

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'enseignement supérieur et de la recherche scientifique

Université Mohammed Seddik Benyahia- Jijel
Faculté des Sciences et de la Technologie
Département d'Electronique



THESE

Présentée pour *l'obtention du diplôme de*

Doctorat en sciences

Spécialité : **ELECTRONIQUE**

Présentée par

Mme. Hana BOUDJEDIR ep BOULASSEL

Intitulée

COMMANDE ADAPTATIVE PAR LES APPROXIMATEURS

UNIVERSELS DES SYSTEMES NON LINEAIRES

Présentée le: **13 Juin 2015**

Devant le jury :

| | | | |
|--------------|---------------------------|-------------------------|-----------------------|
| Président : | Abdelkrim BOUKABOU | Professeur | Université de Jijel |
| Rapporteur : | Omar BOUHALI | Maître de Conférences A | Université de Jijel |
| Examineurs : | Khaled KHELIL | Professeur | Université Souk Ahras |
| | Sabrina AOUAOUDA | Maître de Conférences A | Université Souk Ahras |

ملخص :

تطور العديد من مخططات التحكم التلاؤمي بالشبكات العصبية الاصطناعية بوجود أو عدم وجود إعادة تكوين الحالة لحل العديد من المسائل هو الهدف الأساسي لهذه الأطروحة. وصف عام للمقرب الشامل المستخدم في هذه الأطروحة "شبكة العصبونات اصطناعية" تم اعطاؤه اضافة الى خوارزمية تدريبيه الانتشار الخلفي. المفاهيم الأساسية و نظريات الاستقرار لأنظمة لا خطية اضافة الى تقديم ثلاث تقنيات تحكم استقراريه مع مزايا و عيوب كل واحدة تم اعطاؤهم. أول مخطط تحكم تلاؤمي في هذه الأطروحة تم اقتراحه بهدف حل مشكل ال التحت محركية لطائرة الهليكوبتر بأربعة. استراتيجية التحكم تتطلب ثلاث أطوار هم: اضافة تحكيمات افتراضية, تقدير الكميات المجهولة و اقتراح مجهدات هولونومية معتمدين على العلاقات الفيزيائية لنظام. ثاني مخطط تحكم تلاؤمي مقترح يستعمل شبكة العصبونات اصطناعية لتقدير التحكم المعادل المجهول و انتاج كمية المصحح لمراقب الحالة التلاؤمي بهدف إعادة تكوين حالات غير حساسة لضجيج القياس. الهليكوبتر بأربعة محركات هي النموذج المطبق عليه هذه التقنية , تحكيمات افتراضية و مجهدات هولونومية أخرى تم اقتراحها لجعل النموذج من نفس الفئة المدروسة. المخطط الثالث يستخدم أيضا يستعمل شبكة العصبونات اصطناعية لتقدير التحكم المعادل المجهول. غير أن تقدير شعاع الحالة كان بواسطة نظام مراقبة جديد يتكون من مراقبين على تسلسل هما مراقب الريح الكبير تلاؤمي و مرشح كالمان المغاير. الهدف من هذه التقنية هو انتاج شعاع الحالة غير حساس لضجيج القياس. نظرية ليابونوف أستعملت من أجل تحليل الاستقرار مخططات التحكم المقترحة وادائهم قد أختبر بواسطة التجريب.

كلمات مفتاحية :

تحكم تلاؤمي، الشبكة العصبية الاصطناعية، نظم غير الخطية، استقرار ليابونوف، إعادة تكوين الحالة.

Résumé :

Le développement schémas de commande adaptative avec et sans observateur d'état pour résoudre différents problèmes est l'objectif principal de cette thèse. Une description générale sur l'approximateur universel utilisé dans cette thèse « réseaux de neurones artificiels » est donnée avec son algorithme d'apprentissage rétro-propagation. Les notions fondamentales et les théorèmes de la stabilité des systèmes non linéaires avec la présentation de trois techniques de commande stabilisantes avec les avantages et les inconvénients de chacune sont donnés par la suite. Le premier schéma de commande adaptative dans cette thèse est proposé dans le but de résoudre le problème de sous actionniste du véhicule volant « quadrotor ». La stratégie de commande nécessite trois phases sont : l'ajout des commandes virtuelles, l'estimation des termes inconnus et l'utilisation des contraintes holonomes en se basant sur les relations physiques de système. Le deuxième schéma de commande adaptative proposé utilise les réseaux de neurones artificiels pour estimer la commande équivalente inconnue et pour générer le terme de correction d'un observateur d'état adaptatif afin de reconstruire un vecteur d'état insensible aux bruits de mesure. Le quadrotor est le modèle d'application de cette technique, d'où, des nouvelles commandes virtuelles et contraintes holonomes ont été proposées pour rendre le modèle dynamique de quadrotor de la même classe étudiée. Le troisième schéma, utilise aussi les réseaux de neurones artificiels pour approximer la commande équivalente. Cependant, l'estimation de vecteur d'état a été par un nouveau système d'observation composé de deux observateurs en cascade sont : l'observateur à grand gain adaptatif et un filtre de Kalman modifié. L'objectif de cette technique de fournir un vecteur d'état insensible aux bruits de mesure. La théorie de Lyapunov a été utilisée pour analyser la stabilité des schémas de commande proposés et leurs performances ont été testées par simulation.

