}\Delta u_{Dm}(kT)$

$$u_{Dm}(kT) = -u_{Dm}(kT - T) + K_{uDm}\Delta u_{Dm}(kT). \quad (3.45)$$

Le contrôleur PI+D floue est donc une combinaison des deux contrôleurs : le contrôleur flou PI (3.36) et le contrôleur flou D (3.45). Par conséquent, le contrôleur général PI+D floue est donnée par :

$$\begin{aligned} u_{PIDm} &= u_{PIm}(kT) - u_{Dm}(kT) \\ &= u_{PIm}(kT - T) + K_{uPIm}\Delta u_{PIm}(kT) + u_{Dm}(kT - T) \\ &\quad - K_{uDm}\Delta u_{Dm}(kT). \end{aligned} \quad (3.46)$$

Les entrées du régulateur PI+D floue sont l'erreur $e_{pm}(kT)$, le taux d'erreur $e_{vm}(kT)$, le taux de variation de la sortie $\Delta y_m(kT)$ et $y_{dm}(kT)$. D'autre part, le régulateur PI+D flou a une seule sortie : la sortie de commande $u_{PIDm}(kT)$, qui est utilisée comme entrée pour le système esclave.

La conception du contrôleur flou PI+D comporte sur trois parties : un module de fuzzification, un module de déduction d'une base des règles de connaissances et un module de

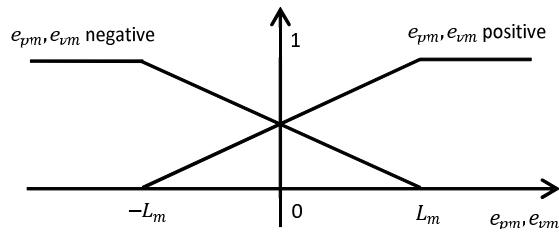


FIGURE 3.4 Fonctions d'appartenance d'entrée PI.

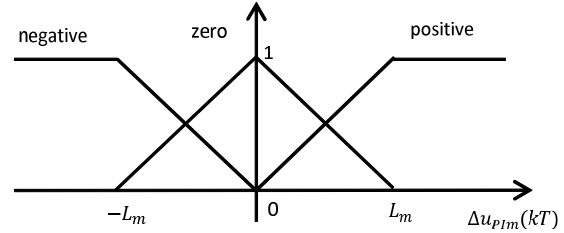


FIGURE 3.5 Fonctions d'appartenance de sortie PI.

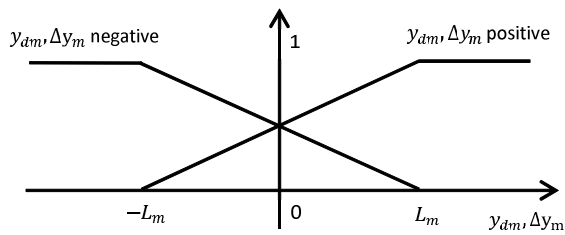


FIGURE 3.6 Fonctions d'appartenance d'entrée D.

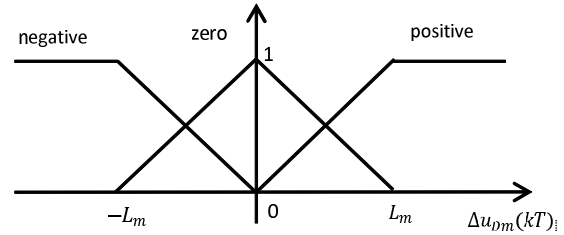


FIGURE 3.7 Fonctions d'appartenance de sortie D.

defuzzification.

Fig. 3.4 et Fig. 3.6 donnent les fonctions d'appartenance pour la fuzzyfication des entrées, alors que Fig. 3.5 et Fig. 3.7 donnent les fonctions d'appartenance des sorties.

La stabilité globale peut être atteinte en ajustant les règles de commande floue des deux contrôleur séparément. Les règles de commande floue sont attribuées en fonction de la structure du régulateur et la position des sorties de comparaison des trajectoires de référence.

En utilisant les fonctions d'appartenance ci-dessus, les règles suivantes peuvent être attribuées pour le régulateur PI flou :

- (R1) IF $e_{pm} = e_{pm}.n$ AND $e_{vm} = e_{vm}.n$ THEN PI-output= $o.n$,
- (R2) IF $e_{pm} = e_{pm}.n$ AND $e_{vm} = e_{vm}.p$ THEN PI-output= $o.z$,
- (R3) IF $e_{pm} = e_{pm}.p$ AND $e_{vm} = e_{vm}.n$ THEN PI-output= $o.z$,
- (R4) IF $e_{pm} = e_{pm}.p$ AND $e_{vm} = e_{vm}.p$ THEN PI-output= $o.p$,

tandis que pour le régulateur D flou, les règles sont donnés comme suit :

- (R5) IF $y_{dm} = y_{dm}.p$ AND $\Delta y_{vm} = \Delta y_{vm}.p$ THEN D-output= $o.z$,
- (R6) IF $y_{dm} = y_{dm}.p$ AND $\Delta y_{vm} = \Delta y_{vm}.n$ THEN D-output= $o.p$,
- (R7) IF $y_{dm} = y_{dm}.n$ AND $\Delta y_{vm} = \Delta y_{vm}.p$ THEN D-output= $o.n$,

(R8) IF $y_{dm} = y_{dm} \cdot n$ AND $\Delta y_{vm} = \Delta y_{vm} \cdot n$ THEN D-output = $o.z$.

D'après les règles, l'action de contrôle, qui vient du PI+D flou, alterne entre la sortie du régulateur PI flou et du D flou. Plus précisément, lorsque le contrôle du PI flou obtient une action, le D flou doit mettre à zéro et vice versa.

De (3.27), on peut noter que seul les signes des erreurs e_m sont utilisées afin de localiser la position des sorties (en-dessous ou au-dessus des trajectoires de référence), la variation de la fonction coût ne peut pas être utilisée car elle est toujours positive.

Nous supposons que pour chaque étape x_m est constante. Ainsi, $e_{pm} = x_m - y_m$ et $e_{vm} = \dot{e}_{pm} = 0 - \dot{y}_m$ sont utilisées pour concevoir les règles.

Par exemple, si e_{pm} est négatif ($e_{pm} \cdot n$) signifie que l'erreur est supérieure à la consigne, et si le taux d'erreur est négatif ($e_{pm} \cdot n$) cela implique que le contrôleur à l'étape précédente a conduit la sortie du système vers le haut ; par conséquent la sortie de la commande, $\Delta u_{pim}(kT)$, doit être réglée pour être négative (R1), et la sortie du contrôleur D flou doit être fixé à zéro (R5). D'autre part, si e_{pm} est positif ($e_{pm} \cdot n$) signifie que l'erreur est supérieure à la consigne, et si le taux d'erreur est positif ($e_{pm} \cdot n$) cela implique que le contrôleur à l'étape précédente a conduit la sortie du système vers le bas ; donc la sortie de la commande, $\Delta u_{pim}(kT)$, doit être réglé à zéro (R2) et le rôle du contrôleur D flou est d'ajouter une valeur positive afin de tirer la sortie vers le bas plus rapidement (R6).

Le reste des règles peut être interprété de la même manière. Dans l'étape de défuzzification, la formule de calcul du centre de masse est utilisée pour obtenir les incréments de commande à la fois pour les contrôleurs PI et D flous :

$$\Delta u_m = \frac{\sum \text{MVI} \times \text{MVO}}{\sum \text{membership value of input}}, \quad (3.47)$$

où MVI est la valeur d'appartenance d'entrée et MVO de la sortie correspondant à la valeur d'appartenance de l'entrée.

Les intersections entre les entrées sont divisées en 20 combinaisons de régions d'entrée adjacentes (IC), comme le montre Fig. 3.8 et Fig. 3.9.

Chaque région sélectionnée a des conditions spécifiques ; par exemple, dans la région IC 1, les conditions sont : $0 < K_{im} \cdot e_{pm}(kT) < L_m$, $-L_m < K_{pm} \cdot e_{vm}(kT) < 0$ and $K_{pm} \cdot e_{vm}(kT) + K_{im} \cdot e_{pm}(kT) > 0$. En utilisant les règles, la formule de défuzzification et les équations suivantes :

$$e_{pm} \cdot p = \frac{K_{im} e_{pm}(kT) + L_m}{2L_m}, \quad e_{pm} \cdot n = \frac{-K_{im} e_{pm}(kT) + L_m}{2L_m},$$

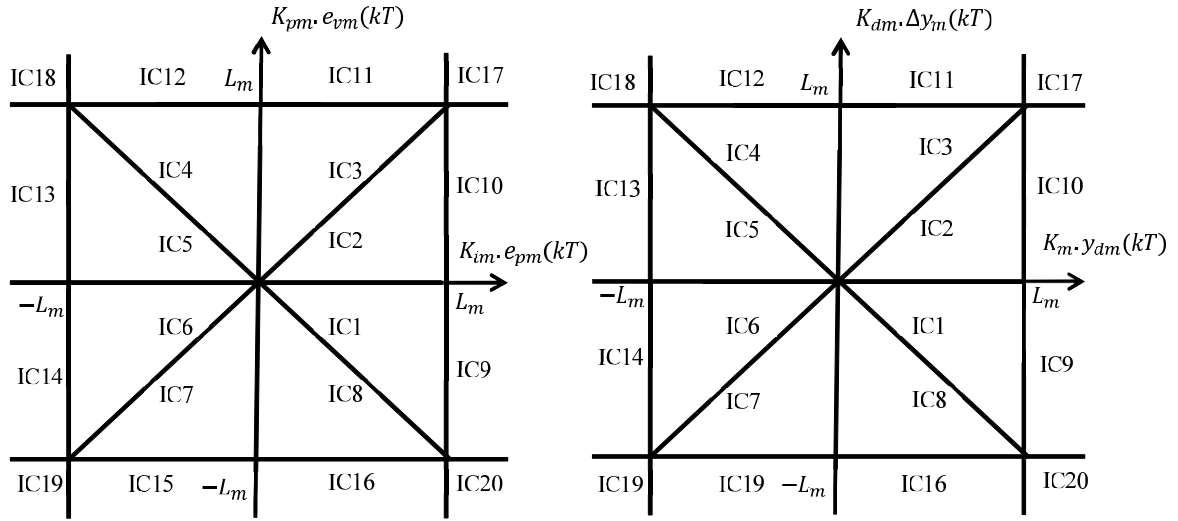


FIGURE 3.8 Régions du régulateur PI floue. FIGURE 3.9 Régions du contrôleur D floue.

$$e_{vm} \cdot p = \frac{K_{pm} e_{vm}(kT) + L_m}{2L_m}, \quad e_{vm} \cdot n = \frac{-K_{pm} e_{vm}(kT) + L_m}{2L_m},$$

$$y_{dm} \cdot p = \frac{K y_{dm}(kT) + L_m}{2L_m}, \quad y_{dm} \cdot n = \frac{-K y_{dm}(kT) + L_m}{2L_m},$$

$$\Delta y_m \cdot p = \frac{K_{dm} \Delta y_m(kT) + L_m}{2L_m}, \quad \Delta y_m \cdot n = \frac{-K_{dm} \Delta y_m(kT) + L_m}{2L_m},$$

On obtient les formules de defuzzification pour l'ensemble des 20 régions du PI flou :

$$\Delta u_{PI_m}(kT) = \frac{L_m [K_{im} \cdot e_{pm}(kT) + K_{pm} \cdot e_{vm}(kT)]}{2(2L_m - K_{im} \cdot |e_{pm}(kT)|)}, \text{ (in IC 1,2,5,6)} \quad (3.48)$$

$$\Delta u_{PI_m}(kT) = \frac{L_m [K_{im} \cdot e_{pm}(kT) + K_{pm} \cdot e_{vm}(kT)]}{2(2L_m - K_{pm} \cdot |e_{vm}(kT)|)}, \text{ (in IC 3,4,7,8)} \quad (3.49)$$

$$\Delta u_{PI_m}(kT) = 1/2 [K_{pm} e_{vm}(kT) + L_m], \text{ (in IC 9,10)} \quad (3.50)$$

$$\Delta u_{PI_m}(kT) = 1/2 [K_{im} e_{pm}(kT) + L_m], \text{ (in IC 11,12)} \quad (3.51)$$

$$\Delta u_{PI_m}(kT) = 1/2 [K_{pm} e_{vm}(kT) - L_m], \text{ (in IC 13,14)} \quad (3.52)$$

$$\Delta u_{PI_m}(kT) = 1/2[K_{im}e_{pm}(kT) - L_m], (\text{in IC 15,16}) \quad (3.53)$$

$$\Delta u_{PI_m}(kT) = 0, (\text{in IC 18,20}) \quad (3.54)$$

$$\Delta u_{PI_m}(kT) = L_m, (\text{in IC 17}) \quad (3.55)$$

$$\Delta u_{PI_m}(kT) = -L_m, (\text{in IC 19}) \quad (3.56)$$

Et que pour le D flou, on obtient les formules suivantes :

$$\Delta u_{Dm}(kT) = \frac{L_m[K_m \cdot y_{dm}(kT) - K_{dm} \cdot \Delta y_m(kT)]}{2(2L_m - K_m \cdot |y_{dm}(kT)|)}, (\text{in IC 1,2,5,6}) \quad (3.57)$$

$$\Delta u_{Dm}(kT) = \frac{L_m[K_m \cdot y_{dm}(kT) - K_{dm} \cdot \Delta y_m(kT)]}{2(2L_m - K_{dm} \cdot |\Delta y_m(kT)|)}, (\text{in IC 3,4,7,8}) \quad (3.58)$$

$$\Delta u_{Dm}(kT) = 1/2[-K_{dm}\Delta y_m(kT) + L_m], (\text{in IC 9,10}) \quad (3.59)$$

$$\Delta u_{Dm}(kT) = 1/2[K_m y_{dm}(kT) - L_m], (\text{in IC 11,12}) \quad (3.60)$$

$$\Delta u_{Dm}(kT) = 1/2[-K_{dm}\Delta y_m(kT) - L_m], (\text{in IC 13,14}) \quad (3.61)$$

$$\Delta u_{Dm}(kT) = 1/2[K_m y_{dm}(kT) + L_m], (\text{in IC 15,16}) \quad (3.62)$$

$$\Delta u_{Dm}(kT) = 0, (\text{in IC 17,19}) \quad (3.63)$$

$$\Delta u_{Dm}(kT) = -L_m, (\text{in IC 18}) \quad (3.64)$$

$$\Delta u_{Dm}(kT) = L_m, (\text{in IC 20}) \quad (3.65)$$

3.3.3 Contrôleur PD+I flou

La conception du contrôleur PD+I flou passe par les mêmes étapes que le PI+D flou[122],[119]. La partie principale du noyau flou est la base de règles de contrôle, qui est conçu en fonction de la structure du contrôleur, et les positions des entrées. Dans cette section, nous résumons les principales étapes de la conception du contrôleur PD+I flou.

La forme du contrôleur classique PD+I dans le -domaine fréquentiel s est :

$$u_{PIDm}(s) = u_{PDm}(s) + u_{Im}(s),$$

où

$$u_{PDm}(s) = (K_{pm}^c + sK_{dm}^c)E_{Jm}(s),$$

et

$$u_{Im}(s) = \frac{K_{im}^c}{s}E_{Jm}(s) \implies u_{PDm}(z) = (K_{pm} + K_{dm} \frac{1-z^{-1}}{1+z^{-1}})E_{Jm}(z),$$

et

$$u_{Im}(z) = K_{im} \frac{T}{2} \frac{1+z^{-1}}{1-z^{-1}}; K_{pm} = K_{pm}^c, K_{dm} = \frac{2}{T}K_{dm}^c, K_{im} = K_{im}^c.$$

Les formes discrètes des contrôleurs PD et I sont donnés à l'aide des transformée en z inverse de la manière suivante :

$$\Delta u_{PDm}(kT) = K_{pm}d_m(k) + K_{dm}r_m(k), \quad (3.66)$$

$$\Delta u_{Im}(kT) = K_m r_m(k) + K_{im}e_{Jm}(k-1), \quad (3.67)$$

où

$$r_m(kT) = \frac{e_{Jm}(k) - e_{Jm}(kT - T)}{T}, \quad (3.68)$$

$$d_m(kT) = \frac{e_{Jm}(kT) + e_{Jm}(kT - T)}{T}, \quad (3.69)$$

$$\Delta u_{PDm}(kT) = \frac{u_{PDm}(kT) + u_{PDm}(kT - T)}{T}, \quad (3.70)$$

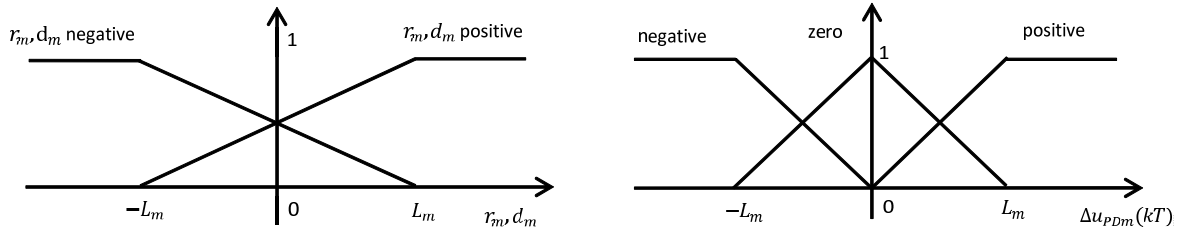


FIGURE 3.10 Fonctions d'appartenance d'entrée PD. FIGURE 3.11 Fonctions d'appartenance de sortie PD.

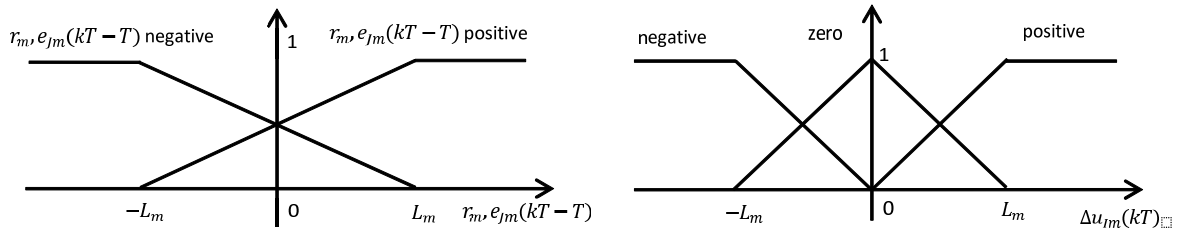


FIGURE 3.12 Fonctions d'appartenance d'entrée I. FIGURE 3.13 Fonctions d'appartenance de sortie PD.

$$\Delta u_{Im}(kT) = \frac{u_{Im}(k) - u_{Im}(kT - T)}{T}. \quad (3.71)$$

Les équations (3.66) et (3.67) peut être réécrite comme suit :

$$u_{PDm}(kT) = -u_{PDm}(kT - T) + T\Delta u_{PDm}(kT), \quad (3.72)$$

$$u_{Im}(kT) = u_{Im}(kT - T) + T\Delta u_{Im}(kT). \quad (3.73)$$

Pour obtenir la version flou des deux contrôleurs, les incréments de commande $T\Delta u_{Im}$ et $T\Delta u_{PDm}$, sont remplacés par des termes flous, (3.72) et (3.73) deviennent :

$$u_{PDm}(kT) = -u_{PDm}(kT - T) + K_{uPDm}\Delta u_{PDm}(kT), \quad (3.74)$$

$$u_{Im}(kT) = u_{Im}(kT - T) + K_{uIm}\Delta u_{Im}(kT). \quad (3.75)$$

Fig. 3.10 et Fig. 3.12 représentent les fonctions d'appartenance pour la fuzzification des entrées, alors que Fig. 3.11 et Fig. 3.13 représentent les fonctions d'appartenance pour les sorties.

En utilisant les fonctions d'appartenance ci-dessus, les règles suivantes peuvent être at-

tribuées pour le contrôleur PD flou :

- (R1) IF $d_m = d_m.n$ AND $r_m = r_m.n$ THEN PD-output= $o.z$,
- (R2) IF $d_m = d_m.n$ AND $r_m = r_m.p$ THEN PD-output= $o.n$,
- (R3) IF $d_m = d_m.p$ AND $r_m = r_m.n$ THEN PD-output= $o.p$,
- (R4) IF $d_m = d_m.p$ AND $r_m = r_m.p$ THEN PD-output= $o.z$,

tandis que les règles du contrôleur flou I sont données comme suit :

- (R5) IF $e_{Jm}(kT - T) = e_{Jm.p}$ AND $r_m = r_m.p$ THEN I-output= $o.p$,
- (R6) IF $e_{Jm}(kT - T) = e_{Jm.p}$ AND $r_m = r_m.n$ THEN I-output= $o.z$,
- (R7) IF $e_{Jm}(kT - T) = e_{Jm.n}$ AND $r_m = r_m.p$ THEN I-output= $o.z$,
- (R8) IF $e_{Jm}(kT - T) = e_{Jm.n}$ AND $r_m = r_m.n$ THEN I-output= $o.n$.