Mots clés :

Commande adaptative, réseau de neurones artificiels, système non linéaires, stabilité de Lyapunov, Observateur d'état.

Remerciements

Remerciements

Tables des Matières

Table des Matières

| | |
|--------------------------|-----|
| Remerciements | i |
| Liste des figures | v |
| Liste des tableaux | vii |

| | |
|------------------------------------|---|
| Introduction générale | 1 |
|------------------------------------|---|

Chapitre 1 : Réseaux de neurones artificiels « RNA »

| | | |
|---------|---|----|
| 1 | Introduction..... | 5 |
| 2 | Origine biologique des réseaux de neurones artificiels | 6 |
| 2.1. | Neurone biologique | 7 |
| 2.2 | Neurone formel | 7 |
| 2.2.1 | Poids..... | 8 |
| 2.2.2 | Seuil | 8 |
| 2.2.3 | Fonction d'activation | 8 |
| 3 | Développement des réseaux de neurones artificiels et les règles d'apprentissage | 9 |
| 3.1. | Perceptron multicouche | 11 |
| 3.1.1 | Algorithme de retro-propagation..... | 12 |
| 3.1.1.a | Algorithme de gradient | 12 |
| 3.1.1.b | Etapes de retro-propagation | 13 |
| 3.2. | Réseau de neurone à fonctions de base radiale..... | 14 |
| 4 | Approximation neuronale des fonctions | 15 |
| 5 | Identification des systèmes par les réseaux de neurones artificiels | 16 |
| 5.1. | Identification directe | 16 |
| 5.2. | Identification du modèle inverse | 16 |
| 5.3. | Application de l'algorithme rétro-propagation | 17 |
| 5.3.1 | Identification neuronale directe..... | 17 |
| 5.3.2 | Identification neuronale inverse | 19 |
| 6 | Conclusion | 21 |

Chapitre 2 : Notions sur les commandes stabilisantes des systèmes non linéaires

| | | |
|-------|---|----|
| 1 | Introduction..... | 22 |
| 2 | Stabilité des systèmes non linéaires | 23 |
| 2.1 | Téorèmes de stabilité | 24 |
| 2.2 | Stabilité locale et stabilité globale..... | 26 |
| 2.3 | Méthodes de Lyapunov | 27 |
| 2.3.1 | Méthodes indirecte de Lyapunov | 27 |
| 2.3.2 | Méthodes directe de Lyapunov | 28 |
| 3 | Commande par backstepping | 28 |
| 3.1 | Algorithme backstepping | 29 |
| 3.2 | Robustesse de backstepping..... | 30 |
| 4 | Commandes par mode glissant | 31 |
| 4.1 | Conception de la loi de commande par mode de glissant | 31 |
| 4.1.1 | surface de glissement | 31 |
| 4.1.2 | Commande glissante | 32 |
| 4.1.3 | Commande équivalente | 33 |
| 4.2 | Analyse de la stabilité | 33 |

| | | |
|-----|--|----|
| 4.3 | Problèmes de la commande par mode glissant | 34 |
| 5 | Commandes adaptative | 35 |
| 5.1 | Commande adaptative à modèle de référence | 36 |
| 5.2 | Commande adaptative auto ajustable..... | 35 |
| 5.3 | Concept de la commande adaptative | 36 |
| 5.4 | Commande adaptative directe et commande adaptative indirecte | 36 |
| 5.5 | Commande adaptative par les approximateurs universels | 37 |
| 5.6 | Divergence des paramètres | 38 |
| 6 | Conclusion | 39 |