Vingt régions adjacentes (IC) sont utilisées pour la défuzzification, comme représenté sur Fig. 3.14 et Fig. 3.15. En utilisant les règles, la formule de défuzzification (3.47), et les équations suivantes :

$$d_{m.p} = \frac{K_{pm}d_m(kT) + L_m}{2L_m}, \quad d_{m.n} = \frac{-K_{pm}d_m(kT - T) + L_m}{2L_m},$$

$$r_{m.p} = \frac{K_{dm}r_m(kT) + L_m}{2L_m}, \quad r_{m.n} = \frac{-K_{dm}r_m(kT) + L_m}{2L_m},$$

$$e_{Jm.p} = \frac{K_{im}e_{Jm}(kT - T) + L_m}{2L_m}, \quad e_{Jm.n} = \frac{-K_{im}e_{Jm}(kT - T) + L_m}{2L_m},$$

$$r_{m.p} = \frac{K_m r_m(kT) + L_m}{2L_m}, \quad r_{m.n} = \frac{-K_m r_m(kT) + L_m}{2L_m},$$

Nous obtenons les formules de defuzzification pour l'ensemble des 20 régions du PD flou :

$$\Delta u_{PDm}(kT) = \frac{L_m[K_{pm} \cdot d_m(kT) - K_{dm} \cdot r_m(kT)]}{2(2L_m - K_{pm} \cdot |d_m(kT)|)}, \text{ (in IC 1,2,5,6)} \quad (3.76)$$

$$\Delta u_{PDm}(kT) = \frac{L_m[K_{pm} \cdot d_m(kT) - K_{dm} \cdot r_m(kT)]}{2(2L_m - K_{dm} \cdot |r_m(kT)|)}, \text{ (in IC 3,4,7,8)} \quad (3.77)$$

$$\Delta u_{PDm}(kT) = 1/2[-K_{dm}r_m(kT) + L_m], \text{ (in IC 9,10)} \quad (3.78)$$

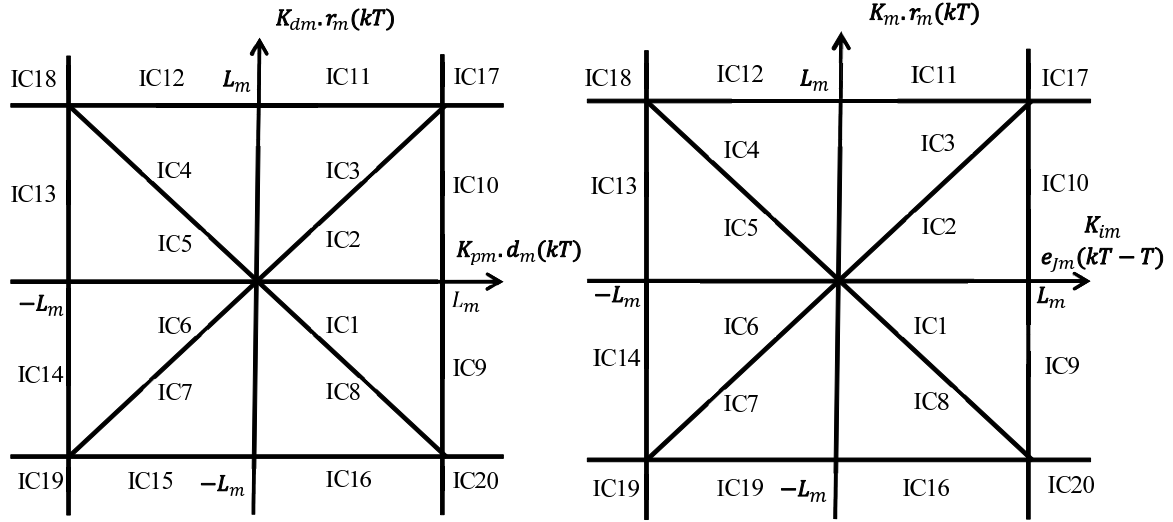


FIGURE 3.14 Régions du régulateur PD floue. FIGURE 3.15 Régions du contrôleur I flou.

$$\Delta u_{PDm}(kT) = 1/2[K_{pm}d_m(kT) - L_m], (\text{in IC 11,12}) \quad (3.79)$$

$$\Delta u_{PDm}(kT) = 1/2[-K_{dm}r_m(kT) - L_m], (\text{in IC 13,14}) \quad (3.80)$$

$$\Delta u_{PDm}(kT) = 1/2[K_{pm}d_m(kT) + L_m], (\text{in IC 15,16}) \quad (3.81)$$

$$\Delta u_{PDm}(kT) = 0, (\text{in IC 19,17}) \quad (3.82)$$

$$\Delta u_{PDm}(kT) = -L_m, (\text{in IC 18}) \quad (3.83)$$

$$\Delta u_{PDm}(kT) = L_m, (\text{in IC 20}) \quad (3.84)$$

tandis que pour la commande I flou, on obtient les formules suivantes :

$$\Delta u_{Im}(kT) = \frac{L_m[K_{im} \cdot e_{Jm}(kT - T) + K_m \cdot r_m(kT)]}{2(2L_m - K_{im} \cdot |e_{Jm}(kT - T)|)}, (\text{in IC 1,2,5,6}) \quad (3.85)$$

$$\Delta u_{Im}(kT) = \frac{L_m[K_{im} \cdot e_{Jm}(kT - T) + K_m \cdot r_m(kT)]}{2(2L_m - K_m \cdot |r_m(kT)|)}, (\text{in IC 3,4,7,8}) \quad (3.86)$$

$$\Delta u_{Im}(kT) = 1/2[K_m r_m(kT) + L_m], (\text{in IC 9,10}) \quad (3.87)$$

$$\Delta u_{Im}(kT) = 1/2[K_{im} e_{Jm}(kT - T) + L_m], (\text{in IC 11,12}) \quad (3.88)$$

$$\Delta u_{Im}(kT) = 1/2[K_m r_m(kT) - L_m], (\text{in IC 13,14}) \quad (3.89)$$

$$\Delta u_{Im}(kT) = 1/2[K_{im} e_{Jm}(kT - T) - L_m], (\text{in IC 15,16}) \quad (3.90)$$

$$\Delta u_{Im}(kT) = 0, (\text{in IC 20,18}) \quad (3.91)$$

$$\Delta u_{Im}(kT) = -L_m, (\text{in IC 19}) \quad (3.92)$$

$$\Delta u_{Im}(kT) = L_m. (\text{in IC 17}) \quad (3.93)$$

3.3.4 Analyse de stabilité

L'état général de la stabilité peut être obtenu en utilisant la deuxième méthode de Lyapunov [119]. La fonction de Lyapunov est choisie comme suit :

$$V = e_{Jm}^2/2 > 0, \quad (3.94)$$

$e_{Jm} = J_m(k)$. La dérivation en temps de V est donnée par :

$$\begin{aligned} \dot{V} = e_{Jm} \dot{e}_{Jm} \approx J_m(k) [J_m(k) - J_m(k-1)] = \\ \left[\sum_{i=-1}^N e_m(k-i)^2 + \lambda \sum_{j=0}^{N_c} (\Delta u_m(k-j))^2 \right] \times \\ \left[\sum_{i=-1}^N e_m(k-i)^2 + \lambda \sum_{j=0}^{N_c} (\Delta u_m(k-j))^2 - \right. \\ \left. \sum_{i=-1}^N e_m(k-1-i)^2 - \lambda \sum_{j=0}^{N_c} (\Delta u_m(k-1-j))^2 \right] \end{aligned}$$

$$= \left[\sum_{i=-1}^N e_m(k-i)^2 + \lambda \sum_{j=0}^{N_c} (\Delta u_m(k-j))^2 \right] \times \\ [e_m(k+1) - e_m(k-1-N) + \Delta u_m(k) - \Delta u_m(k-1-N_c)].$$

Ainsi, si les deux conditions $e_m(k) - e_m(k-1) < 0$ et $\Delta u_m(k) - \Delta u_m(k-1) < 0$ sont vérifiées à chaque étape k alors \dot{V} sera négative. Les conditions de stabilité du PI+D flou sont données comme suit :

$$\begin{aligned} C_1 &= \dot{e}_m \approx e_m(k) - e_m(k-1) \\ &= g_m(k) - f_m(k) + u_m(k) \\ &= g_m(k) - f_m(k) + K_{uPIm} [K_{im} e_{pm}(kT) \\ &\quad + K_{pm} \frac{J_m(kT) - J_m(kT-T)}{T}] \\ &\quad - K_{uDm} \left[-J_m(kT) + K_{dm} \frac{y_m(kT) - y_m(kT-T)}{T} \right] + u_{PIm}(kT-T) \\ &\quad + u_{Dm}(kT-T) < 0, \end{aligned} \tag{3.95}$$

et

$$\begin{aligned} C_2 &= \Delta u_m(kT) - \Delta u_m(kT-T) \\ &= u_m(k) - 2u_m(k-1) + u_m(k-2) \\ &= K_{uPIm} K_{im} [J_m(kT) - 2J_m(kT-T) + J_m(kT-2T)] \\ &\quad + K_{uPIm} K_{pm} \left[\frac{J_m(kT) - 3J_m(kT-T) + 3J_m(kT-2T) - J_m(kT-3T)}{T} \right] \\ &\quad + K_{uDm} [J_m(kT) - 2J_m(kT-T) + J_m(kT-2T)] \\ &\quad + K_{uDm} K_{dm} \left[\frac{-y_m(kT) + 3y_m(kT-T) - 3y_m(kT-2T) + y_m(kT-3T)}{T} \right] \\ &\quad + u_{PIm}(kT-T) + u_{Dm}(kT-T) \\ &\quad - 2u_{PIm}(kT-2T) - 2u_{Dm}(kT-2T) \\ &\quad + u_{PIm}(kT-3T) + u_{Dm}(kT-3T) < 0, \end{aligned} \tag{3.96}$$

alors que les conditions de stabilité du PD+I flou sont données comme suit :

$$\begin{aligned}
C_3 &= \dot{e}_m \approx e_m(k) - e_m(k-1) \\
&= g_m(k) - f_m(k) + u_m(k) \\
&= g_m(k) - f_m(k) \\
&+ K_{uPDm} \left[K_{pm} \frac{J_m(kT) + J_m(kT-T)}{T} + K_{dm} \frac{J_m(kT) - J_m(kT-T)}{T} \right] \\
&+ K_{uIm} \left[K_{im} \frac{J_m(kT) - J_m(kT-T)}{T} + K_{im} J_m(kT-T) \right] - u_{PDm}(kT-T) \\
&+ u_{Im}(kT-T) < 0,
\end{aligned} \tag{3.97}$$

et

$$\begin{aligned}
C_4 &= \Delta u_m(kT) - \Delta u_m(kT-T) \\
&= u_m(k) - 2u_m(k-1) + u_m(k-2) \\
&= K_{uPDm} K_{pm} \left[\frac{J_m(kT) - J_m(kT-T) - J_m(kT-2T) + J_m(kT-3T)}{T} \right] \\
&+ K_{uPDm} K_{dm} \left[\frac{J_m(kT) - 3J_m(kT-T) + 3J_m(kT-2T) - J_m(kT-3T)}{T} \right] \\
&+ K_{uIm} K_{im} \left[\frac{J_m(kT) - 3J_m(kT-T) + 3J_m(kT-2T) - J_m(kT-3T)}{T} \right. \\
&\quad \left. + J_m(kT-T) - 2J_m(kT-2T) + J_m(kT-3T) \right] \\
&- u_{PDm}(kT-T) + 2u_{PDm}(kT-2T) - u_{PDm}(kT-3T) \\
&+ u_{Im}(kT-T) - 2u_{Im}(kT-2T) + u_{Im}(kT-3T) < 0.
\end{aligned} \tag{3.98}$$

A partir de C1 et C2, les conditions suivantes peuvent être définies :

$$\begin{aligned}
K_{im} &\prec \frac{\chi_1}{e_{pm}(kT)} = N_{1pi+d} \\
K_{im} &\prec \frac{\eta_1}{J_m(kT) - 2J_m(kT-T) + J_m(kT-2T)} = M_{1pi+d} \\
K_{pm} &\prec \frac{\chi_2 T}{J_m(kT) - J_m(kT-T)} = N_{2pi+d} \\
K_{ip} &\prec \frac{\eta_2 T}{J_m(kT) - 3J_m(kT-T) + 3J_m(kT-2T) - J_m(kT-3T)} = M_{2pi+d} \\
K_{dm} &\prec \frac{\chi_3 T + J_m(kT)}{y_m(kT) - y_m(kT-T)} = N_{3pi+d} \\
K_{dm} &\prec \frac{\eta_3 T}{-y_m(kT) + 3y_m(kT-T) - 3y_m(kT-2T) + y_m(kT-3T)} = M_{3pi+d}
\end{aligned}$$

$$\begin{aligned}
K_{uPI m} &\prec -\frac{\chi_4 + \chi_5 - \chi_3 K_{uDm} + \chi_6 + \chi_7}{\chi_1 + \chi_2} = N_{4pi+d} \\
K_{uPI m} &\prec -\frac{K_{uDm}(\eta_3 + \eta_4) + \eta_5 + \eta_6}{\eta_1 + \eta_2} = M_{4pi+d} \\
K_{uDm} &\prec \frac{K_{uPI m}(\chi_1 + \chi_2) + \chi_4 + \chi_5 + \chi_6 + \chi_7}{\chi_3} = N_{5pi+d} \\
K_{uDm} &\prec -\frac{K_{uPI m}(\eta_1 + \eta_2) + \eta_5 + \eta_6}{\eta_3 + \eta_4} = M_{5pi+d} \\
g_m(kT) &\prec \chi_4; J_m(kT) - 2J_m(kT - T) + J_m(kT - 2T) \prec \eta_4 \\
-f_m(kT) &\prec \chi_5; u_{PI m}(kT - T) - 2u_{PI m}(kT - 2T) + u_{PI m}(kT - 3T) \prec \eta_5 \\
u_{PI m}(kT - T) &\prec \chi_6; u_{dm}(kT - T) - 2u_{dm}(kT - 2T) + u_{dm}(kT - 3T) \prec \eta_6 \\
u_{dm}(kT - T) &\prec \chi_7,
\end{aligned}$$

où $\chi \prec 0$ et $\eta \prec 0$ sont constants. Par conséquent, les gains devraient être choisis comme suit :

$$\begin{aligned}
K_{im} &\prec \min(N_{1pi+d}, M_{1pi+d}) \\
K_{pm} &\prec \min(N_{2pi+d}, M_{2pi+d}) \\
K_{dm} &\prec \min(N_{3pi+d}, M_{3pi+d}) \\
K_{uPI m} &\prec \min(N_{4pi+d}, M_{4pi+d}) \\
K_{uDm} &\prec \min(N_{5pi+d}, M_{5pi+d})
\end{aligned}$$

De C3, C4, nous obtenons les conditions de stabilité de flou PD+I

$$\begin{aligned}
K_{pm} &\prec \min(N_{1pd+i}, M_{1pd+i}) \\
K_{dm} &\prec \min(N_{2pd+i}, M_{2pd+i}) \\
K_{im} &\prec \min(N_{3pd+i}, M_{3pd+i}) \\
K_{uPDm} &\prec \min(N_{4pd+i}, M_{4pd+i}) \\
K_{uIm} &\prec \min(N_{5pd+i}, M_{5pd+i}),
\end{aligned}$$

où

$$\begin{aligned}
M_{1pd+i} &= \frac{\kappa_1 T}{(J_m(kT) + J(kT - T))} \\
N_{1pd+i} &= \frac{\tau_1 T}{J_m(kT) - J(kT - T) - J_m(kT - 2T) - J_m(kT - 3T)} \\
M_{2pd+i} &= \frac{\kappa_2 T}{(J_m(kT) - J(kT - T))} \\
N_{2pd+i} &= \frac{\tau_2 T}{J_m(kT) - 3J(kT - T) + 3J_m(kT - 2T) - J_m(kT - 3T)} \\
M_{3pd+i} &= \frac{\kappa_3 T}{(J_m(kT) - J(kT - T)/T) + J_m(kT - T)}
\end{aligned}$$

$$\begin{aligned}
N_{3pd+i} &= \tau_3 / [(J_m(kT) - 3J(kT - T) + 3J_m(kT - 2T) - J_m(kT - 3T)) / T] + J_m(kT - T) - \\
&2J_m(kT - 2T) + J_m(kT - 3T), \\
M_{4pd+i} &= \frac{-(K_{ulm}\kappa_3 + \kappa_4 + \kappa_5 + \kappa_6 + \kappa_7)}{\kappa_1 + \kappa_2} \\
N_{4pd+i} &= \frac{K_{ulm}\tau_3 + \tau_4 + \tau_5}{\tau_1 + \tau_2} \\
M_{5pd+i} &= \frac{-(K_{uPDm}(\kappa_1 + \kappa_2) + \kappa_4 + \kappa_5 + \kappa_6 + \kappa_7)}{\kappa_3} \\
N_{5pd+i} &= \frac{K_{ulm}\tau_3 + \tau_4 + \tau_5}{\tau_3} \\
g_m(kT) &\prec \kappa_4 \\
-f_m(kT) &\prec \kappa_5 \\
u_{PDm}(kT - T) &\prec \kappa_6 \\
u_{Im}(kT - T) &\prec \kappa_7, \\
-u_{PDm}(kT - T) - 2u_{PDm}(kT - 2T) + u_{PDm}(kT - 3T) &\prec \tau_4 \\
u_{Im}(kT - T) - 2u_{Im}(kT - 2T) + u_{Im}(kT - 3T) &\prec \tau_5 \\
\text{où } \kappa &\prec 0 \text{ et } \tau &\prec 0 \text{ sont constants.}
\end{aligned}$$

3.3.5 Simulation numérique

Les performances de l'algorithme proposé sont vérifiées, en étudiant la synchronisation de deux systèmes de Lorenz incertaines[101][123].

Le maître est défini par :

$$\begin{cases} \dot{x}_1 = \alpha_1(x_2 - x_1) \\ \dot{x}_2 = (-x_1x_3 + \rho_1x_1 - x_2), \\ \dot{x}_3 = x_1x_2 - \beta_1x_3 \end{cases} \quad (3.99)$$

où x_1, x_2, x_3 sont les variables d'état et $\alpha_1, \rho_1, \beta_1$ sont les paramètres incertains du système.

l'esclave est défini par :

$$\begin{cases} \dot{y}_1 = \alpha_2(y_2 - y_1) + u_1 \\ \dot{y}_2 = (-y_1y_3 + \rho_2y_1 - y_2) + u_2, \\ \dot{y}_3 = y_1y_2 - \beta_2y_3 + u_3 \end{cases} \quad (3.100)$$

où y_1, y_2, y_3 sont les variables d'état, $\alpha_2, \rho_2, \beta_2$ sont les paramètres incertains du système et u_1, u_2, u_3 sont les commandes fournies par les contrôleurs.

Les erreurs de synchronisation sont définies par :

$$\begin{cases} e_1 = x_1 - y_1 \\ e_2 = x_2 - y_2, \\ e_3 = x_3 - y_3 \end{cases} \quad (3.101)$$

et leur dynamique par :

$$\begin{cases} \dot{e}_1 = -\alpha_2(y_2 - y_1) + \alpha_1(x_2 - x_1) + u_1 \\ \dot{e}_2 = -\rho_2 y_1 + y_2 + y_1 y_3 + \rho_1 x_1 - x_2 - x_1 x_3 + u_2. \\ \dot{e}_3 = -y_1 y_2 + \beta_2 y_3 + x_1 x_2 - \beta_1 x_3 + u_3 \end{cases} \quad (3.102)$$

la fonction cout est choisie sous la forme suivante :

$$J_m(k) = \sum_{i=-1}^3 [e_m(k-i)]^2 + \lambda \sum_{j=0}^3 [\Delta u_m(k-j)]^2, \quad (3.103)$$

avec $\lambda = 0.001$.

La prédiction de l'erreur est donnée par l'expression suivante :

$$\begin{aligned} e_m(k+1) &= 0.9497e_m(k) + 0.0141e_m(k-1) \\ &+ 0.6806e_m(k-2) + 0.6440e_m(k-3) \\ &+ 0.051u_m(k-1). \end{aligned} \quad (3.104)$$

L'application de la méthode présentée précédemment a donné les résultats représentés par les figures suivantes. Fig. 3.16 et Fig. 3.17 représentent la prédiction des états x_2 , y_2 , et leurs erreurs de prédiction.

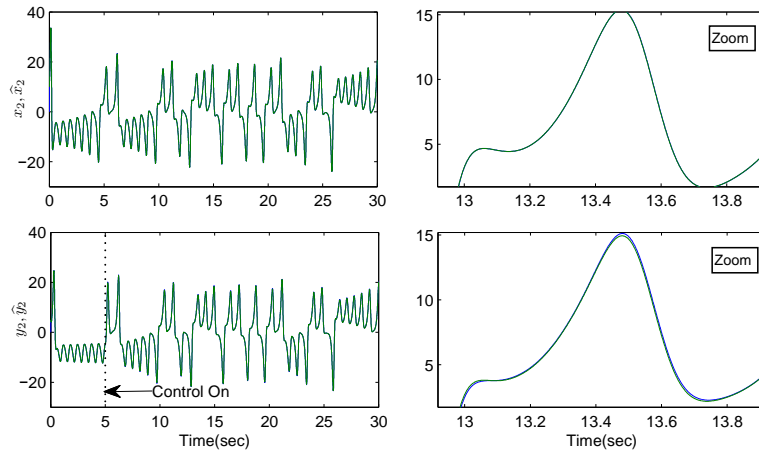
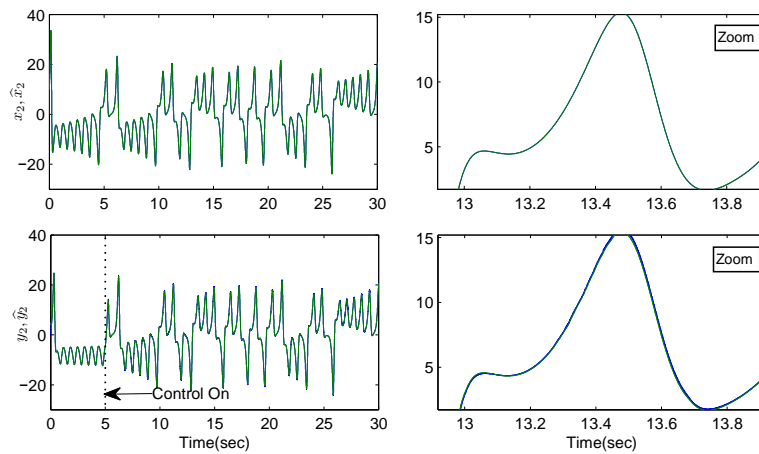
Pour la simulation numérique, les paramètres du système maître et le système esclave sont choisis comme suit :

$$\alpha_1 = 10, \rho_1 = 28, \beta_1 = 8/3, \alpha_2 = 10.5, \rho_2 = 25, \beta_2 = 8/3 + 0.2.$$

Les conditions initiales du système maître et le système esclave sont prises égales à :

$$x_1(0) = 2, x_2(0) = 10, x_3(0) = -6, y_1(0) = -2, y_2(0) = 5, y_3(0) = 1.$$

Dans la première partie de la simulation, nous présentons deux résultats obtenus à l'aide de la commande PD+I floue. Un des termes de prédiction et l'autre sans. Les paramètres de la PD+I floue sont choisis comme : $K_p = 1.014, K_d = 0.594, K_{upd} = 0.1, K = 1, K_i = 2.045, K_{ui} = 0.1, L = 30$.