Chapitre 3 : Commande adaptative d'un système non linéaire sous actionné « Quadrotor »

| | | |
|-----|--|----|
| 1 | Introduction..... | 40 |
| 2 | Principe de fonctionnement et dynamique du quadrotor | 42 |
| 2.1 | Modélisation du quadrotor | 43 |
| 3 | Position de problème..... | 46 |
| 4 | Synthèse d'une commande adaptative pour un système sous-actionné : quadrotor | 48 |
| 4.1 | Commande adaptative indirecte | 50 |
| 4.2 | Contraintes holonomes..... | 51 |
| 4.3 | Analyse de la stabilité | 52 |
| 5 | Résultats de simulation | 53 |
| 6 | Conclusion | 57 |

Chapitre 4 : Commande adaptative par les réseaux de neurones artificiels utilisant un observateur adaptatif neuronal

| | | |
|-----|---|----|
| 1 | Introduction..... | 58 |
| 2 | Position de problème..... | 60 |
| 3 | Synthèse d'une commande adaptative neuronale avec un observateur adaptatif neuronal | 62 |
| 3.1 | Commande adaptative neuronale | 63 |
| 3.2 | Observateur à mode glissant | 65 |
| 3.3 | Conception d'un observateur neuronal adaptatif | 67 |
| 3.4 | Analyse de la stabilité | 69 |
| 4 | Résultats de simulation | 75 |
| 5 | Conclusion | 82 |

Chapitre 5 : Commande adaptative neuronale avec observateur à grand gain adaptative et filtre de Kalman modifié en cascade

| | | |
|-------|--|-----|
| 1 | Introduction..... | 84 |
| 2 | Position de problème..... | 85 |
| 3 | Synthèse d'une commande adaptative neuronale avec un observateur à grand gain adaptative et un filtre de Kalman modifié en cascade | 87 |
| 3.1 | Commande adaptative neuronale | 88 |
| 3.2 | Filtre de kalman | 89 |
| 3.3 | observateur à grand gain | 91 |
| 3.4 | Conception d'un observateur à grand gain adaptatif et filtre de Kalman modifié en cascade..... | 92 |
| 3.4.1 | observateur à grand gain adaptative..... | 92 |
| 3.4.2 | filtre de kalman modifié | 93 |
| 3.5 | Analyse de stabilité | 95 |
| 4 | Résultats de simulation | 102 |

| | |
|----------------------------------|-----|
| 5 Conclusion | 106 |
| Conclusion générale | 108 |
| Bibliographie | 111 |

Liste des figures

Liste des Figures

| | | |
|---------------------|---|----|
| Figure 1.1. | Cellule nerveuse..... | 7 |
| Figure 1.2. | Neurone formel..... | 8 |
| Figure 1.3. | Perceptron à deux couches..... | 10 |
| Figure 1.4. | Problèmes linéairement et non linéairement séparable..... | 11 |
| Figure 1.5. | Perceptron multicouche..... | 11 |
| Figure 1.6. | Mécanisme d'apprentissage supervisé..... | 12 |
| Figure 1.7. | Réseau à fonction de base radiale..... | 15 |
| Figure 1.8. | Identification directe par le RNA..... | 16 |
| Figure 1.9. | Identification inverse par le RNA..... | 17 |
| Figure 1.10. | Signaux utilisés pour l'identification..... | 18 |
| Figure 1.11. | Résultats de l'identification directe..... | 19 |
| Figure 1.12. | Résultats de l'identification de système inverse..... | 20 |
| Figure 2.1. | Point d'équilibre stable..... | 24 |
| Figure 2.2. | Point d'équilibre instable..... | 24 |
| Figure 2.3. | Stabilité asymptotique..... | 25 |
| Figure 2.4. | Point d'équilibre exponentiellement stable..... | 25 |
| Figure 2.5. | Point d'équilibre uniformément stable..... | 26 |
| Figure 2.6. | Stabilité locale et stabilité globale..... | 27 |
| Figure 2.7. | Mode de convergence..... | 32 |
| Figure 2.8. | Mode de glissement..... | 33 |
| Figure 2.9. | phénomène de réticence..... | 34 |
| Figure 2.10. | Commande adaptative à modèle de référence..... | 35 |
| Figure 2.11. | Commande adaptative auto ajustable..... | 36 |
| Figure 3.1. | Configuration d'un quadrotor conventionnel..... | 42 |
| Figure 3.2. | Mouvement du quadrotor..... | 43 |
| Figure 3.3. | Angles d'Euler..... | 44 |
| Figure 3.4. | Quadrotor (système sous-actionné)..... | 46 |
| Figure 3.5. | Schéma de commande adaptative pour un système non linéaire sous-actionné (Quadrotor)..... | 52 |
| Figure 3.6. | Position..... | 54 |
| Figure 3.7. | Vitesses..... | 55 |
| Figure 3.8. | Signal de commande..... | 56 |
| Figure 4.1. | Réseau de neurones artificiel à une seule couche cachée..... | 64 |
| Figure 4.2. | RNA utilisé dans l'observateur proposé..... | 67 |
| Figure 4.3. | Configuration contrôleur/ observateur pour un système SISO de deuxième ordre..... | 69 |
| Figure 4.4. | Quadrotor avec les commandes virtuelles..... | 76 |
| Figure 4.5. | Commande adaptative par les réseaux de neurones artificiels basée sur un observateur adaptatif neuronal d'un quadrotor..... | 77 |
| Figure 4.6. | Position..... | 78 |
| Figure 4.7. | Orientation..... | 79 |
| Figure 4.8. | Vitesse..... | 80 |
| Figure 4.9. | Signal de commande..... | 81 |
| Figure 5-1. | Filtre de kalman..... | 90 |