FIGURE 3.16 Prédicteurs des états x_2, y_2 en utilisant le contrôleur PI+D.FIGURE 3.17 Prédicteurs des états x_2, y_2 en utilisant le contrôleur PD+I .

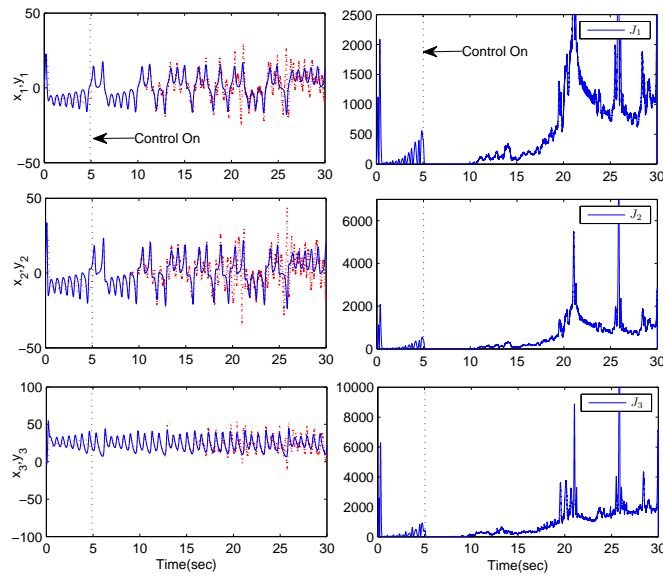


FIGURE 3.18 Synchronisation des systèmes de Lorenz et les variations des fonctions coûts sans termes de prédiction à l'aide de commande floue PD+I.

Dans la deuxième partie, le contrôleur flou PI+D est utilisé à la place du contrôleur flou PD+I. Les paramètres de la PI+D flou sont choisis comme suit : $L = 30, K_{uPI} = 1, K_I = 1, K_P = 1, K_{uD} = 0.001, K_D = 1$.

Fig. 3.18 et Fig. 3.19 montrent les résultats de la synchronisation des deux systèmes et les variations des fonctions coût sans termes de prédiction et avec termes de prédiction respectivement pour le premier cas, tandis que Fig. 3.20 et Fig. 3.21 donnent les résultats pour le second cas.

Pour le premier cas, on remarque qu'en l'absence de termes de prédiction, la synchronisation entre les deux systèmes est rompue, et les fonctions coût prennent des valeurs très importantes. Cependant, en présence des termes de prédiction, la synchronisation est réalisée et les fonctions de coût convergent vers zéro.

Pour le second cas, la synchronisation entre les deux systèmes est réalisée avec et sans termes de prédiction.

Tableau 3.1 résume tous les résultats obtenus avec les deux contrôleurs dans les cas avec et sans termes de prédiction. Dans le cas du régulateur PD+I flou, les termes de prédiction assurent la synchronisation entre les deux systèmes. Par contre, avec le régulateur PI+D flou, les termes de prédiction font du bruit et réduisent les performances du régulateur. Par ailleurs, la performance du régulateur à PI+D flou est meilleure que celle du contrôleur

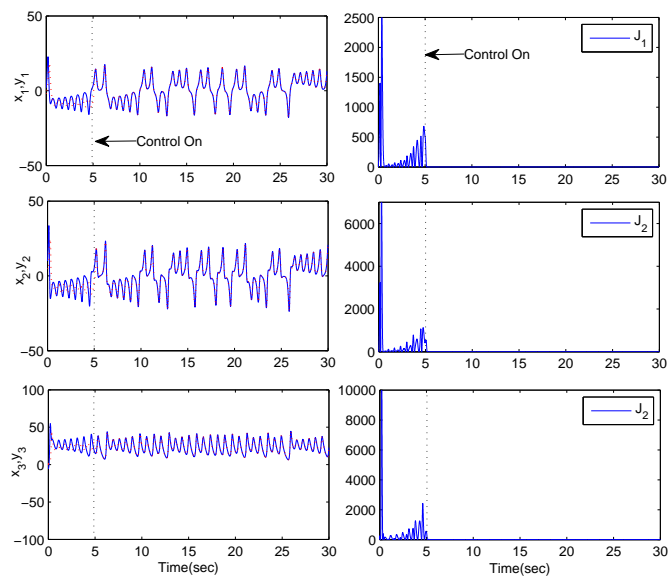


FIGURE 3.19 Synchronisation des systèmes de Lorenz et les variations des fonctions coûts avec termes de prédiction à l'aide de commande floue PD+I.

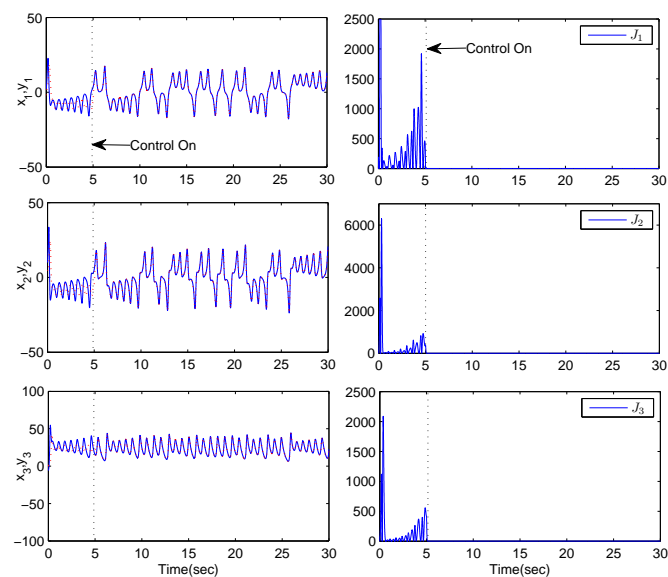


FIGURE 3.20 Synchronisation des systèmes de Lorenz et les variations des fonctions coûts sans termes de prédiction à l'aide de commande floue PI+D.

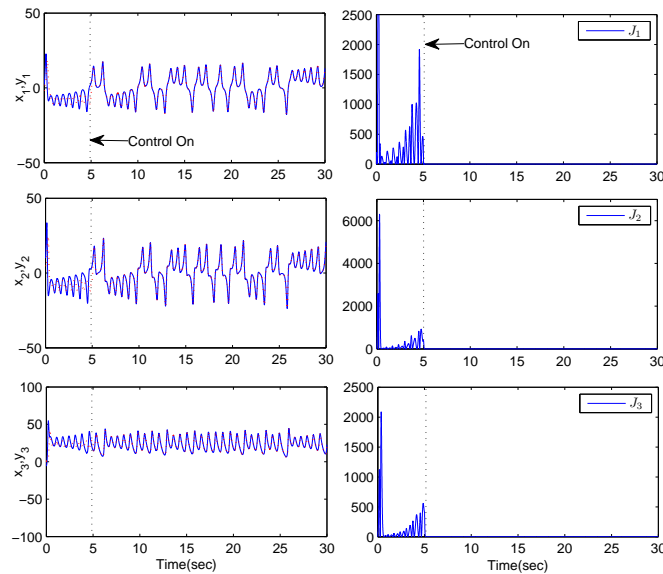


FIGURE 3.21 Synchronisation des systèmes de Lorenz et les variations des fonctions coûts avec termes de prédiction à l'aide de commande floue PI+D.

TABLE 3.1 Les erreurs de synchronisation et de la fonction de coût.

| | sans termes de prédiction | | avec termes de prédiction | |
|-----------------------|---------------------------|--------|---------------------------|--------|
| | PD+I | PI+D | PD+I | PI+D |
| $\sum_{m=1}^3 e_m$ | $1.1 \cdot 10^5$ | 60.8 | 1092.9 | 89.09 |
| $\sum_{m=1}^3 e_{Jm}$ | $2.15 \cdot 10^7$ | 285.11 | 6486.7 | 3699.6 |

PD+I flou dans les deux cas. Le tableau montre également que les termes de prédiction aggravent les résultats, et cela peut être expliqué par : l'erreur de modélisation qui est considérée comme des termes ajoutés aux fonctions coût, le comportement de l'imprévisibilité des systèmes chaotiques, ou la structure de la méthode de contrôle proposée ont besoin d'améliorations.

Bien que la structure de l'algorithme proposé est plus simple que celle d'autres méthodes de la littérature [124–128], le rôle de la prédiction est toujours sujette à caution.

3.4 Contrôle prédictif généralisé basé sur réseau neuronal

Comme mentionné dans la première section, la linéarisation du système contrôlé par la méthode GPC classique restreint sa capacité à faire face à une petite gamme de systèmes

non linéaires. Par conséquent, l'utilisation d'un réseau neuronal pour apprendre la dynamique du système peut améliorer la performance de la GPC pour couvrir une large zone de systèmes non linéaires. D'autre part, pour réduire la charge de calcul dans le processus d'optimisation ; certains outils avancés peuvent être impliqués tels que l'algorithme PSO.

3.4.1 Une introduction générale de la NNGPC

L'algorithme NNGPC se compose de trois blocs principaux [129–131], le modèle de référence qui donne la performance souhaitée pour le système, un algorithme de minimisation de la fonction coût (CFM) et un réseau de neurones qui décrit le dynamique du système contrôlé, comme représenté sur Fig. 3.22.

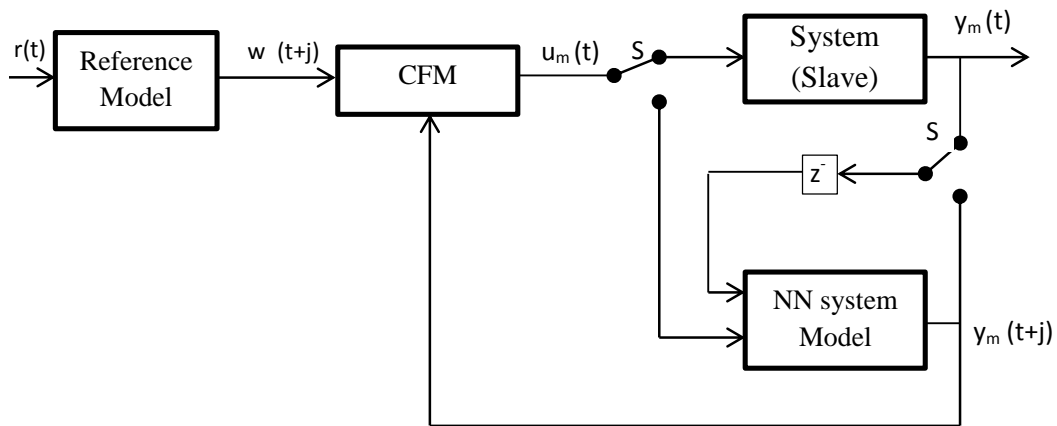


FIGURE 3.22 La structure générale de la NNGPC.

L'algorithme NNGPC commence par le signal d'entrée, $r(t)$, qui est présenté au modèle de référence. Ce modèle produit un signal de référence de suivi, $w(t+j)$, qui est utilisé comme une entrée du bloc CFM. L'algorithme de CFM produit une sortie qui est soit utilisée comme entrée au système ou au modèle du système. Le double interrupteur double throw pôle, S, est réglé lorsque le CFM a résolu pour la meilleure entrée, $u(t)$, qui permettra de minimiser la fonction de coût. Entre les échantillons, le commutateur est réglé sur le modèle du système où l'algorithme de CFM utilise ce modèle pour calculer la séquence de contrôle optimal. Une fois que la fonction de coût est minimisée, cette entrée est transmise à au système. Les principales étapes de NNGPC peuvent être résumées à ce qui suit :

1. utilisation du modèle de référence puis génération de la trajectoire de référence,
2. Débute par l'entrée passé de commande calculée, et de prédire la performance du système en utilisant le modèle,

3. Calculer la nouvelle séquence de commande qui minimise la fonction de coût,
4. Répétez l'étape 2 et 3 jusqu'à ce que la minimisation souhaitée obtenue,
5. Envoyer le premier élément de la séquence optimale au système,
6. Répétez les étapes ci-dessus pour chaque étape.

Un modèle NARX peut être utilisé pour décrire la dynamique des systèmes non-linéaires contrôlés ;

$$\begin{aligned} \hat{y}(k+1) = f(y(k), y(k-1), \dots, y(k-n_y), \dots, \\ u(k), u(k-1), \dots, u(k-n_u)), \end{aligned} \quad (3.105)$$

où n_y et n_u sont les retards sur leurs entrées, $u(k)$ et $y(k)$ représentent l'entrée et la sortie du système, respectivement, $f(\cdot)$ est une fonction non linéaire inconnue pour être estimée par un réseau de neurones artificiels. La formation initiale du modèle de réseau neuronal doit être effectuée hors ligne avant que le contrôle est testé. La configuration de bloc pour former un réseau neuronal pour modéliser les systèmes contrôlés est représenté sur la Fig. 3.23.

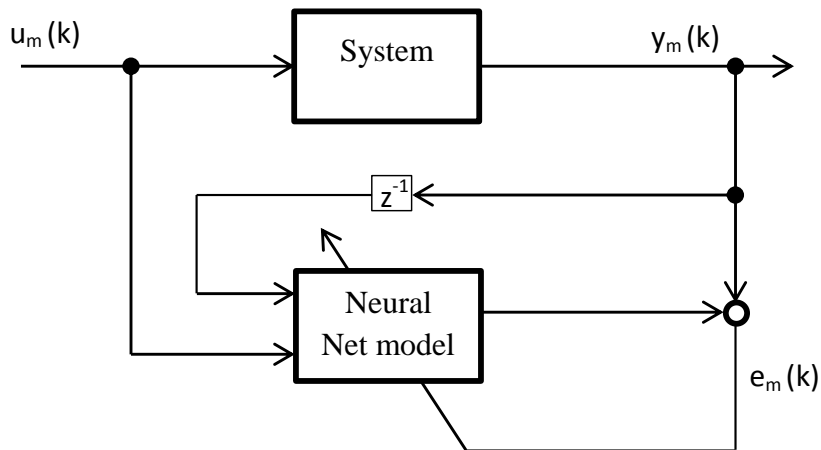


FIGURE 3.23 Schéma de l'apprentissage du réseau neuronal off-line.

Le réseau et le système de contrôle reçoivent la même entrée. Cependant, le réseau possède une entrée supplémentaire qui provient de la sortie du système. Cette entrée va aider le réseau à capturer la dynamique du système. Pendant le processus de formation, les poids sont ajustés de telle sorte qu'un ensemble d'entrées produit l'ensemble de sorties souhaitées. L'erreur entre la réponse du réseau et celle du système, est calculée. Cette erreur est utilisée pour mettre à jour les coefficients de pondération.

Fig. 3.24 donne la structure de prédiction, alors que Fig. 3.25 représente un réseau de neurones, multicouches avec une structure de retard. Les entrées du réseau de neurones sont

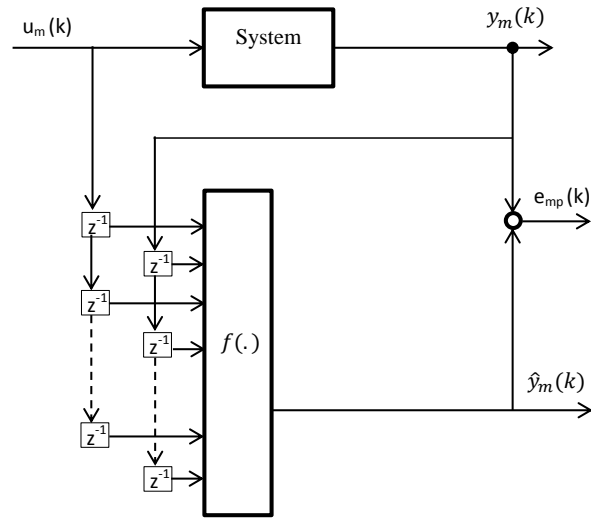


FIGURE 3.24 One-step prediction en mode série-parallel.

constitués des u et y , et leurs entrées de retard correspondant. Chaque entrée est associée à son noeud respectif. Le réseau a une couche cachée contenant de nombreux nœuds cachés qui utilisent la fonction tangente hyperbolique. Les noeuds de sortie utilisent une fonction linéaire de sortie. Ainsi, la sortie du réseau de neurones peut être réécrite sous la forme

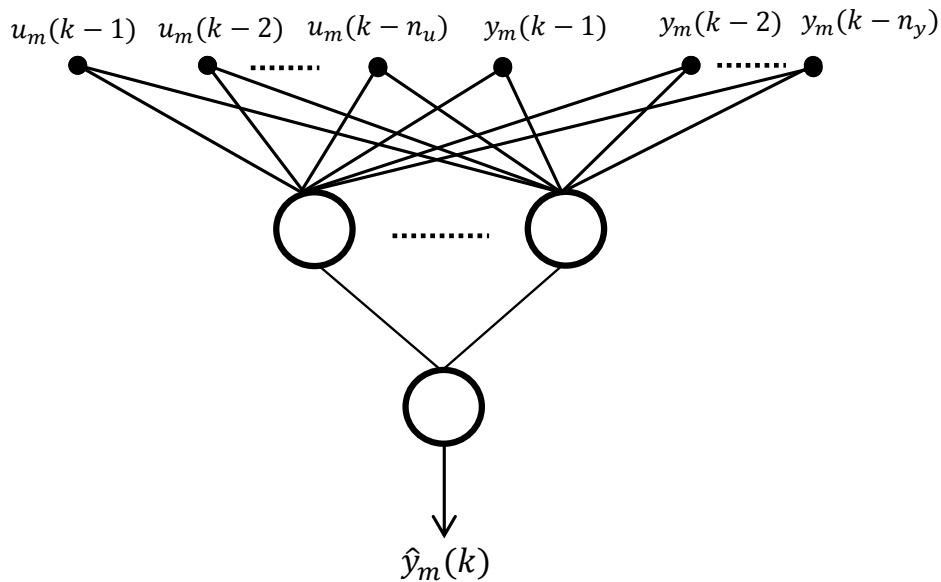


FIGURE 3.25 Réseau de neurones multicouche , avec une structure en temps tardé.

$$\hat{y}(k) = \sum_{j=1}^{hid} w_j \tanh(net_j(k)) + b, \tag{3.106}$$

et

$$net_j(k) = \sum_{i=1}^{n_u} w_{j,i}u(k-i) + \sum_{i=1}^{n_y} w_{j,n_u+i}y(k-i) + b_j, \quad (3.107)$$

où

$y(k)$ est la sortie du réseau de neurones, $f_j(\cdot)$ est la sortie de la fonction pour le j^{th} noeud de la couche cachée, $net_j(k)$ est le niveau d'activation de j^{th} noeud de sortie, hid est le nombre de noeuds cachés dans la couche cachée, w_j les poids reliant le j^{th} noeud caché au noeud de sortie, $w_{j,i}$ le poids connection entre le i^{th} noeud d'entrée au j^{th} noeud caché, b_j est le parti pris de la j^{th} noeud caché, b le biais sur le signe de tête de sortie.

Le réseau neuronal est utilisé comme un modèle pour prédire la dynamique du système à un certain temps futur $k+n$. Ceci peut être réalisé en déplaçant l'équation (3.106) et (3.107). Fig. 3.26 montre comment un prédicteur est calculé.

$$\hat{y}(k) = \sum_{j=1}^{hid} w_j \tanh(net_j(k+n)) + b, \quad (3.108)$$

et

$$\begin{aligned} net_j(k) = & \sum_{i=1}^{n_u} w_{j,i} \begin{cases} u(k+n-i), n - N_u \leq i \\ u(k+N_u), n - N_u \geq i \end{cases} \\ & + \sum_{i=1}^{\min(n,n_y)} w_{j,n_u+i} \hat{y}(k+n-i) \\ & + \sum_{i=1}^{n_y} w_{j,n_u+i} y(k+n-i) \\ & + b_j, \end{aligned} \quad (3.109)$$

3.4.2 Simulation numérique

Avant d'appliquer la NNGPC au système, le réseau neuronal est formé en utilisant MATLAB (Neural network system identification), qui repose sur des algorithmes de Levenberg-Marquardt et erreur quadratique moyenne. La fonction coût suivante est utilisée pour la synchronisation des systèmes chaotiques incertains :

$$J(k) = \sum_{j=1}^2 [y_m(k+j) - x_m(k+j)]^2 \quad (3.110)$$

L'algorithme général pour la synchronisation des systèmes chaotiques est le même que

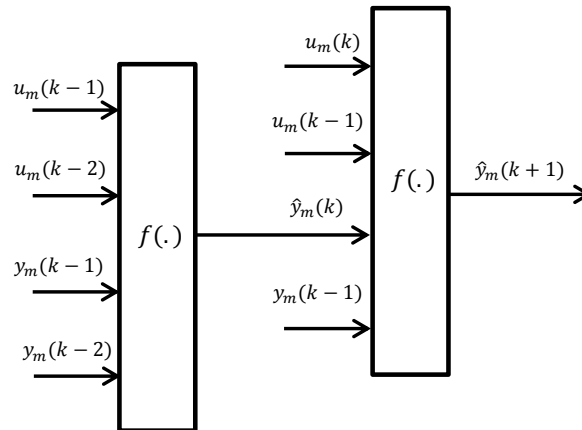


FIGURE 3.26 Two-step prediction

le NNGPC décrit ci-dessus. Toutefois, la référence $r(t)$ et $w(t+1)$ seront remplacés par $x_m(t)$ et $x_m(t+j)$, respectivement. Le modèle de référence est le même pour le modèle du système esclave, dans lequel les actions de commande sont supprimées. Dans le cas où le modèle de référence n'est pas exact, il peut être maintenue constante pendant la trajectoire future. Dans le cas d'un système MIMO, chaque canal est étudié séparément et a son propre modèle et le contrôleur (CFM). l'algorithme PSO est utilisé comme CFM pour trouver la séquence de contrôle optimale avec horizon de commande (N_u) est 2, la taille de l'essaim est de 30, le nombre d'itérations est égal à 40. L'objectif de cette étude est de montrer comment la NNGPC peut faire face à la synchronisation des systèmes chaotiques incertains. Trois systèmes sont utilisés pour vérifier les performance de la NNGPC.

Synchronisation des systèmes de Duffing

Dans cette section, une synchronisation des deux systèmes Duffing avec des incertitudes est simulée. Le système maître est décrit par l'équation suivante :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -0.4x_2 + 1.1x_1 - x_1^3 - 2.1\cos(1.8t). \end{cases} \quad (3.111)$$

Et le système esclave est décrit par l'équation suivante :

$$\begin{cases} \dot{y}_1 = y_2 \\ \dot{y}_2 = -0.1y_2 + 1.8y_1 - y_1^3 - 1.1\cos(0.4t) + u. \end{cases} \quad (3.112)$$

L'objectif du contrôle est de synchroniser le système esclave Duffing avec le système maître. L'action du contrôle u est ajouté au système esclave. Les conditions initiales des deux systèmes sont $x(0) = [2, -3]^T$ et $y(0) = [0, 1]^T$. Dans ce cas particulier, le système contrôlé étudié est un système SIMO de la forme suivante :

$$\begin{cases} \dot{y}_m = y_{m+1} \\ \dot{y}_n = g(x, t). \end{cases} \quad (3.113)$$

Par conséquent et afin de les synchroniser, la fonction coût doit être reformulée comme suit :

$$J_m(k) = \sum_{j=1}^2 [y_m(k+j) - x_m(k+j)]^2 + \dots + [y_n(k+j) - x_n(k+j)]^2 \quad (3.114)$$

Le réseau de neurones utilisé pour modéliser le système dispose de six entrées ($y_1(k-1), y_1(k-2), y_2(k-1), y_2(k-2), u(k-1), u(k-2)$) avec deux noeuds dans la couche cachée et deux sorties ($y_1(k), y_2(k)$). Le signal d'entrée u appliquée au système est une séquence finie de variation aléatoire uniformément répartis sur $[-4, 4]$. Les résultats de la synchronisation des deux systèmes et le signal de commande générés sont illustrés dans la Fig. 3.27.

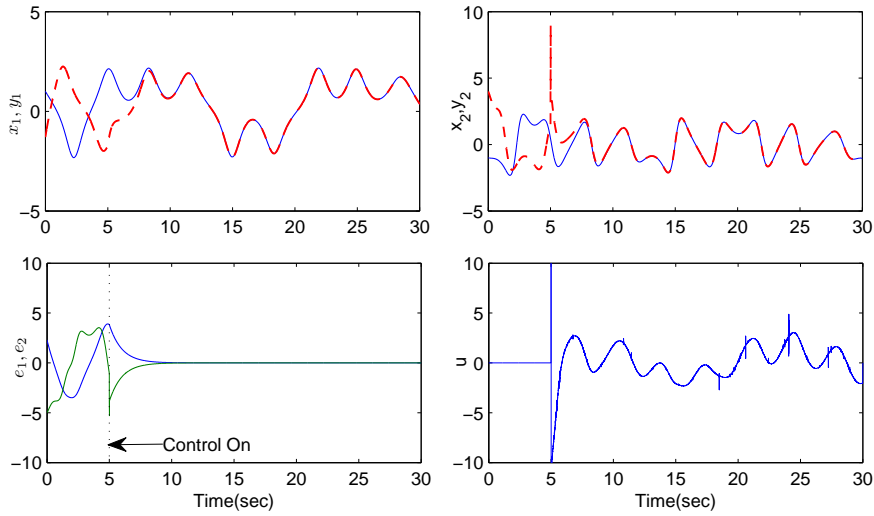


FIGURE 3.27 La synchronisation, les variations d'erreur, et l'action de commande pour les systèmes Duffing.

synchronzaiton des systèmes de Lorenz

Le maître est défini par :

$$\begin{cases} \dot{x}_1 = 10(x_2 - x_1) \\ \dot{x}_2 = (-x_1x_3 + 28x_1 - x_2), \\ \dot{x}_3 = x_1x_2 - 8/3x_3 \end{cases} \quad (3.115)$$

et l'esclave par :

$$\begin{cases} \dot{y}_1 = 11(y_2 - y_1) + u_1 \\ \dot{y}_2 = (-y_1y_3 + 25y_1 - y_2) + u_2, \\ \dot{y}_3 = y_1y_2 - (8/3 + 0.4)y_3 + u_3 \end{cases} \quad (3.116)$$

Pour chaque canal, un modèle spécifique est attribué. Un réseau neuronal structuré en 6-3-1 est utilisé. Les entrées du modèle sont $y_m(k-1), y_m(k-2), y_m(k-3), y_m(k-4), u_m(k-1), u_m(k-2)$, alors qu'il ne dispose que d'une sortie $\hat{y}(k)$. La fonction coût utilisée pour ce cas est la suivante :

$$J_m(k) = \sum_{j=1}^2 [y_m(k+j) - x_m(k+j)]^2 \quad (3.117)$$

Les conditions initiales des deux systèmes sont $x(0) = [-6, 0, 4]^T$ et $y(0) = [2, -1, 0]^T$.

Les résultats de la synchronisation des systèmes, le signal d'entrée de commande et les erreurs de synchronisation sont illustrés sur Fig. 3.28 et Fig. 3.29.

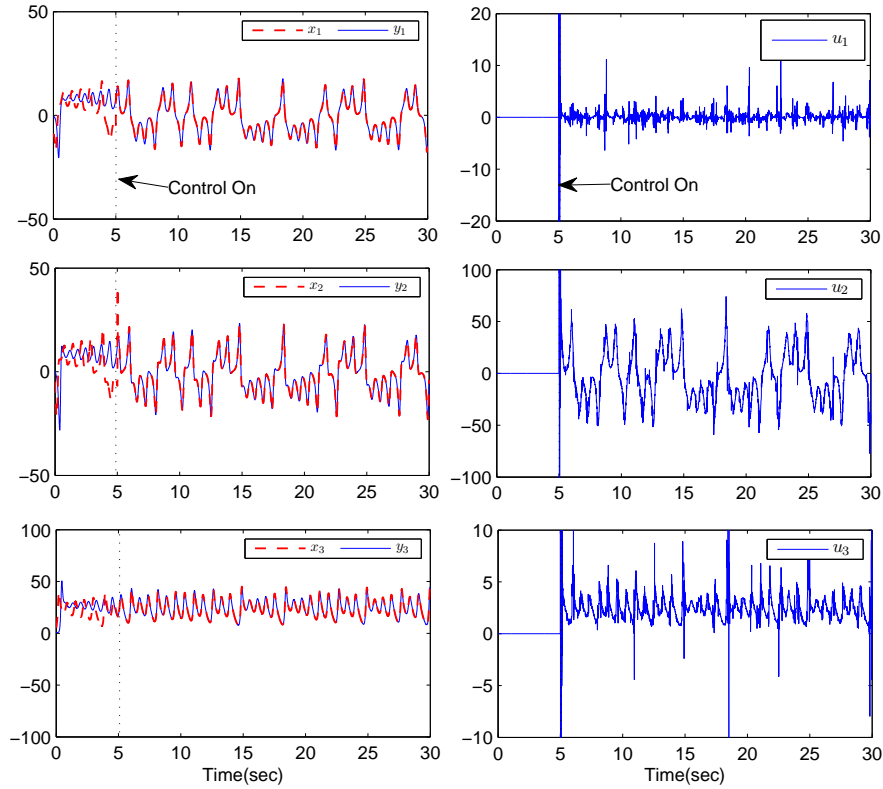


FIGURE 3.28 La synchronisation et l'action de commande pour les systèmes de Lorenz.

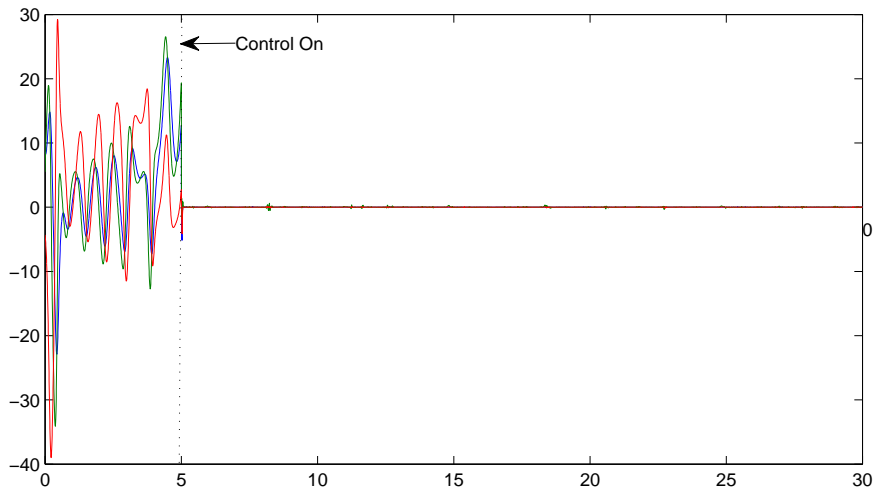


FIGURE 3.29 Dynamique des états d'erreur de synchronisation pour les systèmes de Lorenz.

synchronzaiton des systèmes hyperchaotic 4D Lu

Dans ce cas, la performance de l'algorithme proposé est vérifié pour la synchronisation des deux systèmes hyperchaotic incertains de Lu. Le maître est défini par :

$$\begin{cases} \dot{x}_1 = 17(x_2 - x_1) \\ \dot{x}_2 = 11x_2 - x_1x_3 + x_4, \\ \dot{x}_3 = x_1x_2 - 4x_3 \\ \dot{x}_4 = x_3 - 1.5x_4 \end{cases} \quad (3.118)$$

et l'esclave par :

$$\begin{cases} \dot{y}_1 = 14(y_2 - y_1) + u_1 \\ \dot{y}_2 = 9y_2 - y_1y_3 + y_4 + u_2, \\ \dot{y}_3 = y_1y_2 - 6y_3 + u_3 \\ \dot{y}_4 = y_3 - 0.9y_4 + u_4 \end{cases} \quad (3.119)$$

La structure du réseau de neurones utilisé est la même que celle utilisée pour le cas précédent. Les conditions initiales des deux systèmes sont $x(0) = [-1.3, 0.6, 0.8, -0.3]^T$ et $y(0) = [-9.9, -4.9, 5.1, 10.1]^T$. Les résultats sont illustrés sur Fig. 3.30 et Fig. 3.31.

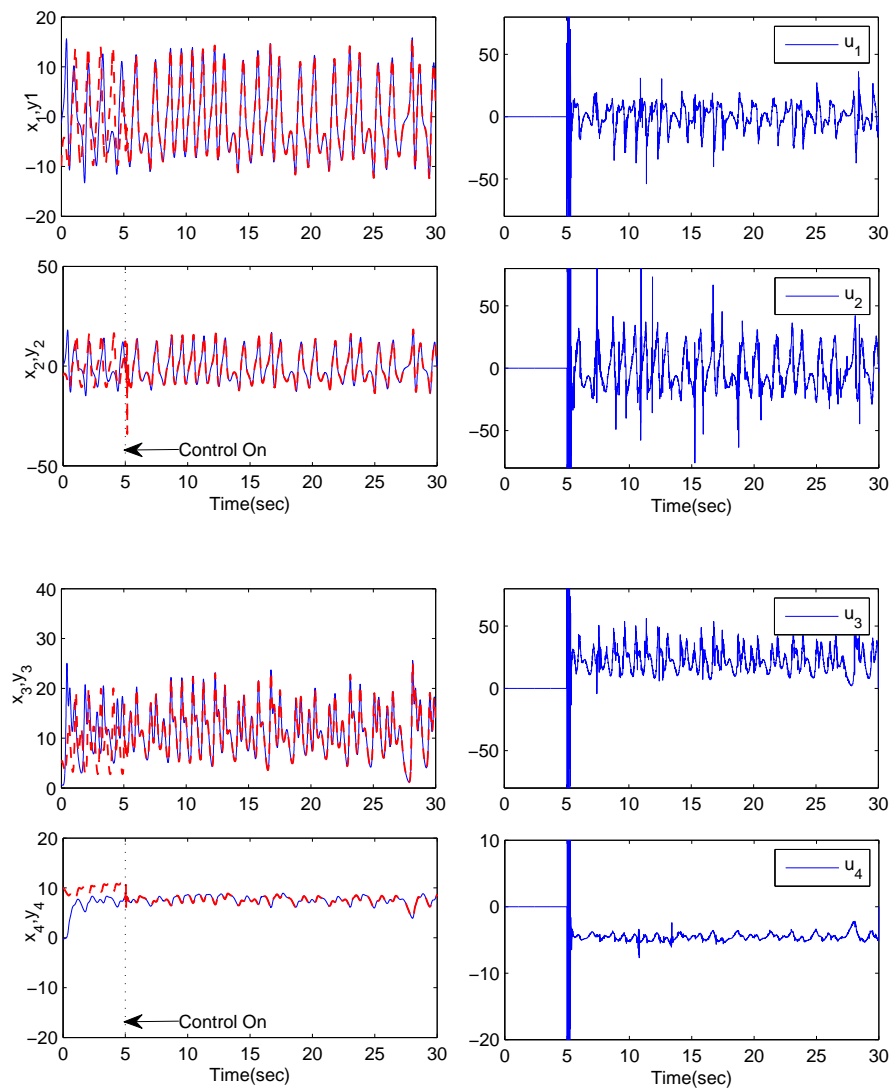


FIGURE 3.30 La synchronisation, et l'action de commande pour les systèmes Lu.

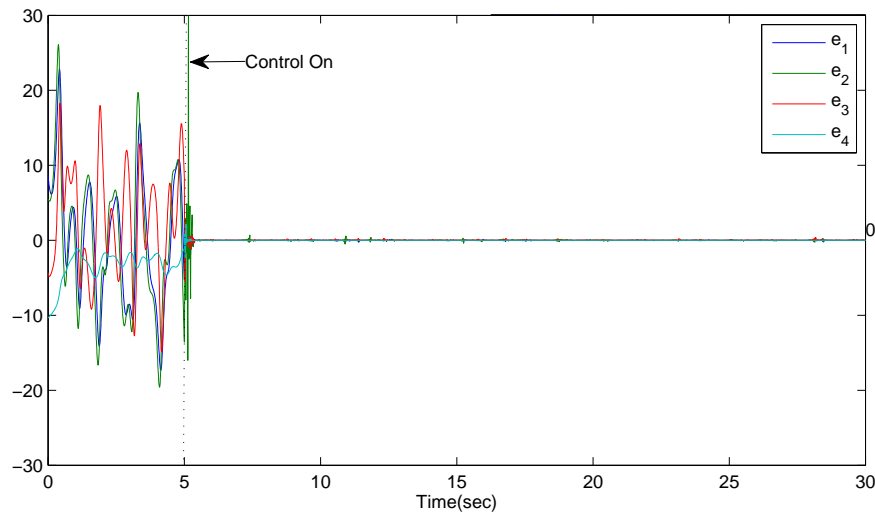


FIGURE 3.31 Dynamique des états d'erreur de synchronisation pour les systèmes Lu

3.5 Conclusion

Dans ce chapitre, nous avons étudié la synchronisation des systèmes chaotiques incertains en utilisant deux approches différentes de GPC. La première est basée sur les contrôleurs PID flous, et le second est basé sur un modèle de réseau neuronal. Dans le premier cas, le contrôleur n'a pas besoin d'un modèle précis pour le processus d'optimisation, seulement un modèle linéaire (ARX) est suffisant. Cependant, le rôle de la prédiction n'est pas clair et dépend du contrôleur utilisé dans l'optimisation de la fonction coût. Par ailleurs, il semble que le régulateur PID flou ait le seul élément qui assure la synchronisation dans laquelle le modèle est utilisé en tant que composant axillaire. En revanche, la seconde approche, qui a la même structure que la majorité des algorithmes de GPC, a besoin d'un modèle en tant que composant essentiel utilisé par le CFM entre les échantillons pour trouver la séquence de contrôle optimal. Un échec pour identifier le comportement du système contrôlé conduira automatiquement à la destruction de la stabilité. Pour parvenir à la synchronisation des systèmes chaotiques incertains, les deux approches sont appliquées avec succès avec une légère différence dans le rôle des termes de prédiction dans le cas des contrôleurs PID flous. En outre, le comportement chaotique des systèmes n'a pas un effet évident sur la performance des deux approches.

Chapitre 4

Synchronisation hybride des systèmes chaotiques incertains

4.1 Introduction

Parmi les nombreux types de synchronisation proposées, on peut citer la synchronisation complète (CS)[25], la anti-synchronisation (AS)[15] et la synchronisation projective(PS)[38]. La synchronisation complète se caractérise par des variables d'état d'égalité en constante évolution dans le temps alors que l'anti-synchronisation est un phénomène dans lequel les variables d'état des systèmes synchronisés avec des valeurs initiales différentes ont les mêmes valeurs absolues, mais de signes opposés.

Dans la synchronisation hybride des systèmes chaotiques[18], les états impairs des systèmes maître et esclave sont complètement synchronisés alors que les états pairs sont anti-synchronisés. Par exemple, si le système maître et le système esclave ont trois états chacun, le premier et le troisième état sont complètement synchronisés, tandis que le second état est anti-synchronisé. Par conséquent, dans la synchronisation hybride, la synchronisation complète (CS) et l'anti-synchronisation (AS) coexistent dans le système. Cette coexistence des deux types de synchronisation permet de nouvelles applications et l'amélioration de la sécurité dans certains systèmes de communication et de cryptage chaotique.

Comme mentionné dans le chapitre 2, les types de synchronisation, tels que la synchronisation de multi-commutation, peuvent être utilisés pour résoudre le problème de la dépendance linéaire. Dans ce chapitre, nous nous intéressons au problème de la condition LI dans la synchronisation hybride des systèmes chaotiques incertains. De plus, nous allons vérifier si ce type de synchronisation entraîne un problème dans le cas où la commande prédictive généralisée (GPC) basée sur le réseau neuronal est utilisée.

Définition du problème

Dans cette section, nous allons mettre en évidence le concept principal de la synchronisation hybride en donnant quelques exemples simples. Considérons les systèmes maître et esclave suivants :

$$\dot{x} = f(t, x) \quad (4.1)$$

$$\dot{y} = f(t, y) + U(t, x, y), \quad (4.2)$$

où $x(t), y(t) \in \mathfrak{R}^n$ sont les vecteurs d'état de dimension n , des systèmes maître et esclave, respectivement ; $f, g : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$ sont les vecteurs non linéaires des fonctions continues ; $U(t, x, y)$ est le vecteur de commande .

Le vecteur d'erreur est défini comme suit :

$$e(t) = y(t) - Mx(t), \quad (4.3)$$

où M est une matrice diagonale constante, i.e., $M = \text{diag}(m_1, m_2, \dots, m_n)$ ($m_i \neq 0, i = 1, 2, \dots, n$).

Le système maître et le système esclave sont dits prêts pour réaliser la synchronisation hybride, s'il existe un contrôleur $U(t, x, y)$ tel que l'équation :

$$\lim_{t \rightarrow \infty} \|e(t)\| = \lim_{t \rightarrow \infty} \|y(t, y) - Mx(t)\| = 0$$

est satisfaite pour des conditions initiales arbitraires $x(0)$ et $y(0)$.

Pour une synchronisation hybride(1.-1.1), c'est-à-dire une synchronisation où les états pairs seraient anti-synchronisés et les états impairs synchronisés, les facteurs des éléments impairs de la matrice M $m_{i\text{odd}} = 1$, alors que les facteurs pairs $m_{i\text{even}} = -1$.

Pour illustrer ce principe on considère les deux exemples suivants.

Exemple 1 (Synchronisation hybride de deux systèmes de Rikitake identiques) *Le système de Rikitake maître est décrit par les équations suivantes :*

$$\begin{cases} \dot{x}_1 &= -bx_1 + x_2x_3, \\ \dot{x}_2 &= -bx_2 + (x_3 - a)x_1, \\ \dot{x}_3 &= 1 - x_1x_2, \end{cases} \quad (4.4)$$

et le système esclave par :

$$\begin{cases} \dot{y}_1 = -by_1 + y_2y_3 + u_1(t), \\ \dot{y}_2 = -by_2 + (y_3 - a)y_1 + u_2(t), \\ \dot{y}_3 = 1 - y_1y_2 + u_3(t), \end{cases} \quad (4.5)$$

où $u(t) = [u_1(t), u_2(t), u_3(t)]^T$ représente le contrôleur à concevoir pour que les états impairs des deux systèmes se synchronisent et les états pairs s'anti-synchronisent. Les états des erreurs dans ce cas sont définis comme suit :

$$\begin{cases} e_1 = y_1 - x_1, \\ e_2 = y_2 + x_2, \\ e_3 = y_3 - x_3. \end{cases} \quad (4.6)$$

C'est-à-dire que le premier et le troisième état des deux systèmes doivent se synchroniser alors que le deuxième état des deux systèmes doivent être anti-synchronisés.