| | | |
|--------------------|---|-----|
| Figure 5.2. | Observateur à grand gain adaptatif et filtre de kalman modifié en cascade | 94 |
| Figure 5.3. | Pendule inversé | 102 |
| Figure 5.4. | Commande adaptative neuronale avec observateur à grand gain adaptative et filtre de Kalman modifié en cascade (système de 2 ^{ième} ordre)..... | 103 |
| Figure 5.5. | Sortie du système | 104 |
| Figure 5.6. | Vitesse angulaire | 105 |
| Figure 5.7. | Signal de commande | 106 |

Liste des tableaux

Liste des Tableaux

| | |
|--|-----|
| Tableau 1.1. Fonctions d'activation..... | 9 |
| Tableau 3.1. Paramètres du quadrotor..... | 53 |
| Tableau 3.2. Paramètres Contrôleur..... | 54 |
| Tableau 4.1. Paramètres Contrôleur/ Observateur du quadrotor..... | 77 |
| Tableau 5.1. Paramètres de la commande adaptative | 103 |
| Tableau 5.2. Paramètres de OGGA et FKM en cascade | 103 |

Introduction générale

Introduction Générale

Les systèmes physiques sont, non linéaires, interconnectés et soumis à des perturbations internes et externes. La synthèse d'une loi de commande par l'automaticien, nécessite souvent le passage par la modélisation du processus à commander. Cependant, la détermination des équations mathématiques représentant le système, utilise des hypothèses idéales simplifiant le modèle dynamique obtenu qui ne décrit pas fidèlement le système. Ceci limite l'utilisation des techniques de commandes conventionnelles qui sont basées sur le modèle dynamique. Citons par exemples les commandes : PID [Bou04], linéarisante [Lee09] et backstepping [Adi07]. La commande par mode glissant [Buh86, Ond07] est caractérisée par sa robustesse vis-à-vis les perturbations et les variations paramétriques malgré sa dépendance de modèle dynamique. Néanmoins, cette robustesse est due principalement au choix approprié de l'amplitude de signal discontinu.

La commande adaptative est définie comme une technique à paramètre variables, dont, des mécanismes d'ajustement automatiques sont utilisés pour que le système puisse s'adapter aux changements de l'environnement et compenser les erreurs de modélisation. Dans [Ioa84, Mor06, Slo91, Tao03, Yas06] des schémas de commande adaptative conventionnelle ont été développés. Le point commun entre ces travaux est que la commande adaptative est applicable que aux les systèmes linéaire où bien aux les systèmes linéairement paramétrisés. Alors, l'utilisation d'un approximateur non-linéaire est nécessaire pour que la commande adaptative puisse être utilisée pour les systèmes non linéaires complexes. Les approximateurs universels tels que : réseaux de neurones artificiels [Far09], systèmes flous [Zem09], réseaux d'ondelettes [Hus09],..., offrent une solution potentielle à ce problème inhérent à la commande adaptative conventionnelle.