Le système dynamique de l'erreur est obtenu en soustrayant le système (4.4) du système (4.5), tel que

$$\begin{cases} \dot{e}_1 = -be_1 + y_2y_3 - x_2x_3 + u_1(t), \\ \dot{e}_2 = -be_2 + (y_3 - a)y_1 + (x_3 - a)x_1 + u_2(t), \\ \dot{e}_3 = -y_1y_2 + x_1x_2 + u_3(t). \end{cases} \quad (4.7)$$

En choisissant les fonctions de commande :

$$\begin{cases} u_1(t) = -y_2y_3 + x_2x_3 + V_1(t) \\ u_2(t) = -(y_3 - a)y_1 - (x_3 - a)x_1 + V_2(t) \\ u_3(t) = y_1y_2 - x_1x_2 + V_3(t), \end{cases} \quad (4.8)$$

on obtient :

$$\begin{cases} \dot{e}_1 = -be_1 + V_1(t), \\ \dot{e}_2 = -be_2 + V_2(t), \\ \dot{e}_3 = V_3(t). \end{cases} \quad (4.9)$$

où $V_1(t), V_2(t), V_3(t)$ sont les entrées des commande linéaires. L'objectif est de trouver la commande de rétroaction appropriée, qui rende le système (4.9) définie négative de telle sorte que l'erreur converge vers zéro quand $t \rightarrow \infty$. Ce qui implique que les systèmes (4.4)

et (4.5) sont globalement synchronisés. Nous choisissons

$$\begin{bmatrix} V_1(t) \\ V_2(t) \\ V_3(t) \end{bmatrix} = P \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \end{bmatrix},$$

où P est 3×3 matrice constante. Afin de rendre le système (4.9) asymptotiquement stable, les choix appropriés des éléments de la matrice P doit se faire de telle manière à ce que le système de rétroaction doit avoir toutes ses valeurs propres avec une partie réelle négative.

On Considère le choix suivant pour la matrice P :

$$P = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Le système d'erreur devient

$$\begin{cases} \dot{e}_1 = -be_1, \\ \dot{e}_2 = -be_2, \\ \dot{e}_3 = -e_3. \end{cases} \quad (4.10)$$

Les trois valeurs propres $\lambda_i (i = 1, 2, 3)$ du système sont : $-b, -b, -1$. L'erreur e_1, e_2, e_3 va donc converger vers zéro quand $t \rightarrow \infty$. Par conséquent, le système (4.10) est stable, ce qui assure bien la synchronisation des systèmes (4.4) et (4.5).

Pour la simulation numérique de cet exemple, nous utilisons la méthode d'Euler avec un pas $T=0,001$. Les paramètres du système Rikitake sont choisis comme suit $a = 5, b = 2$ ce qui assure pour le système Rikitake un comportement chaotique.

Les valeurs initiales du système maître et du système esclave respectivement, sont pris comme : $x(0) = [-15, -17, 15]^T$ and $y(0) = [5, 0.5, 1.5]^T$, respectivement.

Les résultats de la synchronisation hybride des deux systèmes sont représentés sur Fig. 4.1, alors que Fig. 4.2 représente les variations d'erreur entre les états des deux systèmes.

Exemple 2 (Synchronisation hybride de deux systèmes de Lorenz identiques) La méthode de contrôle utilisée dans l'exemple précédent, est également mise en œuvre pour la synchronisation de deux systèmes de Lorenz identiques. Le système maître est décrit par les

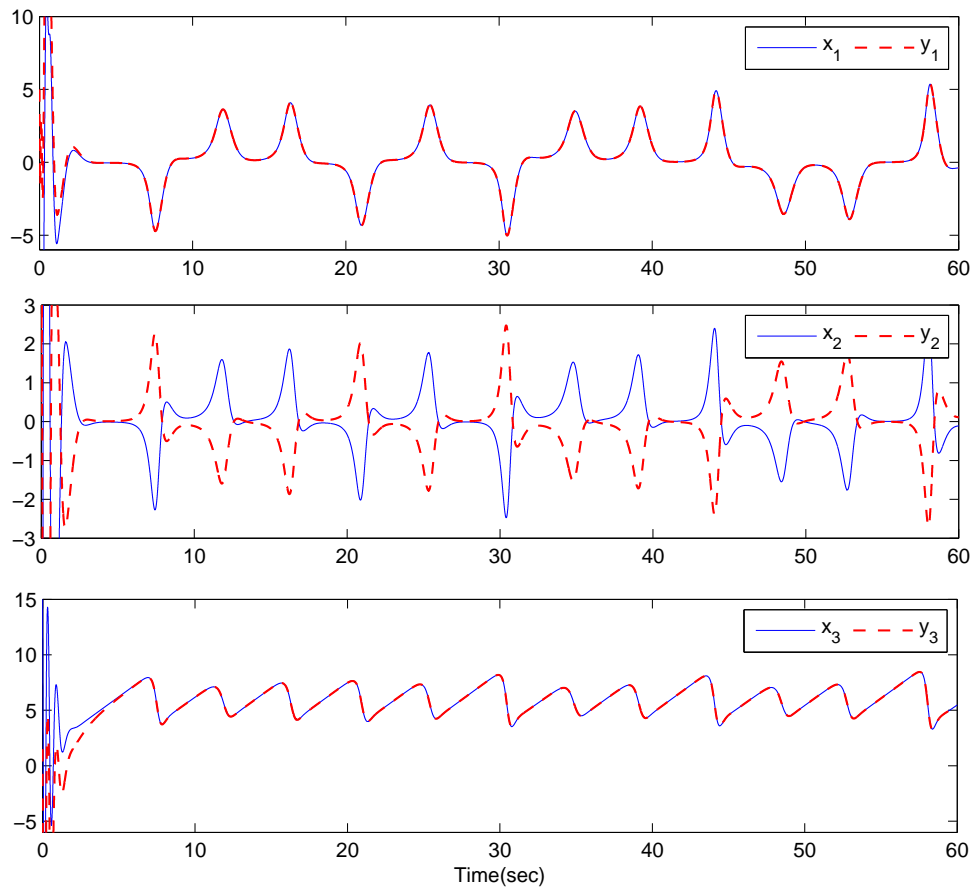


FIGURE 4.1 Synchronisation hybride de deux systèmes de Rikitake identiques.

équations suivantes :

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1), \\ \dot{x}_2 = cx_1 - x_1x_3 - x_2, \\ \dot{x}_3 = x_1x_2 - bx_3, \end{cases} \quad (4.11)$$

où x_1, x_2, x_3 représentent les variables d'état et a, b, c, d les paramètres constants, positifs du système.

Le système esclave contrôlé est défini comme suit :

$$\begin{cases} \dot{y}_1 = a(y_2 - y_1) + u_1(t), \\ \dot{y}_2 = cy_1 - y_1y_3 - y_2 + u_2(t), \\ \dot{y}_3 = y_1y_2 - by_3 + u_3(t), \end{cases} \quad (4.12)$$

avec y_1, y_2, y_3 les variables d'état et $u_1(t), u_2(t), u_3(t)$ les éléments du contrôleur à conce-

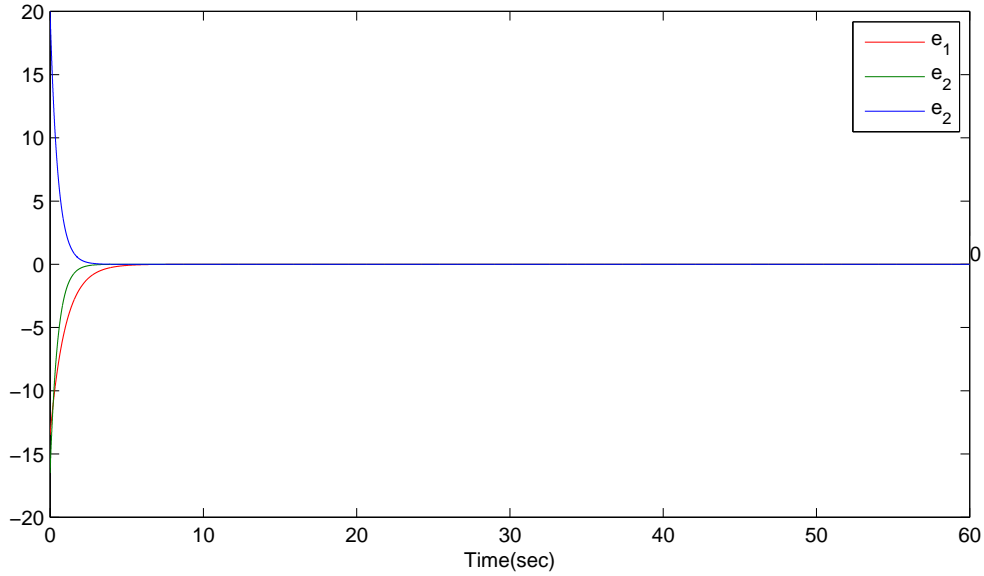


FIGURE 4.2 L'évolution des fonctions d'erreur e_1, e_2, e_3 durant la synchronisation hybride des deux systèmes de Rikitake identiques.

voir:

Comme pour l'exemple précédent, on considère une synchronisation hybride où les états impairs des deux systèmes se synchronisent et les états pairs s'anti-synchronisent. Les erreurs de synchronisation sont donc définies comme suit :

$$\begin{cases} e_1 = y_1 - x_1, \\ e_2 = y_2 + x_2, \\ e_3 = y_3 - x_3. \end{cases} \quad (4.13)$$

Et leur dynamique est :

$$\begin{cases} \dot{e}_1 = a(y_2 - y_1) - a(x_2 - x_1) + u_1(t), \\ \dot{e}_2 = -y_1 y_3 + c y_1 - y_2 - x_1 x_3 + c x_1 - x_2 + u_2(t), \\ \dot{e}_3 = y_1 y_2 - b y_3 - x_1 x_2 + b x_3 + u_3(t). \end{cases} \quad (4.14)$$

Nous considérons le contrôleur non linéaire actif défini par :

$$\begin{cases} u_1(t) = -e_1 - a(y_2 - y_1) + a(x_2 - x_1) \\ u_2(t) = -e_2 + y_1 y_3 - c y_1 + y_2 + x_1 x_3 - c x_1 + x_2 \\ u_3(t) = -e_3 - y_1 y_2 + b y_3 + x_1 x_2 - b x_3, \end{cases} \quad (4.15)$$

Et en substituant (4.15) dans (4.14), on obtient le système linéaire suivant :

$$\begin{cases} \dot{e}_1 = -e_1, \\ \dot{e}_2 = -e_2, \\ \dot{e}_3 = -e_3. \end{cases} \quad (4.16)$$

En considérant la fonction de Lyapunov suivante

$$V(e) = \frac{1}{2} e^T e = \frac{1}{2} (e_1^2 + e_2^2 + e_3^2), \quad (4.17)$$

qui est une fonction définie positive sur \mathfrak{R}^3 et en différenciant (4.17) le long des trajectoires de (4.16), nous obtenons

$$\dot{V}(e) = -ae_1^2 - e_2^2 - de_3^2, \quad (4.18)$$

une fonction définie négative sur \mathfrak{R}^3 , ce qui vérifie la théorie de la stabilité de Lyapunov et assure que la dynamique de l'erreur est globalement stable. Pour la simulation numérique, les paramètres choisis pour les systèmes Lorenz sont :

$$a = 10, b = 8/3, c = 28$$

les conditions initiales du système maître sont :

$$x_1(0) = 21, x_2(0) = 18, x_3(0) = 20$$

Les conditions initiales du système esclave (4.12) et ceux de l'esclave sont :

$$y_1(0) = 12, y_2(0) = 34, y_3(0) = 15$$

Les résultats de la synchronisation hybride des deux systèmes sont représentés par Fig. 4.3, tandis que Fig. 4.4 représente les variations de la fonctions d'erreur.

4.2 La synchronisation hybride et la condition LI

La condition LI aborde le problème de l'estimation des paramètres inconnus. Dans le chapitre 2, nous avons introduit comment la dépendance linéaire des termes de synchronisation de deux systèmes chaotiques conduit à l'échec de l'estimation des paramètres inconnus dans le cas de d'synchronisation CS et de l'utilisation d'un contrôle non linéaire actif. Nous

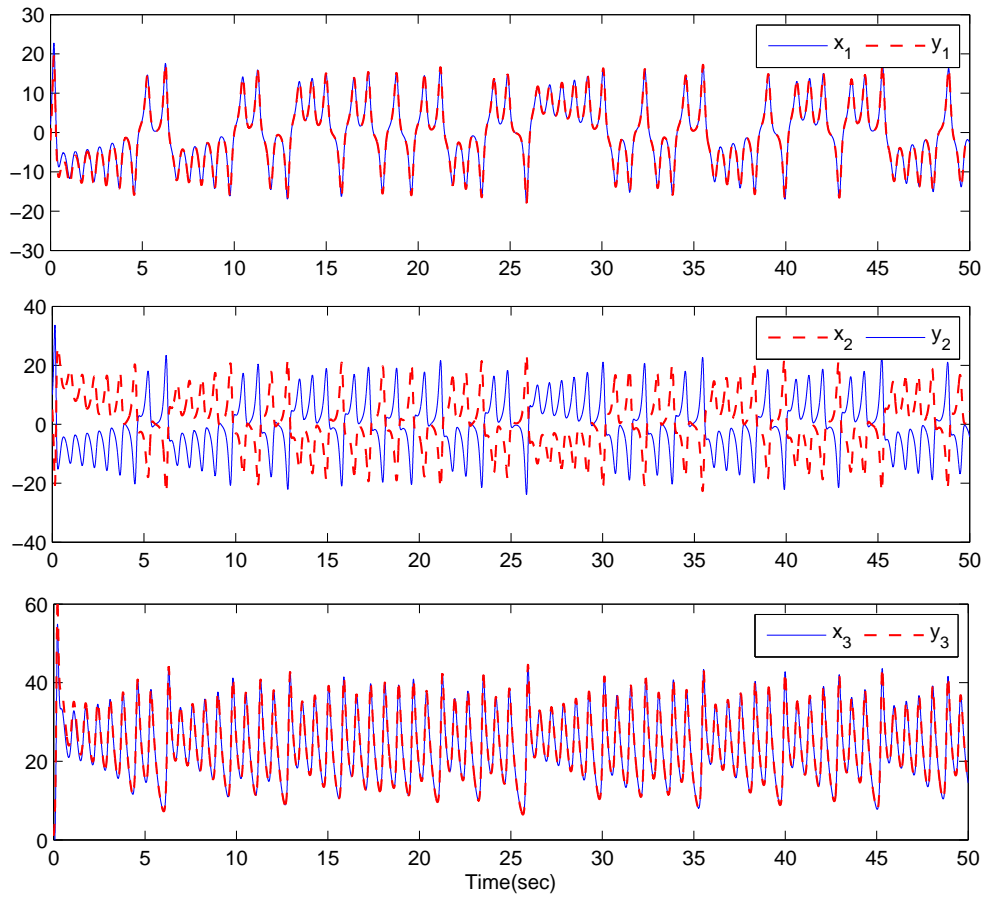


FIGURE 4.3 Synchronisation hybride des deux systèmes Lorenz.

avons également introduit la façon d'éviter le problème de la synchronisation externe par commutation des orbites de synchronisation ou en éliminant les termes de perturbation qui permettent d'obtenir l'estimation des paramètres inconnus.

Dans cette section, nous allons poursuivre l'étude de ce problème avec la synchronisation hybride des systèmes chaotiques incertains en l'illustrant à travers l'exemple suivant.

On considère le système chaotique maître, défini comme suit :

$$\begin{cases} \dot{x}_1 = a_1(x_2 - x_1), \\ \dot{x}_2 = c_1x_1 - x_1x_3 - x_2, \\ \dot{x}_3 = x_1x_2 - b_1x_3, \end{cases} \quad (4.19)$$

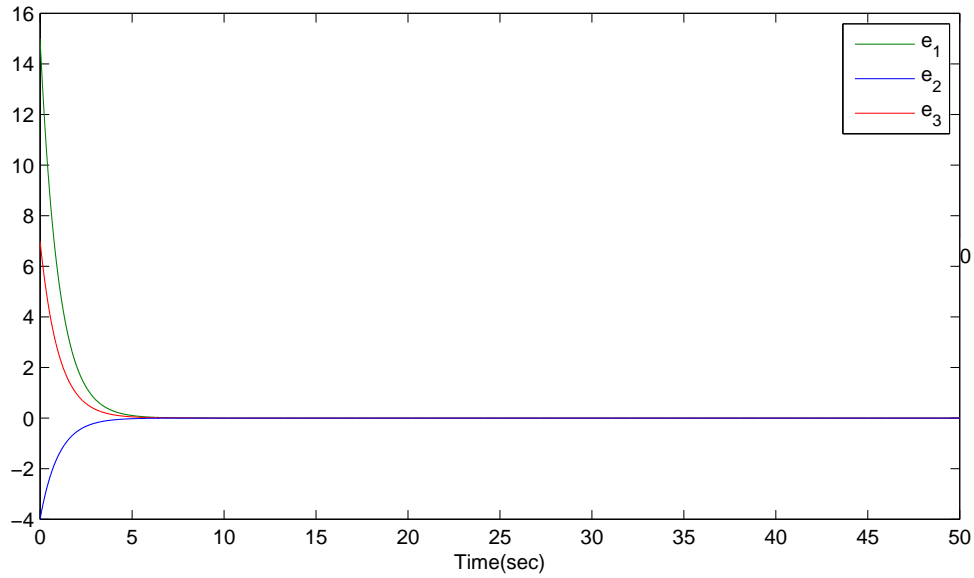


FIGURE 4.4 L'évolution des fonctions d'erreur e_1, e_2, e_3 de deux systèmes de Lorenz.

et le système esclave avec lequel, on veut le synchroniser :

$$\begin{cases} \dot{y}_1 = a_2(y_2 - y_1) + u_1, \\ \dot{y}_2 = c_2y_1 - y_1y_3 - y_2 + u_2, \\ \dot{y}_3 = y_1y_2 - b_2y_3 + u_3, \end{cases} \quad (4.20)$$

où x_1, x_2, x_3 et y_1, y_2, y_3 sont les variables d'état du maître et de l'esclave respectivement, et $u_1(t), u_2(t), u_3(t)$ sont les contrôleurs actifs à concevoir.

Pour la synchronisation hybride(1.-1.1) des ces systèmes, les erreurs de synchronisation sont définies comme suit :

$$\begin{cases} e_1 = y_1 - x_1, \\ e_2 = y_2 + x_2, \\ e_3 = y_3 - x_3. \end{cases} \quad (4.21)$$

Et

$$\begin{cases} \dot{e}_1 = a_2(y_2 - y_1) - a_1(x_2 - x_1) + u_1, \\ \dot{e}_2 = -y_1y_3 + c_2y_1 - y_2 - x_1x_3 + c_1x_1 - x_2 + u_2, \\ \dot{e}_3 = y_1y_2 - b_2y_3 - x_1x_2 + b_1x_3 + u_3, \end{cases} \quad (4.22)$$

Les fonctions de contrôle sont choisies comme suit :

$$\begin{cases} u_1 = -e_1 - \hat{a}_2(y_2 - y_1) + \hat{a}_1(x_2 - x_1), \\ u_2 = -e_2 + y_1 y_3 - \hat{c}_2 y_1 + y_2 + x_1 x_3 - \hat{c}_1 x_1 + x_2, \\ u_3 = -e_3 - y_1 y_2 + \hat{b}_2 y_3 + x_1 x_2 - \hat{b}_1 x_3, \end{cases} \quad (4.23)$$

où $\hat{a}_1, \hat{a}_2, \hat{b}_1, \hat{b}_2, \hat{c}_1, \hat{c}_2$ sont les valeurs estimées des paramètres $a_1, a_2, b_1, b_2, c_1, c_2$.

Si on pose $\tilde{a}_1 = a_1 - \hat{a}_1, \tilde{a}_2 = a_2 - \hat{a}_2, \tilde{b}_1 = b_1 - \hat{b}_1, \tilde{b}_2 = b_2 - \hat{b}_2, \tilde{c}_1 = c_1 - \hat{c}_1, \tilde{c}_2 = c_2 - \hat{c}_2$, alors la dynamique du système d'erreur est décrite comme :

$$\begin{cases} \dot{e}_1 = -e_{11} + \tilde{a}_2(y_2 - y_1) - \tilde{a}_1(x_2 - x_1), \\ \dot{e}_2 = -e_{12} + \tilde{c}_2 y_1 - \tilde{c}_1 x_1, \\ \dot{e}_3 = -e_{13} - \tilde{b}_2 y_3 + \tilde{b}_1 x_3, \end{cases} \quad (4.24)$$

On définit les lois d'adaptation des paramètres :

$$\begin{cases} \dot{\hat{a}}_1 = -e_1(x_2 - x_1), \\ \dot{\hat{b}}_1 = e_3 x_3, \\ \dot{\hat{c}}_1 = e_2 x_1, \\ \dot{\hat{a}}_2 = e_1(y_2 - y_1), \\ \dot{\hat{b}}_2 = -e_3 y_3, \\ \dot{\hat{c}}_2 = e_2 y_1. \end{cases} \quad (4.25)$$

Et on choisit la fonction Lyapunov suivante :

$$V = \frac{1}{2}(e^T e + \tilde{a}_1^2 + \tilde{a}_2^2 + \tilde{b}_1^2 + \tilde{b}_2^2 + \tilde{c}_1^2 + \tilde{c}_2^2). \quad (4.26)$$

La dérivation en temps de V le long des trajectoires de la dynamique d'erreur conduit à :

$$\begin{aligned} \dot{V} &= e_1 \dot{e}_1 + e_2 \dot{e}_2 + e_3 \dot{e}_3 - \tilde{a}_1 \dot{\hat{a}}_1 - \tilde{b}_1 \dot{\hat{b}}_1 - \tilde{c}_1 \dot{\hat{c}}_1 - \tilde{a}_2 \dot{\hat{a}}_2 - \tilde{b}_2 \dot{\hat{b}}_2 - \tilde{c}_2 \dot{\hat{c}}_2 \\ &= -e_1^2 - e_2^2 - e_3^2 < 0. \end{aligned} \quad (4.27)$$

V est une fonction définie positive et \dot{V} est une fonction définie négative, i.e. les états du système esclave contrôlé va suivre les états du système maître asymptotiquement.