Le réseau de neurones artificiels, est le résultat de l'analogie avec le cerveau humain. Le besoin de produire des machines intelligentes qui peuvent réagir aux variations de l'environnement était la raison de création des RNA. En utilisant les informations entrée-sortie par l'algorithme d'apprentissage rétro propagation (Backpropagation), le RNA est considéré comme un identificateur puissant des systèmes. De plus, selon le théorème d'approximation universel, cet approximateur est capable d'approximer n'importe quelle fonction continue, inconnue. Aujourd'hui on peut trouver plusieurs schémas de commande adaptative qui ont exploités cette propriété pour estimer soit les dynamiques inconnues (commande adaptative indirecte), soit la loi de commande (commande adaptative directe) et cela en utilisant plusieurs

types des RNA tels que les réseaux multicouche [Nar00, Rai08], les réseaux à fonction de base radiale [Mac05, Ond01], les réseaux récurrent [Her09, Poq97]. Dans [Pir02], un RNA multicouche a été utilisé pour estimer les paramètres de PID. Un autre RNA a été employé pour approximer la commande PID dans [Shu05], dont, les fonctions d'activation étaient les actions de PID. Même la commande robuste par mode glissant a profitée des propriétés de RNA, dans [Liu05, Tap01], l'approximateur a été employé pour estimer en ligne la commande équivalente inconnue. On trouve même l'utilisation de RNA pour l'approximation de la commande glissante comme dans [Hua08]. Un RNA a été employé pour estimer les dynamiques inconnues de la loi de commande synthétisée par la technique de backstepping dans [Lin06, Zha00], la même chose dans [Nec07] où la linéarisation entrée-sortie est choisie pour la stabilisation de processus.

La majorité des schémas des contrôleurs adaptatifs sont constitués de trois termes : le premier est un terme adaptatif conçu pour l'approximation des non linéarités inconnues, le deuxième est un terme robuste de type mode glissant [Cha01, Wan01], ou de type H_∞ [Bou00, Mok06] employé pour la compensation des erreurs d'approximation et le troisième terme est ajouté pour garantir la stabilisation de système. Cependant, la commande adaptative est limitée par le problème de l'explosion des paramètres estimés, si le système est soumis à des incertitudes non paramétriques durant des longues durées [Slo91]. Les algorithmes de projection [Goa02, Ioa06], la zone-morte [Far06, Ioa06], σ -switching [Ioa84, Ioa06], σ -modification [Die08, Die09], e-modification [Das09, Rai08] sont les techniques souvent utilisées pour éviter ce problème. Dans [Bou10, Bou12, Coz06], deux approximateurs de même type pour estimer les mêmes incertitudes. La singularité de la loi de commande est un autre problème rencontré par la commande adaptative indirecte si le gain de commande estimé prend la valeur zéro. Dans [Xu01a, Xu01b, Kos99], la structure de la loi de commande a été changée pour éviter ce problème même si le gain de commande estimé prend la valeur zéro. Dans [Cha01], le mécanisme d'adaptation est basé sur les algorithmes de projection et dans [Tin07], l'adaptation est conçue pour estimer l'inverse de gain de commande.

La mise en œuvre de toutes les lois de commandes requiert la connaissance du vecteur d'état complet à chaque instant. Cependant, plusieurs raisons pratiques et techniques empêchent la disponibilité de toutes les informations nécessaires. De ce fait, l'utilisation d'un observateur s'avère nécessaire pour reconstruire les variables d'états non mesurables ou difficiles à mesurer. L'implémentation d'un observateur pour les systèmes linéaires a été l'objet de plusieurs travaux depuis des années [Che95] d'où, l'estimation des états est faite par de l'observateur de Luenberger [Lue71] ou par le filtre de Kalman [Kar10] si le système est sous l'effet des bruits.

Le succès connu dans ce domaine est dû à la vérification facile de principe de séparation. Cependant, le design d'un observateur pour les systèmes non linéaire et la démonstration de ce principe présentent un grand challenge. Parmi les observateurs qu'on trouve; l'observateur à grand gain et/ou à mode glissant [Ala09, Roa10]. Cependant, ces observateurs présentent deux phénomènes indésirables à savoir : le phénomène de pic et l'amplification des bruits de mesure. Pour remédier ces problèmes, plusieurs travaux de recherches ont été effectués entre autres nous citons [Bul97, Far09, Kha08]. On trouve d'autres types d'observateurs comme dans [Kim04], où un observateur avec une structure en cascade est développé. L'observateur dans [Lee07] est basé sur l'observateur de Luenberger et un terme additionnel discontinu. Dans [Bod11], le terme de correction de l'observateur est basé sur la technique PI. Dans [Pra08], un observateur non linéaire basé sur le modèle prédictif de contrôleur est utilisé par l'intermédiaire des systèmes flous et de filtre de kalman. [Kal11], combine entre l'observateur à grand gain et à mode glissant pour l'estimation des états. Dans [Oue12] un observateur à grand gain adaptatif est utilisé.

Dans cette thèse, des schémas de commande adaptative avec et sans observateur sont développés pour traiter quelques types de problèmes. Le travail effectué est divisée en cinq chapitres comme suit :

Un bref historique sur l'origine biologique de Réseaux de Neurones Artificiels (RNA) et les étapes les plus importantes dans son développement sont présentées dans le premier chapitre. L'apprentissage par l'algorithme rétro-propagation est exposé et des applications en identification directe et inverse sont données. Le théorème d'approximation universel est donné afin de montrer les capacités des RNA.

Dans le deuxième chapitre, les définitions théoriques de la stabilité au sens de Lyapunov sont données. De plus, les deux méthodes de Lyapunov (directe et linéarisante) utilisées pour étudier la stabilité des systèmes non linéaires sont présentées afin de les utilisées dans les schémas de commande proposées. Par la suite, trois types de commandes stabilisantes sont exposés, avec leurs avantages et leurs inconvénients, ces techniques à savoir : la technique de commande par backstepping, la commande par mode de glissement et la commande adaptative. Une recherche bibliographique est effectuée sur les solutions proposées dans la littérature pour résoudre les limitations de chacune. De plus, pour montrer; pourquoi nous avons opté pour la commande adaptative intelligente dans ce travail de thèse.

Le troisième chapitre présente le développement d'un schéma de commande adaptative pour le système sous actionnée « quadrotor ». Une description générale de modèle d'application,

son principe de fonctionnement et son modèle dynamique sont présentés au début. Des commandes virtuelles et des contraintes holonomes sont proposées pour traiter le problème de manque de deux commandes dans le quadrotor. Les termes inconnus sont estimés en ligne par un mécanisme d'adaptation. La validation de schéma proposé est montrée par simulation à la fin de chapitre.

Le quatrième chapitre comporte la proposition d'un schéma de commande adaptative neuronale basé sur un observateur d'état pour une classe des systèmes non linéaires multi variable. La capacité d'approximation des RNA est utilisée pour l'estimation de la commande équivalente. La reconstruction de vecteur d'état est faite par un observateur neuronal adaptatif, d'où le but est de minimiser la sensibilité aux bruits causés par les capteurs. Les performances de schéma proposé sont montrées par simulation numérique sur le modèle dynamique d'un quadrotor où de nouvelles commandes virtuelles et contraintes holonomes sont utilisées pour rendre le système de la même classe étudiée.

La synthèse de la commande adaptative par RNA dans le cinquième chapitre est semblable à celle du chapitre quatre mais pour un système non linéaires mono variable. Bien que, l'estimation de toutes les variables d'états est obtenue par l'implémentation de deux observateurs en cascade à savoir : un observateur à grand gain adaptatif (OGGA) et un filtre de kalman modifié (FKM). Où la sortie du premier observateur est utilisée comme entrée du deuxième observateur. L'idée de base est de combiner les performances des deux observateurs pour générer des signaux débruités et garantir les performances en boucle fermée. Le pendule inversé est choisi comme l'exemple d'application pour valider la technique proposée.

L'analyse de la stabilité, la bornitude de tous les signaux et la robustesse vis-à-vis les erreurs d'approximation de chaque schéma de commande proposé sont obtenues par l'utilisation de la méthode directe de Lyapunov.

Nous terminons par une conclusion générale sur les travaux présentés et développés dans cette thèse et les perspectives envisageables sur les travaux futurs.

Chapitre I

Réseaux de Neurones Artificiels « RNA »

1. Introduction

Dans ce chapitre, nous présentons l'approximateur universel choisi dans ce travail de thèse, c'est le Réseau de Neurones Artificiels (RNA). Créer des machines à la même intelligence que l'homme était l'idée derrière l'apparition des RNAs. Alors, l'objectif des recherches était de créer un modèle approximatif au cerveau. Le neurone formel était le premier résultat de ces recherches. Le neurone formel, est la représentation mathématique de la cellule nerveuse, et la règle d'apprentissage utilisée était basée sur la nature physiologiste des neurones biologiques. Ce modèle était capable de réaliser quelques opérations arithmétiques et logiques [Zil01]. Le succès du neurone formel a permis le développement du premier Réseau de Neurones Artificiels (RNA), c'est le perceptron. Cependant, ce type a connu des limitations si le problème est non linéairement séparable. Le RNA multi couches entraîné par l'algorithme d'apprentissage rétro-propagation (Backpropagation) du gradient [Gau99] a donné une solution au problème rencontré. L'algorithme rétro-propagation est considéré comme une règle d'apprentissage universel pour d'autres approximateurs tels que les systèmes flous [Zem09]. L'efficacité des RNAs multi couches dans l'approximation des termes complexes était la raison d'élargir le domaine d'utilisation de cet approximateur, à savoir : traitement d'image, reconnaissance des formes [Zil01], l'identification des processus [Tli06], traitement de signal [Lot08]...etc.