Pour la simulation, on choisit pour le maître : $a_1 = 10, b_1 = 8/3 + 0.4, c_1 = 28$ et pour l'esclave : $a_2 = 9, b_2 = 8/3 + 0.2, c_2 = 27$. Les valeurs initiales des deux systèmes sont : $x_1(0) = 2, x_2(0) = 10, x_3(0) = -6$ and $y_1(0) = -2, y_2(0) = 5, y_3(0) = 1$.

De l'étude faite dans le chapitre 2, nous pouvons prédire quels paramètres vont converger

vers les vraies valeurs. les conditions suivantes se produisent lorsque $t \rightarrow \infty$:

$$\begin{aligned} e_1 &\rightarrow 0, e_2 \rightarrow 0, e_3 \rightarrow 0 \\ x_1 &= y_1|_{t \rightarrow \infty}, -x_2 = y_2|_{t \rightarrow \infty}, x_3 = y_3|_{t \rightarrow \infty} \end{aligned}$$

En insérant ces conditions dans les équations(4.24), nous obtenons

$$\begin{cases} (a_2 - \hat{a}_2)(y_2 - y_1) - (a_1 - \hat{a}_1)(x_2 - x_1) = 0, \\ (c_2 - \hat{c}_2)y_1 + (c_1 - \hat{c}_1)x_1 = 0, \\ -(b_2 - \hat{b}_2)y_3 + (b_1 - \hat{b}_1)x_3 = 0, \end{cases} \quad (4.28)$$

$$\begin{cases} (a_2 - \hat{a}_2)(y_2 - y_1) - (a_1 - \hat{a}_1)(x_2 - x_1) = (a_2 - \hat{a}_2)(-x_2 - x_1) - (a_1 - \hat{a}_1)(x_2 - x_1)|_{t \rightarrow \infty}, \\ (c_2 - \hat{c}_2)y_1 + (c_1 - \hat{c}_1)x_1 = x_1((c_2 - \hat{c}_2) + (c_1 - \hat{c}_1))|_{t \rightarrow \infty}, \\ -(b_2 - \hat{b}_2)y_3 + (b_1 - \hat{b}_1)x_3 = x_3((b_1 - \hat{b}_1) - (b_2 - \hat{b}_2))|_{t \rightarrow \infty}, \end{cases} \quad (4.29)$$

De l'équation (4.29), nous pouvons remarquer que seuls les termes de fonction a_1 et a_2 seront couronnées de succès.

Fig. 4.5 montre la synchronisation hybride entre les deux systèmes alors que Fig. 4.6 montre la variation du système d'erreur. Les valeurs initiales des paramètres estimés sont $\hat{a}_1(0) = 2, \hat{b}_1(0) = 10, \hat{c}_1(0) = -6, \hat{a}_2(0) = -2, \hat{b}_2(0) = 5, \hat{c}_2(0) = 1$. La variation des paramètres estimés sont présentés dans figure Fig. 4.7.

D'après les résultats de la simulation, nous remarquons que seule l'estimation des paramètres a_1 et a_2 est correct.

Pour y remédier, nous utilisons dans une deuxième étape la deuxième méthode proposée dans le chapitre précédent et qui repose sur l'élimination des termes de perturbation causée par les incertitudes des paramètres vers la synchronisation hybride des systèmes chaotiques incertains. Les erreurs des paramètres des systèmes maître et esclave sont données par

$$\begin{cases} e_a &= a_2 - a_1 \\ e_c &= c_2 - c_1. \\ e_b &= b_2 - b_1 \end{cases} \quad (4.30)$$

En utilisant l'équation (4.21) et après un calcul simple, la dynamique des états d'erreur est

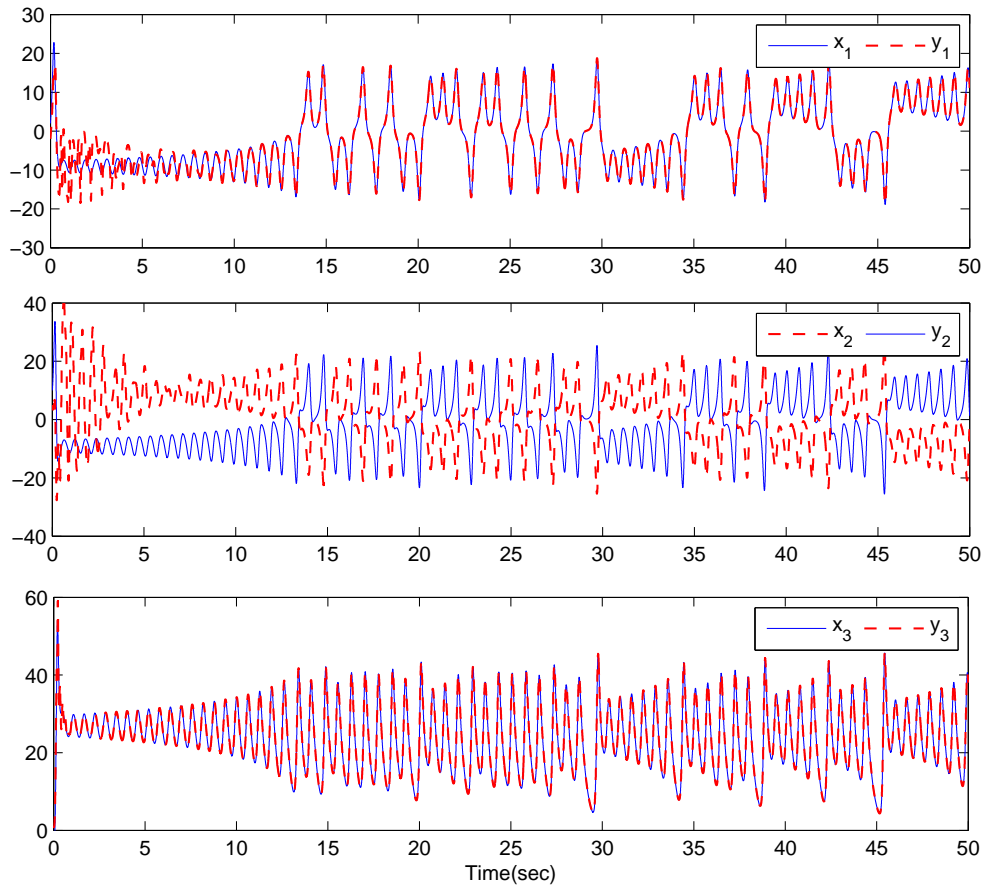


FIGURE 4.5 Synchronisation hybride de deux systèmes de Lorenz incertains.

donnée par :

$$\begin{cases} \dot{e}_1 = g_1(y) + a_1(e_2 - e_1) + u_1 \\ \dot{e}_2 = g_2(y) - y_2 - y_1 y_3 + c_1(x_1 + y_1) - x_2 - x_1 x_3 + u_2, \\ \dot{e}_3 = g_3(y) - b_1 e_3 + y_1 y_2 - x_1 x_2 + u_3 \end{cases} \quad (4.31)$$

où

$$\begin{cases} g_1(y) = e_a(y_2 - y_1) \\ g_2(y) = e_c y_1, \\ g_3(y) = -e_b y_3 \end{cases} \quad (4.32)$$

représente les termes de perturbation.

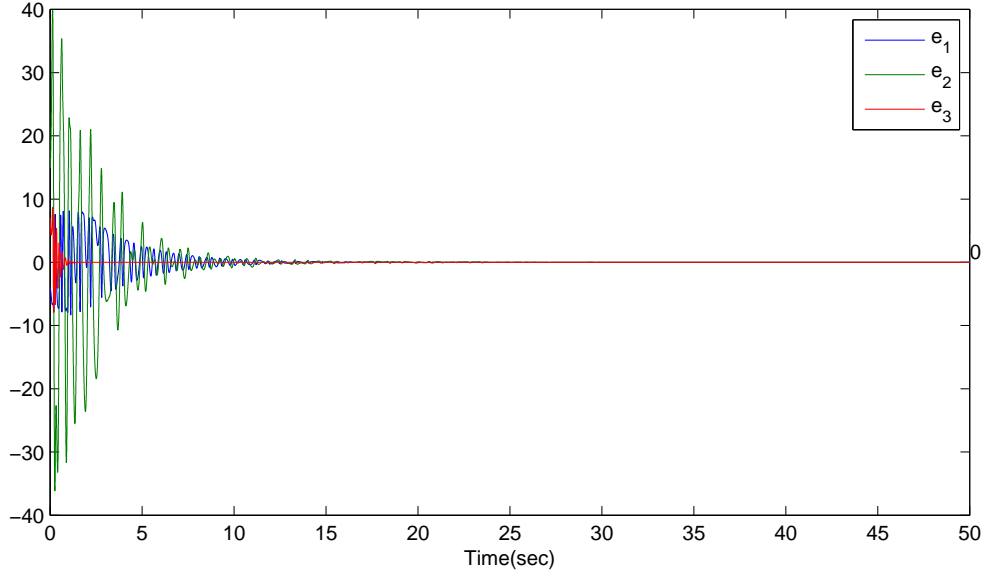


FIGURE 4.6 L'évolution des fonctions d'erreur e_1, e_2, e_3 de deux systèmes de Lorenz incertains

Ces termes seront éliminés par l'action de l'EOP et (4.31) devient

$$\begin{cases} \dot{e}_1 = a_1(e_2 - e_1) + u_1 \\ \dot{e}_2 = -y_2 - y_1 y_3 + c_1(x_1 + y_1) - x_2 - x_1 x_3 + u_2 \\ \dot{e}_3 = -b_1 e_3 + y_1 y_2 - x_1 x_2 + u_3 \end{cases} \quad (4.33)$$

Par conséquent, les contrôleurs actifs sont définis comme suit :

$$\begin{cases} u_1 = -e_1 - a_1(e_2 - e_1) \\ u_2 = -e_2 + y_2 + y_1 y_3 - c_1(x_1 + y_1) + x_2 + x_1 x_3 \\ u_3 = -e_3 - y_1 y_2 + x_1 x_2 + b_1 e_3 \end{cases} \quad (4.34)$$

Les paramètres utilisés dans la conception des contrôleurs actifs sont fixés à $a_1 = a_2 = 10, b_1 = b_2 = 8/3$ et $c_1 = c_2 = 28$, ce qui limite le contrôleur pour atteindre la synchronisation lorsque les paramètres des deux systèmes sont les mêmes que ceux des paramètres fixes des contrôleurs actifs.

La substitution de(4.34) dans (4.33) donne un système linéaire de la forme

$$\begin{cases} \dot{e}_1 = -e_1 \\ \dot{e}_2 = -e_2 \\ \dot{e}_3 = -e_3 \end{cases} \quad (4.35)$$

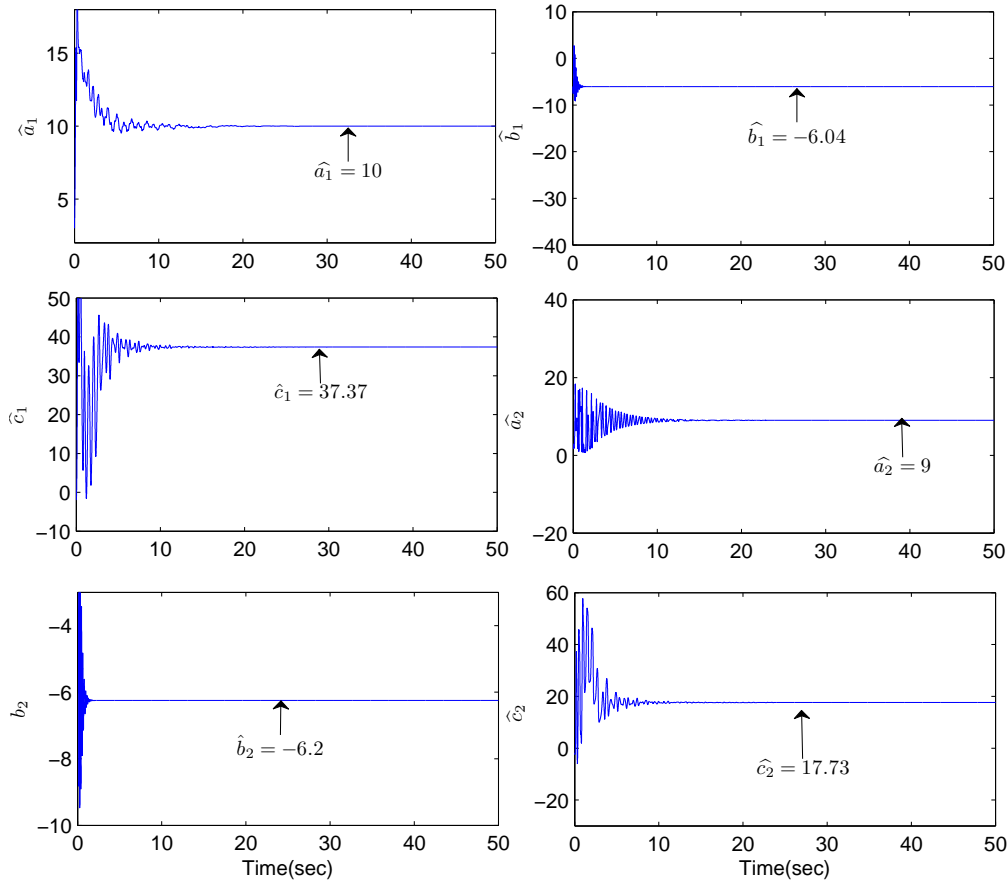


FIGURE 4.7 L'identification des paramètres en utilisant le contrôle active.

En choisissant une fonction de Lyapunov de la forme :

$$V(e) = \frac{1}{2}(e_1^2 + e_2^2 + e_3^2). \quad (4.36)$$

Dont la dérivée est :

$$\dot{V}(e) = -e_1^2 - e_2^2 - e_3^2, \quad (4.37)$$

ceci implique que le système contrôlé est globalement asymptotiquement stable. La même EOP (PI flou) est utilisée pour la synchronisation hybride. Les paramètres sont estimés de la même manière que celle décrite dans le chapitre 2.

Les paramètres du système maître sont : $a_1 = 10, b_1 = 8/3 + 0.4, c_1 = 28$, et ceux de l'esclave : $a_2 = 9, b_2 = 8/3 + 0.2, c_2 = 27$. Les valeurs initiales des deux systèmes sont : $x_1(0) = 2, x_2(0) = 10, x_3(0) = -6$ and $y_1(0) = -2, y_2(0) = 5, y_3(0) = 1$. Et ceux des para-

mètres estimés sont : $\hat{a}_1(0) = 2, \hat{b}_1(0) = 10, \hat{c}_1(0) = -6, \hat{a}_2(0) = -2, \hat{b}_2(0) = 5, \hat{c}_2(0) = 1$.

Les résultats des simulations sont représentés par Fig. 4.8 qui montre bien la synchronisation hybride entre les deux systèmes. Fig. 4.9 montre la variation des erreurs alors que Fig. 4.10 représente la variation des paramètres estimés..

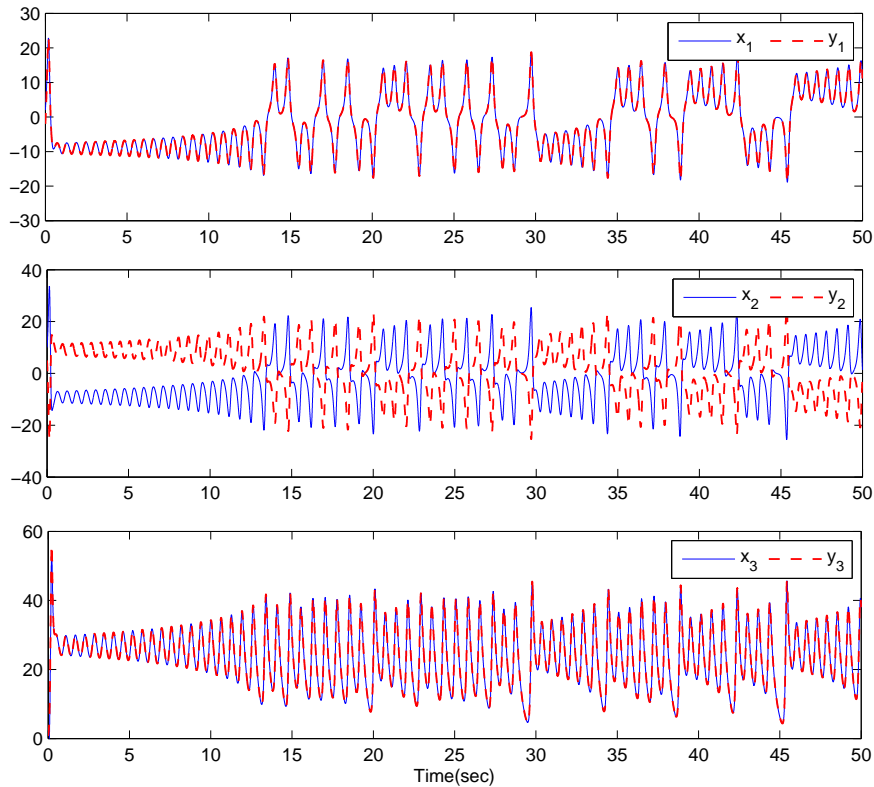


FIGURE 4.8 Synchronisation hybride de deux systèmes de Lorenz identiques en utilisant EOP.

4.3 La synchronisation hybride en utilisant la commande NNGPC

Comme mentionné dans le chapitre 3, l'algorithme NNGPC se compose de trois parties principales : le modèle NN du système, la fonction de coût et l'algorithme d'optimisation. Dans ce chapitre, nous allons vérifier les performances des NNGPC pour la synchronisation hybride des systèmes chaotiques incertains. La seule différence entre la synchronisation CS et la synchronisation hybride réside dans les facteurs du système d'erreur. Par conséquent,

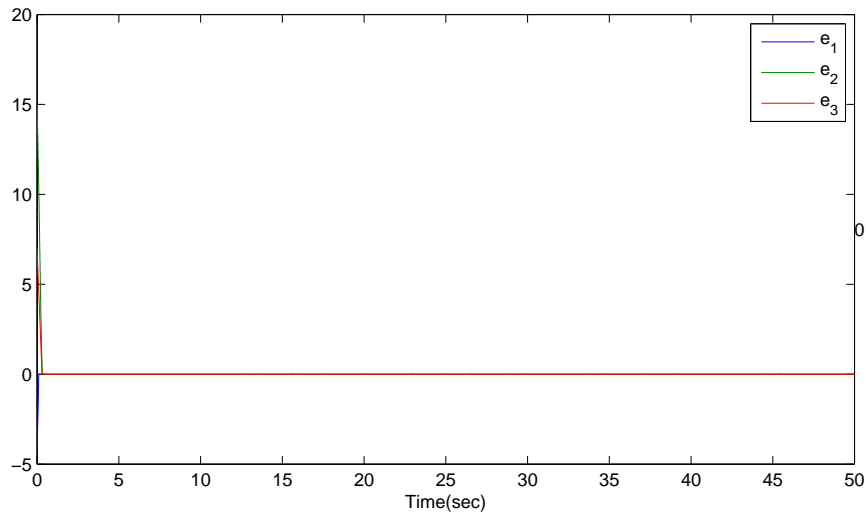


FIGURE 4.9 L'évolution des fonctions d'erreur e_1, e_2, e_3 de deux systèmes de Lorenz identiques en utilisant EOP.

les fonctions paires de la fonction coût sont définies par :

$$J(k) = \sum_{j=1}^2 [y_m(k+j) + x_m(k+j)]^2 \quad (4.38)$$

Et les fonctions impaires par :

$$J(k) = \sum_{j=1}^2 [y_m(k+j) - x_m(k+j)]^2 \quad (4.39)$$

Pour la simulation, on utilise les mêmes systèmes, paramètres et conditions initiales que ceux utilisés dans la section précédente. Fig. 4.11 représente les résultats de la synchronisation d'hybride entre les deux systèmes lorsque le contrôle est appliqué à partir de $t = 5$ et Fig. 4.12 montre la variation du système d'erreur.