En automatique, les RNAs sont souvent implémentés dans des schémas de commande, où la propriété d'approximation universelle [Spo02], est exploitée pour la mise en œuvre de la loi de commande synthétisée en approximant les parties inconnues. Cependant, les techniques de commande classique peuvent utiliser les RNAs afin d'améliorer des performances du système bouclé. Citons entre autres : la commande par mode glissant [Slo91], où les RNAs sont souvent

utilisés pour approximer la commande équivalente [Bou12, Zha05] et/ou bien la commande de correction [Hua08]. La technique Backstepping [Kha96] a profité des propriétés des RNAs aussi, comme dans [Das09, Die08] ; les fonctions non linéaires inconnues dans le modèle dynamique ont été estimées par cet approximateur. Le même principe est utilisé dans [Nec07] avec la technique de linéarisation entrée-sortie [Slo91]. Egalement, on trouve dans la littérature quelques utilisations de cet approximateur dans les systèmes d'observation comme dans les travaux : [Bou11b, Bou14, Die08] et ce travail de thèse.

Dans ce chapitre, nous présentons quelques notions de base sur les RNAs ; la structure du neurone formel et l'analogie avec la cellule nerveuse, les étapes les plus importantes du développement des RNAs, l'algorithme d'apprentissage rétro-propagation et son utilisation en identification, et le théorème d'approximation universel.

Ce chapitre est organisé comme suit : dans la section 2, les origines biologiques des réseaux de neurones artificiels sont présentées. Un bref historique sur le développement des réseaux de neurones artificiels (les topologies multicouches et l'algorithme rétro-propagation) est exposé dans la section 3. Le théorème d'approximation universel par les réseaux de neurones artificiels est présenté dans section 4. En se basant sur ce dernier des schémas d'identification directe et inverse des systèmes par les réseaux de neurones artificiels sont présentées dans la section 5. La conclusion est donnée dans la dernière section.

2. Origine biologique des réseaux de neurones artificiels

Depuis plusieurs décades, l'objectif des chercheurs était le développement des machines avec la même intelligence que l'être humain. Pour cela, le fonctionnement de cerveau humain est modélisé ce qui a donné naissance aux Réseaux de Neurones Artificiels (RNAs).

Un réseau de neurones artificiel, est composé de plusieurs éléments simples connectés entre eux par des liaisons pondérées (poids). Plusieurs topologies ont été développées à savoir : multicouche, à connexion locale, à connexions récurrentes, connexion complète..., dont, la forme de connections entre ses éléments définit son fonctionnement. De la même manière comme le cerveau, un réseau de neurone artificiel peut effectuer plusieurs tâches (reconnaissance de forme, modélisation et commande des processus,...etc.) par l'ajustement des pondérations. Autrement dit, la reconnaissance des tâches est enregistré par le changement dans les pondérations, cette phase est appelée apprentissage.

2.1. Neurone biologique

Le cerveau humain est composé d'un réseau de neurones très compliqué, dont, il contient plus que 10^{10} [Zil01] cellules nerveuse connectées entre eux. La cellule nerveuse ou le neurone biologique est composée d'un corps cellulaire (ou bien soma), d'un axone et de plusieurs dendrites [SAR06] comme il est présenté par la Figure 1.1.