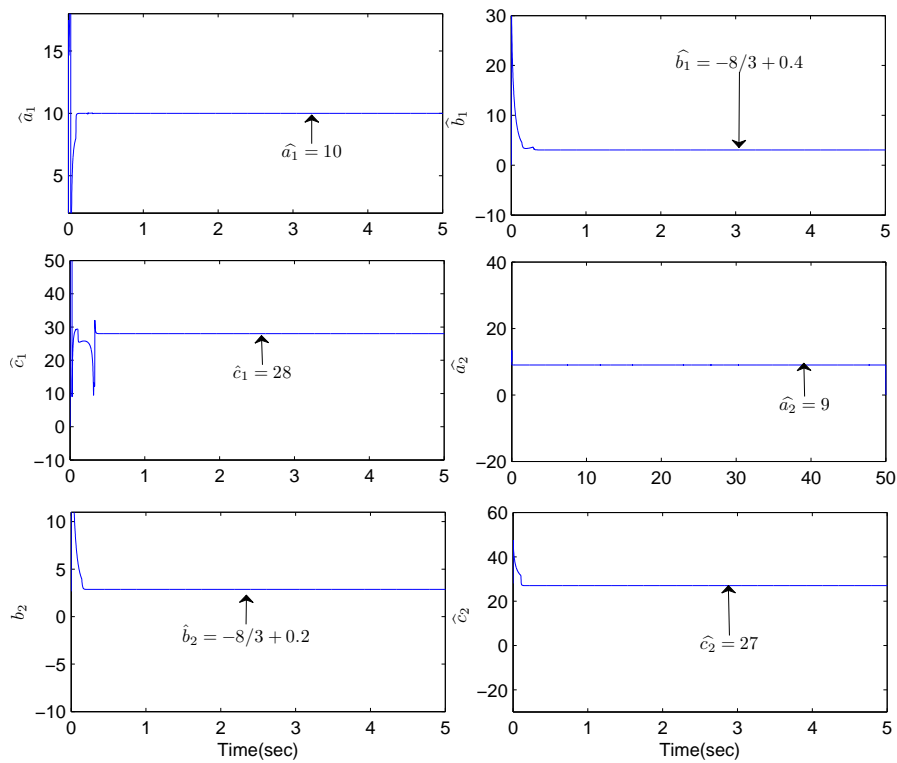


FIGURE 4.10 L'identification des paramètres en utilisant EOP.

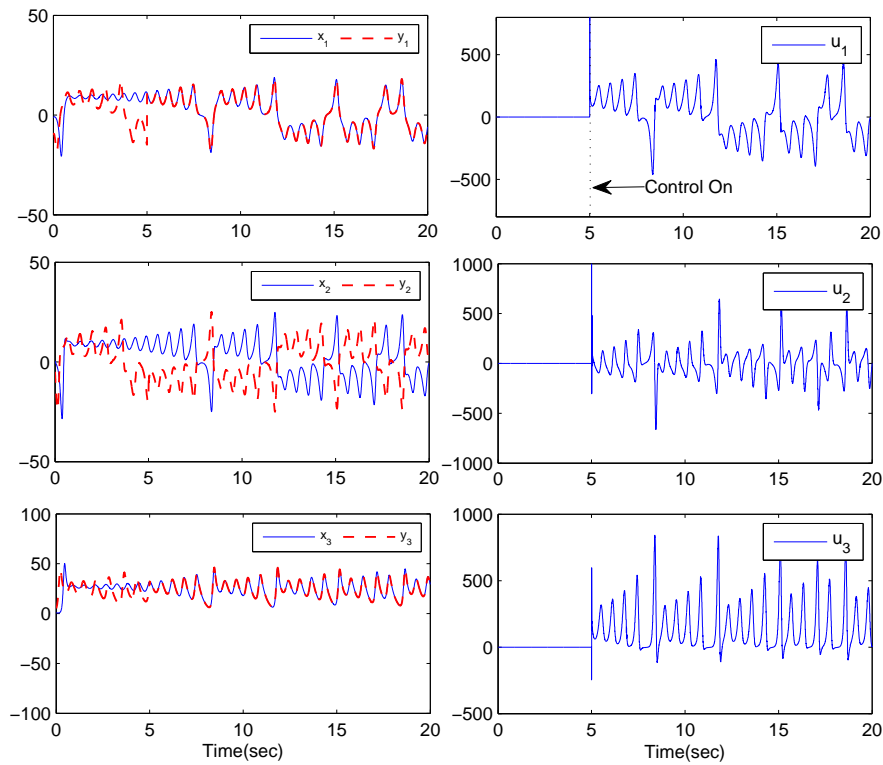


FIGURE 4.11 Synchronisation hybride de deux systèmes de Lorenz identiques utilisant NNGPC.

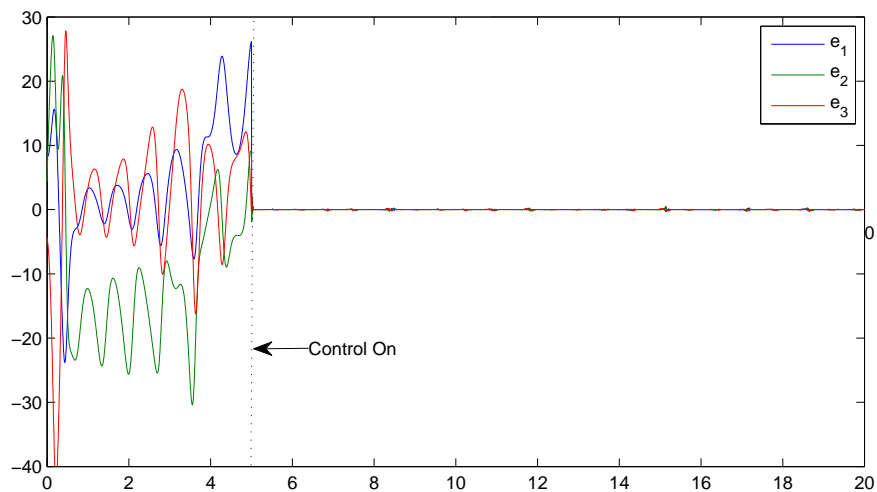


FIGURE 4.12 L'évolution des fonctions d'erreur e_1, e_2, e_3 de deux identiques Lorenz systèmes NGPC.

4.4 Conclusion

Dans ce chapitre, nous avons étendu l'étude réalisée dans les chapitres 2 et 3 pour la synchronisation hybride des systèmes chaotiques incertains. L'effet de la synchronisation hybride sur le problème de la condition LI est partiel comparativement avec la synchronisation multi-commutation. La synchronisation hybride ne permet d'inverser l'orbite des états paires en multipliant les états dans le système maître par -1 . Cette action permet seulement de briser la dépendance linéaire de certains termes, qui aident les lois d'adaptation pour estimer les paramètres réels des systèmes. De plus, l'élimination des termes de perturbation, permet d'obtenir une estimation précise des paramètres des deux systèmes.

L'algorithme NNGPC réalise la synchronisation hybride en raison de sa capacité à conserver les mêmes performances au cours du processus de synchronisation, même si les paramètres ne sont pas connus et variable.

Conclusion générale

En tant que partie centrale de cette étude, nous considérons l'effet de la synchronisation hybride des systèmes chaotiques par une méthode de contrôle. Au début, nous avons souligné l'importance de la synchronisation du chaos, et les principales caractéristiques des systèmes chaotiques. Ensuite, nous avons vérifié les performances de certaines méthodes de contrôle avancées telles que la technique adaptative avec méthode de contrôle active et la commande prédictive généralisée basée sur les contrôleurs PID flous et réseau neuronal. À la première étape, les méthodes de contrôle étudiées sont vérifiées pour une synchronisation complète, après nous avons étendu l'étude à la synchronisation hybride.

La synchronisation des systèmes chaotiques a atteint un point de saturation, dans lequel presque toutes les méthodes de contrôle ont été utilisées. Ce qui impose aux chercheurs de trouver de nouveaux problèmes dont les solutions ouvriraient de nouveaux horizons. Le problème de la condition de la condition d'indépendance linéaire abordé dans le chapitre 2 est un exemple de la façon dont le problème de la synchronisation des systèmes chaotiques incertains peut affecter l'estimation des paramètres. Nous pouvons remarquer que les améliorations proposées sont axées sur la résolution du problème d'estimation plutôt que celui d'atteindre la synchronisation complète des systèmes. Par exemple, en changeant le type de synchronisation, en utilisant la synchronisation multi commutation ou la synchronisation QS, le problème peut être résolu. Le problème la condition d'indépendance linéaire peut aussi être résolu en éliminant les termes de perturbation au lieu d'adapter le dispositif de commande. Comme mentionné dans le chapitre 1, le chaos peut apparaître dans des systèmes très simples et les problèmes de synchronisation relevés peuvent également apparaître dans les cas les plus simples. Par conséquent, notre attention ne doit pas être toujours portée sur des systèmes complexes afin de trouver des problèmes intéressants.

L'imprévisibilité des systèmes chaotiques ouvre la possibilité de l'émergence de certains problèmes engendrés par l'utilisation de la commande prédictive, pour résoudre le problème

de la synchronisation des systèmes chaotiques incertains. Par conséquent, l'utilisation d'une méthode de commande prédictive basée sur un modèle comme élément essentiel dans l'algorithme n'est pas une bonne idée. Par conséquent, la méthode de contrôle PID floue prédictive a été utilisée. Cette approche est intéressante, mais le rôle du modèle linéaire n'est pas clair. Cette méthode peut être considérée comme une exception parce qu'elle brise l'idée courante sur la commande prédictive basée sur un modèle. Lorsque l'une des méthodes de contrôle prédictif à base de modèles, tels que la NNGPC, est utilisée, la synchronisation est atteinte sans aucun problème. Cependant, la précision du modèle est très importante, si les modèles ne sont pas exacts, la synchronisation sera détruite, ce qui est complètement à l'opposé de la régulation PID floue prédictive. En général, la commande prédictive peut faire face à la synchronisation des systèmes chaotiques incertains.

Parfois, les types de synchronisation peuvent être utilisés pour résoudre le problème de la condition LI tel que discuté au chapitre 2. Toutefois, la synchronisation hybride peut résoudre partiellement le problème de l'état de LI si la compare avec la synchronisation QS ou la synchronisation multi-commutation. En utilisant la commande prédictive, la synchronisation hybride n'a pas d'effet clair sur la performance de la méthode NNGPC étudié dans le chapitre 3.

Bien que la synchronisation hybride des systèmes chaotiques utilisant la rétroaction soit un sujet intéressant, il doit être approfondi pour pouvoir conclure quant à son application pour la conception du système de chiffrement numérique ou d'un protocole d'échange de clés. Comme perspective à ce travail, nous projetons d'améliorer la méthode de contrôle étudiée pour traiter le problème des systèmes chaotiques avec time-delay, et d'ordre fractionnaire. Nous allons également essayer de trouver une solution au problème de la régulation PID floue prédictive en termes du rôle du modèle, puis étendre l'étude à d'autres nouvelles techniques de la méthode de commande prédictive.

Bibliographie

- [1] Edward Ott, Celso Grebogi, and James A Yorke. Controlling chaos. *Physical review letters*, 64(11) :1196, 1990.
- [2] Guanrong Chen and Xiaoning Dong. *Methodologies, Perspectives and Applications*. World Scientific, 1998.
- [3] Chyun-Chau Fuh and Pi-Cheng Tung. Controlling chaos using differential geometric method. *Physical Review Letters*, 75(16) :2952, 1995.
- [4] Guan-rong Chen and Xiao-ning Dong. On feedback control of chaotic continuous-time systems. *IEEE Transactions on Circuits and Systems I : Fundamental Theory and Applications*, 40(9) :591–601, 1993.
- [5] Louis M Pecora and Thomas L Carroll. Synchronization in chaotic systems. *Physical review letters*, 64(8) :821, 1990.
- [6] Muthusamy Lakshmanan and Kallummal Murali. *Chaos in nonlinear oscillators : controlling and synchronization*, volume 13. World scientific, 1996.
- [7] Bernd Blasius, Amit Huppert, and Lewi Stone. Complex dynamics and phase synchronization in spatially extended ecological systems. *Nature*, 399(6734) :354–359, 1999.
- [8] Seung Kee Han, Christian Kurrer, and Yoshiki Kuramoto. Dephasing and bursting in coupled neural oscillators. *Physical Review Letters*, 75(17) :3190, 1995.
- [9] Kevin M Cuomo and Alan V Oppenheim. Circuit implementation of synchronized chaos with applications to communications. *Physical review letters*, 71(1) :65, 1993.
- [10] Lj Kocarev and U Parlitz. General approach for chaotic synchronization with applications to communication. *Physical review letters*, 74(25) :5028, 1995.
- [11] K Murali and M Lakshmanan. Secure communication using a compound signal using sampled-data feedback. *Applied Mathematics and Mechanics*, 11 :1309–1315, 2003.
- [12] Hongjie Yu and Yanzhu Liu. Chaotic synchronization based on stability criterion of linear systems. *Physics Letters A*, 314(4) :292–298, 2003.
- [13] Michael G Rosenblum, Arkady S Pikovsky, and Jürgen Kurths. From phase to lag synchronization in coupled chaotic oscillators. *Physical Review Letters*, 78(22) : 4193, 1997.

- [14] Gamal M Mahmoud and Emad E Mahmoud. Phase and antiphase synchronization of two identical hyperchaotic complex nonlinear systems. *Nonlinear Dynamics*, 61(1) : 141–152, 2010.
- [15] Zuo-Lei Wang and Xue-Rong Shi. Anti-synchronization of liu system and lorenz system with known or unknown parameters. *Nonlinear Dynamics*, 57(3) :425–430, 2009.
- [16] Chin Yi Chee and Daolin Xu. Secure digital communication using controlled projective synchronisation of chaos. *Chaos, Solitons & Fractals*, 23(3) :1063–1070, 2005.
- [17] Guo-Hui Li. Modified projective synchronization of chaotic system. *Chaos, Solitons & Fractals*, 32(5) :1786–1790, 2007.
- [18] K Sebastian Sudheer and M Sabir. Hybrid synchronization of hyperchaotic lu system. *PRAMANA-journal of physics*, 73 :781–786, 2009.
- [19] Ronnie Mainieri and Jan Rehacek. Projective synchronization in three-dimensional chaotic systems. *Physical Review Letters*, 82(15) :3042, 1999.
- [20] Sha Wang, Yongguang Yu, and Guoguang Wen. Hybrid projective synchronization of time-delayed fractional order chaotic systems. *Nonlinear Analysis : Hybrid Systems*, 11 :129–138, 2014.
- [21] Hassan Salarieh and Mohammad Shahrokhi. Adaptive synchronization of two different chaotic systems with time varying unknown parameters. *Chaos, Solitons & Fractals*, 37(1) :125–136, 2008.
- [22] M Mossa Al-Sawalha and Mohd Salmi Md Noorani. Adaptive anti-synchronization of two identical and different hyperchaotic systems with uncertain parameters. *Communications in Nonlinear Science and Numerical Simulation*, 15(4) :1036–1047, 2010.
- [23] Xian-Feng Li, Andrew Chi-Sing Leung, Xiu-Ping Han, Xiao-Jun Liu, and Yan-Dong Chu. Complete (anti-) synchronization of chaotic systems with fully uncertain parameters by adaptive control. *Nonlinear Dynamics*, 63(1) :263–275, 2011.
- [24] Fei Yu, Chunhua Wang, Yan Hu, and Jinwen Yin. Antisynchronization of a novel hyperchaotic system with parameter mismatch and external disturbances. *Pramana, J. Phys*, 79(1) :81–93, 2012.
- [25] MT Yassen. Chaos synchronization between two different chaotic systems using active control. *Chaos, Solitons & Fractals*, 23(1) :131–140, 2005.
- [26] Xiaoxin Liao and Guanrong Chen. Chaos synchronization of general lur'e systems via time-delay feedback control. *International Journal of Bifurcation and Chaos*, 13(01) :207–213, 2003.
- [27] Wu Xiao-Qun and Lu Jun-An. Parameter identification and backstepping control of uncertain lü system. *Chaos, Solitons & Fractals*, 18(4) :721–729, 2003.

- [28] Diyi Chen, Runfan Zhang, Xiaoyi Ma, and Si Liu. Chaotic synchronization and anti-synchronization for a novel class of multiple chaotic systems via a sliding mode control scheme. *Nonlinear Dynamics*, 69(1) :35–55, 2012.
- [29] Wafaa Jawaada, Mohd Salmi Md Noorani, and M Mossa Al-sawalha. Robust active sliding mode anti-synchronization of hyperchaotic systems with uncertainties and external disturbances. *Nonlinear Analysis : Real World Applications*, 13(5) :2403–2413, 2012.
- [30] Yi-You Hou, Teh-Lu Liao, and Jun-Juh Yan. H synchronization of chaotic systems using output feedback control design. *Physica A : Statistical Mechanics and Its Applications*, 379(1) :81–89, 2007.
- [31] Diyi Chen, Weili Zhao, Julien Clinton Sprott, and Xiaoyi Ma. Application of takagi-sugeno fuzzy model to a class of chaotic synchronization and anti-synchronization. *Nonlinear Dynamics*, 73(3) :1495–1505, 2013.
- [32] Thomas L Carroll and Louis M Pecora. Synchronizing chaotic circuits. *IEEE transactions on circuits and systems*, 38(4) :453–456, 1991.
- [33] Michael G Rosenblum, Arkady S Pikovsky, and Jürgen Kurths. Phase synchronization of chaotic oscillators. *Physical review letters*, 76(11) :1804, 1996.
- [34] S Boccaletti and DL Valladares. Characterization of intermittent lag synchronization. *Physical Review E*, 62(5) :7497, 2000.
- [35] Michael A Zaks, Eun-Hyoung Park, Michael G Rosenblum, and Jürgen Kurths. Alternating locking ratios in imperfect phase synchronization. *Physical review letters*, 82(21) :4228, 1999.
- [36] Daolin Xu, Wee Leng Ong, and Zhigang Li. Criteria for the occurrence of projective synchronization in chaotic systems of arbitrary dimension. *Physics Letters A*, 305(3) :167–172, 2002.
- [37] Dibakar Ghosh. Generalized projective synchronization in time-delayed systems : Nonlinear observer approach. *Chaos : An interdisciplinary journal of nonlinear science*, 19(1) :013102, 2009.
- [38] Dibakar Ghosh and Sikha Bhattacharya. Projective synchronization of new hyperchaotic system with fully unknown parameters. *Nonlinear Dynamics*, 61(1) :11–21, 2010.
- [39] Manfeng Hu, Zhenyuan Xu, and Rong Zhang. Full state hybrid projective synchronization in continuous-time chaotic (hyperchaotic) systems. *Communications in Nonlinear Science and Numerical Simulation*, 13(2) :456–464, 2008.
- [40] Chuandong Li, Qian Chen, and Tingwen Huang. Coexistence of anti-phase and complete synchronization in coupled chen system via a single variable. *Chaos, Solitons & Fractals*, 38(2) :461–464, 2008.

- [41] Kevin M Cuomo, Alan V Oppenheim, and Steven H Isabelle. Spread spectrum modulation and signal masking using synchronized chaotic systems. Technical report, DTIC Document, 1992.
- [42] Alan V Oppenheim, Gregory W Wornell, Steven H Isabelle, and Kevin M Cuomo. Signal processing in the context of chaotic signals. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 4, pages 117–120. IEEE, 1992.
- [43] K Murali and M Lakshmanan. Transmission of signals by synchronization in a chaotic van der pol–duffing oscillator. *Physical Review E*, 48(3) :R1624, 1993.
- [44] Leon O Chua, Makoto Itoh, Ljupco Kocarev, and Kevin Eckert. Chaos synchronization in chua’s circuit. *Journal of Circuits, Systems, and Computers*, 3(01) :93–108, 1993.
- [45] Gabriel Pérez and Hilda A Cerdeira. Extracting messages masked by chaos. *Physical Review Letters*, 74(11) :1970, 1995.
- [46] Kevin M Short. Steps toward unmasking secure communications. *International Journal of Bifurcation and Chaos*, 4(04) :959–977, 1994.
- [47] Ljupco Kocarev. Chaos synchronization of high-dimensional dynamical systems. In *Circuits and Systems, 1995. ISCAS’95., 1995 IEEE International Symposium on*, volume 2, pages 1009–1012. IEEE, 1995.
- [48] Ulrich Parlitz, Leon O Chua, Lj Kocarev, KS Halle, and A Shang. Transmission of digital signals by chaotic synchronization. *International Journal of Bifurcation and Chaos*, 2(04) :973–977, 1992.
- [49] Kevin M Cuomo, Alan V Oppenheim, and Steven H Strogatz. Synchronization of lorenz-based chaotic circuits with applications to communications. *IEEE Transactions on circuits and systems II : Analog and digital signal processing*, 40(10) :626–633, 1993.
- [50] Ying-Cheng Lai and Celso Grebogi. Synchronization of chaotic trajectories using control. *Physical Review E*, 47(4) :2357, 1993.
- [51] Chien-Chong Chen. Direct chaotic dynamics to any desired orbits via a closed-loop control. *Physics Letters A*, 213(3-4) :148–154, 1996.
- [52] Mario Di Bernardo. A purely adaptive controller to synchronize and control chaotic systems. *Physics Letters A*, 214(3-4) :139–144, 1996.
- [53] Günyaz Ablay and Tunc Aldemir. Synchronization of different chaotic systems using generalized active control. In *ELECO’6th International Conference on Electrical and Electronics Engineering*, 2009.
- [54] MT Yassen. Controlling chaos and synchronization for new chaotic system using linear feedback control. *Chaos, Solitons & Fractals*, 26(3) :913–920, 2005.

- [55] Xiaohui Tan, Jiye Zhang, and Yiren Yang. Synchronizing chaotic systems using backstepping design. *Chaos, Solitons & Fractals*, 16(1) :37–45, 2003.
- [56] Tomasz Kapitaniak, Marek Sekieta, and Maciej Ogorzalek. Monotone synchronization of chaos. *International Journal of Bifurcation and Chaos*, 6(01) :211–217, 1996.
- [57] Ricardo Femat, José Alvarez-Ramírez, and Guillermo Fernández-Anaya. Adaptive synchronization of high-order chaotic systems : a feedback with low-order parametrization. *Physica D : Nonlinear Phenomena*, 139(3) :231–246, 2000.
- [58] Yongguang Yu and Suochun Zhang. Adaptive backstepping synchronization of uncertain chaotic system. *Chaos, Solitons & Fractals*, 21(3) :643–649, 2004.
- [59] Her-Terng Yau. Design of adaptive sliding mode controller for chaos synchronization with uncertainties. *Chaos, Solitons & Fractals*, 22(2) :341–347, 2004.
- [60] Her-Terng Yau and Cheng-Shion Shieh. Chaos synchronization using fuzzy logic controller. *Nonlinear analysis : Real world applications*, 9(4) :1800–1810, 2008.
- [61] Mou Chen and Wen-hua Chen. Robust adaptive neural network synchronization controller design for a class of time delay uncertain chaotic systems. *Chaos, Solitons & Fractals*, 41(5) :2716–2724, 2009.
- [62] Chao-Lin Kuo, Neng-Sheng Pai, and Her-Terng Yau. Adaptive fuzzy sliding mode controller design for lorenz system. In *Chaos-Fractals Theories and Applications, 2009. IWCFTA'09. International Workshop on*, pages 122–125. IEEE, 2009.
- [63] Da Lin, Xingyuan Wang, Fuzhong Nian, and Yonglei Zhang. Dynamic fuzzy neural networks modeling and adaptive backstepping tracking control of uncertain chaotic systems. *Neurocomputing*, 73(16) :2873–2881, 2010.
- [64] Choon Ki Ahn. T-s fuzzy synchronization for chaotic systems via delayed output feedback control. *Nonlinear Dynamics*, 59(4) :535–543, 2010.
- [65] Haipeng Su, Runzi Luo, and Yanhui Zeng. The exponential synchronization of a class of fractional-order chaotic systems with discontinuous input. *Optik-International Journal for Light and Electron Optics*, 2016.
- [66] P Balasubramaniam and P Muthukumar. Synchronization of chaotic systems using feedback controller : An application to diffie–hellman key exchange protocol and elgamal public key cryptosystem. *Journal of the Egyptian Mathematical Society*, 22(3) :365–372, 2014.
- [67] P Muthukumar and P Balasubramaniam. Feedback synchronization of the fractional order reverse butterfly-shaped chaotic system and its application to digital cryptography. *Nonlinear Dynamics*, 74(4) :1169–1181, 2013.
- [68] Juhn-Horng Chen, Hsien-Keng Chen, and Yu-Kai Lin. Synchronization and anti-synchronization coexist in chen–lee chaotic systems. *Chaos, Solitons & Fractals*, 39(2) :707–716, 2009.

- [69] Wen Sun, Zhong Chen, Yibing Lü, and Shihua Chen. An intriguing hybrid synchronization phenomenon of two coupled complex networks. *Applied Mathematics and Computation*, 216(8) :2301–2309, 2010.
- [70] Xia Huang, Zhen Wang, and Yuxia Li. Hybrid synchronization of hyperchaotic lü system based on passive control. In *Chaos-Fractals Theories and Applications (IWCFTA), 2010 International Workshop on*, pages 34–38. IEEE, 2010.
- [71] V Sundarapandian and S Sivaperumal. Sliding controller design of hybrid synchronization of four-wing chaotic systems. *International Journal of Soft Computing*, 6(5) : 224–231, 2011.
- [72] Sundarapandian Vaidyanathan and Karthikeyan Rajagopal. Hybrid synchronization of hyperchaotic wang-chen and hyperchaotic lorenz systems by active non-linear control. *International journal of systems signal control & engineering applications*, 4(3) :55–61, 2011.
- [73] Sundarapandian Vaidyanathan. Hybrid chaos synchronization of liu and lü systems by active nonlinear control. In *Trends in Computer Science, Engineering and Information Technology*, pages 1–10. Springer, 2011.
- [74] Sundarapandian Vaidyanathan and Suresh Rasappan. Hybrid synchronization of hyperchaotic qi and lü systems by nonlinear control. In *International Conference on Computer Science and Information Technology*, pages 585–593. Springer, 2011.
- [75] V Sundarapandian. Hybrid synchronization of lorenz and pehlivan chaotic systems by active nonlinear control. *International Journal of Advances in Science and Technology*, 3(2) :7–12, 2011.
- [76] Sundarapandian Vaidyanathan. Hybrid chaos synchronization of hyperchaotic newton—leipnik systems by sliding mode control. *International Journal of Control Theory and Computer Modelling*, 1(2) :1–10, 2011.
- [77] V Sundarapandian. Hybrid synchronization of hyperchaotic rössler and hyperchaotic lorenz systems by active control. *Internat. J. Advances in Science and Technology*, 2 (4) :1–10, 2011.
- [78] V Sundarapandian. Hybrid synchronization of liu and chen systems by active non-linear control. *Int. J. of Mathematical Sciences and Applications*, 1(3) :1139–1146, 2011.
- [79] Suresh Rasappan and Sundarapandian Vaidyanathan. Hybrid synchronization of n-scroll chua and lur’e chaotic systems via backstepping control with novel feedback. *Archives of Control Sciences*, 22(3) :343–365, 2012.
- [80] Xuerong Shi and Zuolei Wang. The alternating between complete synchronization and hybrid synchronization of hyperchaotic lorenz system with time delay. *Nonlinear Dynamics*, 69(3) :1177–1190, 2012.
- [81] Xiaobing Zhou, Zhangbiao Fan, Dongming Zhou, and Xiaomei Cai. Passivity-based adaptive hybrid synchronization of a new hyperchaotic system with uncertain parameters. *The Scientific World Journal*, 2012, 2012.

- [82] Zhen Zheng, Manchun Tan, and Qunfang Wang. Hybrid synchronization of two delayed systems with uncertain parameters. In *International Symposium on Neural Networks*, pages 285–292. Springer, 2012.
- [83] SK Agrawal, M Srivastava, and S Das. Hybrid synchronization between different fractional order hyperchaotic systems using active control method. *J. Nonlinear Syst. Appl*, 4(3) :70–76, 2013.
- [84] Sundarapandian Vaidyanathan. Hybrid chaos synchronization of 3-cells cellular neural network attractors via adaptive control method. *International Journal of Pharm-Tech Research*, 8(8) :61–73, 2015.
- [85] Sundarapandian Vaidyanathan. Adaptive synchronization of generalized lotka-volterra three-species biological systems. *Int J PharmTech Res*, 8(5) :928–937, 2015.
- [86] Sivaperumal Sampath and Sundarapandian Vaidyanathan. Hybrid synchronization of identical chaotic systems via novel sliding control method with application to sampath four-scroll chaotic system. *International Journal of Control Theory and Applications*, 9(1) :221–235, 2016.
- [87] U Parlitz. Estimating model parameters from time series by autosynchronization. *Physical Review Letters*, 76(8) :1232, 1996.
- [88] Zhiyong Sun, Wenzhi Zhu, Gangquan Si, Yue Ge, and Yanbin Zhang. Adaptive synchronization design for uncertain chaotic systems in the presence of unknown system parameters : a revisit. *Nonlinear Dynamics*, 72(4) :729–749, 2013.
- [89] K Sebastian Sudheer and M Sabir. Adaptive modified function projective synchronization between hyperchaotic lorenz system and hyperchaotic lu system with uncertain parameters. *Physics Letters A*, 373(41) :3743–3748, 2009.
- [90] Xing-Yuan Wang and Peng Sun. Multi-switching synchronization of chaotic system with adaptive controllers and unknown parameters. *Nonlinear Dynamics*, 63(4) : 599–609, 2011.
- [91] Hong-Tao Lu Xiang-Jun Wu. Generalized projective lag synchronization between different hyperchaotic systems with uncertain parameters. *Nonlinear Dynamics*, 66 (1) :185–200, 2011.
- [92] Chi-Ching Yang. Adaptive synchronization of lü hyperchaotic system with uncertain parameters based on single-input controller. *Nonlinear Dynamics*, 63(3) :447–454, 2011.
- [93] E. Saleh M. M. El-Dessoky, M. T. Yassen. Adaptive modified function projective synchronization between two different hyperchaotic dynamical systems. *Math.Probl.Eng*, 2012(810626) :16, 2012.
- [94] Zakaria Driss and Noura Mansouri. A novel adaptive approach for synchronization of uncertain chaotic systems using fuzzy pi controller and active control method. *Journal of Control Engineering and Applied Informatics*, 18(3) :3–13, 2016.

- [95] Guanrong Chen and Hao Ying. Stability analysis of nonlinear fuzzy pi control systems. In *Industrial Fuzzy Control and Intelligent Systems, 1993., IFIS'93., Third International Conference on*, pages 128–133. IEEE, 1993.
- [96] Dave Misir, Heidar A Malki, and Guanrong Chen. Design and analysis of a fuzzy proportional-integral-derivative controller. *Fuzzy sets and systems*, 79(3) :297–314, 1996.
- [97] KS Tang, Kim Fung Man, Guanrong Chen, and Sam Kwong. An optimal fuzzy pid controller. *Industrial Electronics, IEEE Transactions on*, 48(4) :757–765, 2001.
- [98] Radu-Emil Precup, Marius-Lucian Tomescu, and Claudia-Adina Dragos. Stabilization of rössler chaotic dynamical system using fuzzy logic control algorithm. *International Journal of General Systems*, 43(5) :413–433, 2014.
- [99] IM Ginarsa, A Soeprijanto, and MH Purnomo. Controlling chaos and voltage collapse using an anfis-based composite controller-static var compensator in power systems. *International Journal of Electrical Power & Energy Systems*, 46 :79–88, 2013.
- [100] Maryam Sadeghi and Bagher Menhaj. Controlling chaos in a defined trajectory using adaptive fuzzy logic algorithm. In *NUMERICAL ANALYSIS AND APPLIED MATHEMATICS ICNAAM 2012 : International Conference of Numerical Analysis and Applied Mathematics*, volume 1479, pages 2237–2240. AIP Publishing, 2012.
- [101] Zakaria Driss and Noura Mansouri. Synchronization of uncertain chaotic systems using generalized predictive control based on fuzzy pid controllers. In *Proceedings of the 7th International Joint Conference on Computational Intelligence*, pages 25–30, 2015. ISBN 978-989-758-157-1. doi : 10.5220/0005581900250030.
- [102] UE Vincent. Synchronization of rikitake chaotic attractor using active control. *Physics Letters A*, 343(1) :133–138, 2005.
- [103] Sachin Bhalekar and Varsha Daftardar-Gejji. Synchronization of different fractional order chaotic systems using active control. *Communications in Nonlinear Science and Numerical Simulation*, 15(11) :3536–3546, 2010.
- [104] David W Clarke, C Mohtadi, and PS Tuffs. Generalized predictive control—part i. the basic algorithm. *Automatica*, 23(2) :137–148, 1987.
- [105] J. Smith. *Process Control Theory and Applications*. Springer-Verlag London, London, 1nd edition, 2004.
- [106] R. Kennel, A. Linder, and M. Linke. Generalized predictive control (gpc)-ready for use in drive applications ? In *Power Electronics Specialists Conference, 2001. PESC. 2001 IEEE 32nd Annual*, volume 4, pages 1839–1844 vol. 4, 2001. doi : 10.1109/PESC.2001.954389.
- [107] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3) :338–353, 1965.
- [108] Dimitri Van De Ville, Mike Nachtegael, Dietrich Van der Weken, Etienne E Kerre, Wilfried Philips, and Ignace Lemahieu. Noise reduction by fuzzy image filtering. *Fuzzy Systems, IEEE Transactions on*, 11(4) :429–436, 2003.

- [109] Peter J Woolf and Yixin Wang. A fuzzy logic approach to analyzing gene expression data. *Physiological Genomics*, 3(1) :9–15, 2000.
- [110] Tamoghna Das and Indra Narayan Kar. Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots. *Control Systems Technology, IEEE Transactions on*, 14(3) :501–510, 2006.
- [111] RAJESWARI KOTHANDARAMAN, Lavanya Satyanarayana, and Lakshmi Ponnusamy. Grey fuzzy sliding mode controller for vehicle suspension system. *Journal of Control Engineering and Applied Informatics*, 17(3) :12–19, 2015.
- [112] Aissa Rebai, Kamel Guesmi, and Boualem Hemici. Design of an optimized fractional order fuzzy pid controller for a piezoelectric actuator. *Journal of Control Engineering and Applied Informatics*, 17(3) :41–49, 2015.
- [113] Vikas Kumar. Novel ai based on-line sequential learning technique for high performance dc servo motor control. *Journal of Control Engineering and Applied Informatics*, 17(2) :3–11, 2015.
- [114] KS Tang, Kim Fung Man, Guanrong Chen, and Sam Kwong. An optimal fuzzy pid controller. *Industrial Electronics, IEEE Transactions on*, 48(4) :757–765, 2001.
- [115] Z. Driss and N. Mansouri. Generalized predictive control based on fuzzy sliding-mode control strategy and fuzzy pi controller for synchronization of uncertain chaotic systems. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, pages 1–5, Aug 2015. doi : 10.1109/FUZZ-IEEE.2015.7337912.
- [116] Mehdi Roopaei, Mansoor Zolghadri Jahromi, Bijan Ranjbar-Sahraei, and Tsung-Chih Lin. Synchronization of two different chaotic systems using novel adaptive interval type-2 fuzzy sliding mode control. *Nonlinear Dynamics*, 66(4) :667–680, 2011.
- [117] Her-Terng Yau and Chieh-Li Chen. Chattering-free fuzzy sliding-mode control strategy for uncertain chaotic systems. *Chaos, Solitons & Fractals*, 30(3) :709–718, 2006.
- [118] Nouredine Bouarroudj. A hybrid fuzzy fractional order pid sliding-mode controller design using pso algorithm for interconnected nonlinear systems. *Journal of Control Engineering and Applied Informatics*, 17(1) :41–51, 2015.
- [119] Jialiang Lu, Guanrong Chen, and Hao Ying. Predictive fuzzy pid control : theory, design and simulation. *Information Sciences*, 137(1) :157–187, 2001.
- [120] Konstantinos E Parsopoulos, Michael N Vrahatis, and IGI Global. *Particle swarm optimization and intelligence : advances and applications*. Information Science Reference Hershey, 2010.
- [121] Russ C Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, volume 1, pages 39–43. New York, NY, 1995.

- [122] Heidar Malki, Huaidong Li, Guanrong Chen, et al. New design and stability analysis of fuzzy proportional-derivative control systems. *Fuzzy Systems, IEEE Transactions on*, 2(4) :245–254, 1994.
- [123] Zakaria Driss and Noura Mansouri. Synchronization of chaotic systems with unknown parameters using predictive fuzzy pid control. In *Computational Intelligence*, pages 216–240. Springer, 2015.
- [124] Zheng Yan and Jun Wang. Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks. *Industrial Informatics, IEEE Transactions on*, 8(4) :746–756, 2012.
- [125] Zenghui Wang and Yanxia Sun. Generalized predictive control based on particle swarm optimization for linear/nonlinear process with constraints. In *Computational Intelligence and Natural Computing Proceedings (CINC), 2010 Second International Conference on*, volume 1, pages 303–306. IEEE, 2010.
- [126] Ying Song, Zengqiang Chen, and Zhuzhi Yuan. New chaotic pso-based neural network predictive control for nonlinear process. *Neural Networks, IEEE Transactions on*, 18(2) :595–601, 2007.
- [127] Javier Causa, Gorazd Karer, Alfredo Núñez, Doris Sáez, Igor Škrjanc, and Borut Zupančič. Hybrid fuzzy predictive control based on genetic algorithms for the temperature control of a batch reactor. *Computers & chemical engineering*, 32(12) : 3254–3263, 2008.
- [128] Julian Mercieca and Simon Fabri. Particle swarm optimization for nonlinear model predictive control. In *ADVCOMP 2011, The Fifth International Conference on Advanced Engineering Computing and Applications in Sciences*, pages 88–93, 2011.
- [129] Sadhana K Chidrawar, Sujata Bhaskarwar, and Balasaheb M Patre. Implementation of neural network for generalized predictive control : a comparison between a newton raphson and levenberg marquardt implementation. In *Computer Science and Information Engineering, 2009 WRI World Congress on*, volume 1, pages 669–673. IEEE, 2009.
- [130] Hai-Rong Sun, Peng Li, and Li-Hui Zhou. A strategy of generalized predictive control based on neural network. 2004.
- [131] D. Soloway and P. J. Haley. Neural generalized predictive control. In *Intelligent Control, 1996., Proceedings of the 1996 IEEE International Symposium on*, pages 277–282, Sep 1996. doi : 10.1109/ISIC.1996.556214.

Abstract

The aim of the thesis is to consider the performance of some control methods towards the problem of hybrid synchronization of uncertain chaotic systems. These systems are interesting due to the sensitive dependence on initial conditions or what is called the butterfly effect; this feature makes chaotic systems exhibit random behavior. Therefore, the synchronization of chaotic systems with uncertain parameters is an attractive problem, especially in the field of secure communication, that has been intensively studied in the last two decades. The problem of uncertainties on the parameters, in the field of nonlinear control systems, has been solved especially when advanced control methods are used. However, the effect of synchronization of two chaotic systems with uncertain parameters and hybrid synchronization type on some advanced control method is still questionable. The thesis is composed of three main points, each of them dealing with the studied problem from a different aspect.

In the first point, we considered the performance of adaptive synchronization in which self-tuning adaptive control and active control method are used. This approach has been used intensively to solve the studied problem, but many problems have been reported, and one of the problems is the wrong estimation of the unknown parameters. This problem occurs when two identical or partially identical chaotic systems are synchronized in which the synchronization prevents the adaptation laws to estimate the true values of the unknown parameters. This problem is considered in chapter 2 of the thesis in which we explained the problem, gave the proposed solutions, and the contribution.

In the second point, predictive control, particularly generalized predictive control (GPC), has been considered. This approach is sort of complex control method in which the accuracy of the model plays a vital role during the control process. Therefore, we examined, in chapter 3, the performance of GPC towards the studied problem. This chapter consists of two parts. Part one focuses on GPC based on fuzzy PID controllers in which we discussed the roll of the linear model. Part two investigates the performance of GPC based on a neural network model (NNGPC).

In the first point and the second point, the synchronization type used is complete synchronization. Therefore, in the third point (chapter 4), we addressed the problem of hybrid synchronization of uncertain chaotic systems in which we investigated the performance of the studied control methods.

In general, the effect of synchronization or hybrid synchronization of uncertain chaotic systems on some advanced control methods is partial. The field of chaos synchronization using feedback control is an independent field of research, but the performance of studied methods may not be satisfying when the field of secure communication is involved.

نبذة مختصرة

ان الهدف من هذه الرسالة هو النظر في أداء بعض أساليب التحكم لحل مشكلة التزامن الهجين لأنظمة الفوضى الغير مؤكدة. هذه الانظمة مثيرة للاهتمام بسبب الحساسية الشديدة للقيم الأولية أو ما يسمى بتأثير الفراشة. هذه الميزة تجعل أنظمة الفوضى تسلك سلوك عشوائي، لذلك فإن تزامن النظم الفوضوية مع البارامترات الغير مؤكدة هي مشكلة مهمة، خاصة في مجال الاتصالات الآمنة، التي تم دراستها بشكل مكثف في العقدين الماضيين. ان مشكلة البارامترات الغير مؤكدة تم حلها في مجال أنظمة التحكم اللاخطية، خاصة عند استخدام أساليب التحكم المتقدمة. ومع ذلك، فإن تأثير تزامن نظامين من الفوضى مع بارامترات غير مؤكدة و مع نوع التزامن الهجين على بعض أساليب التحكم المتقدمة لا يزال موضع شك. تتكون الأطروحة من ثلاث نقاط رئيسية، كل منها يتناول المشكلة المدروسة من جانب مختلف.

في النقطة الأولى، نظرنا في أداء المزامنة التكميلية التي يتم فيها استعمال التحكم التكميلي ذو التعديل الذاتي وطريقة التحكم النشط. وقد استخدم هذا النهج بشكل مكثف لحل المشكلة المدروسة، ولكن تم الإبلاغ عن العديد من المشاكل، و واحدة من هذه المشاكل هي مشكلة التقدير الخاطئ للبارامترات الغير معروفة. تحدث هذه المشكلة عندما تتم مزامنة نظامين متطابقين أو متطابقين جزئياً، حيث يمنع التزامن قوانين التكيف من تقدير القيم الحقيقية للبارامترات الغير معروفة. وقد تم النظر في هذه المشكلة في الفصل الثاني من الرسالة التي شرحنا فيها المشكلة، وقدمنا الحلول المقترحة والمساهمة.

في النقطة الثانية، تم النظر في طريقة التحكم التنبؤي، وخاصة التحكم التنبؤي المعمم (ت.ت.م). هذه الطريقة هي نوع من الطرق التحكم المعقدة التي يلعب فيها دقة النموذج دوراً حيوياً أثناء عملية التحكم. لذلك فقد بحثنا في الفصل الثالث أداء هذا النمط من المتحكمات لحل المشكلة المدروسة. يتكون هذا الفصل من جزأين. يركز الجزء الأول على ت.ت.م على أساس وحدات تحكم ب.ب.د الغامضة التي ناقشنا فيها دور النموذج الخطي، وناقش الجزء الثاني أداء ت.ت.م على أساس نموذج الشبكة العصبية.

ان نوع المزامنة المستخدمة في النقطة الأولى والنقطة الثانية هي المزامنة الكاملة، لذلك في النقطة الثالثة (الفصل 4)، عالجتنا مشكلة التزامن الهجين لأنظمة الفوضى الغير مؤكدة التي قمنا فيها بالتحقيق في أداء أساليب التحكم المدروسة.

وبوجه عام، إن تأثير التزامن الكامل أو التزامن الهجين لأنظمة الفوضى الغير مؤكدة على بعض أساليب التحكم المتقدمة هو تأثير جزئي، زيادتنا على ذلك فإن مجال تزامن أنظمة الفوضى باستخدام التحكم اللاخطي هو حقل بحث مستقل، ولكن أداء الطرق المدروسة قد تختلف عند استعمالها في مجال الاتصالات الآمنة.