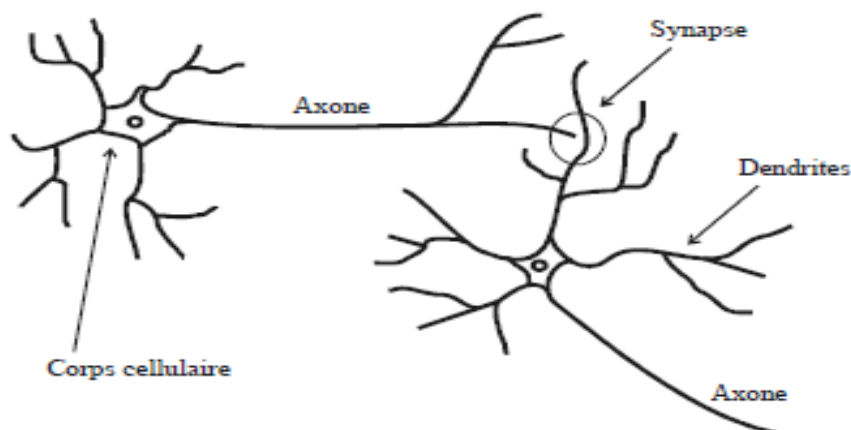


Figure 1.1. Cellule nerveuse.

Le principe de transmission des signaux entre les cellules nerveuse est très simple. Les **dendrites** transmettent les signaux de l'extérieur vers le **corps cellulaire** de neurone. Ce dernier est responsable sur la sommation et le seuillage des signaux. C'est-à-dire, si la somme résultante dépasse certain seuil, le neurone envoie un signal par son **axone** vers un autre neurone. Dans le cas contraire, la somme sera détruite rapidement et aucune action ne sera prête [Sar06].

L'espace intercellulaire entre l'axone et les dendrites s'appelle **synapse** [Sar06, Zil01]. Elle est de nature chimique ayant un effet électrique [Sar06] qu'on peut le mesurer. L'importance de ces synapses est dans la capacité de changement lorsque le cerveau est en phase d'apprentissage.

En se basant sur les notions : dendrites, corps cellulaire, axone, synapse, et le principe de transmission des signaux, les chercheurs ont créés un modèle approximatif à la cellule nerveuse, c'est le neurone formel. rate

2.2. Neurone formel

Le premier résultat était en 1943 par McCulloch et Pitts [Zil01], dont, l'apparition de neurone formel. Ce dernier, est un processus élémentaire (automate) à n entrées et une seule

sortie. Chaque entrée est associée à un poids synaptiques V qui représente la force de la connexion (Voir Figure 1.2).

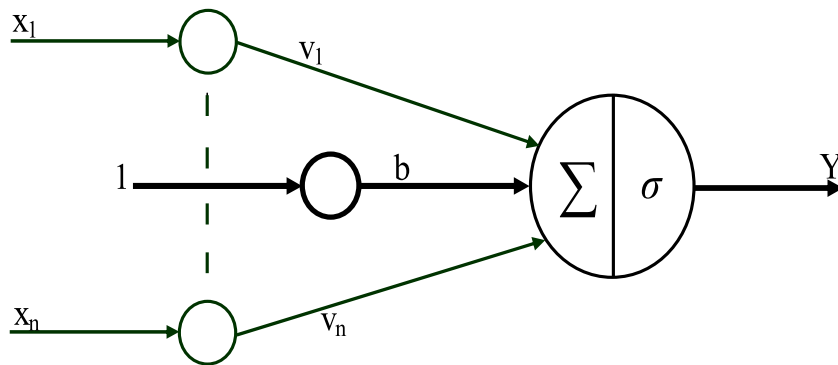


Figure 1.2. Neurone formel.

Le neurone formel est composé de trois parties comme suit :

2.1.1. Poids

Les coefficients v_i sont les poids associés à chaque entrée x_i . Dont, chaque entrée est multiplié par le poids qui lui correspond est le résultat sera $(V^T X)$ avec : $V = [v_1 \ \dots \ v_n]^T$ et $X = [x_1 \ \dots \ x_n]^T$.

2.1.2. Seuil

Le biais b présente le seuil de neurone formel, ce paramètre additionnel est considéré comme le poids de l'entrée unité.

2.1.3. Fonction d'activation

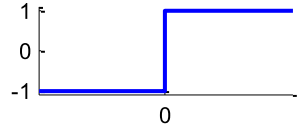
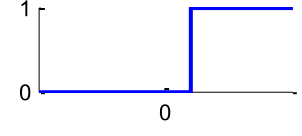
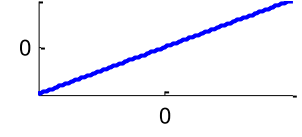
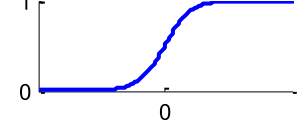
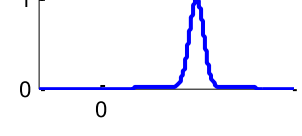
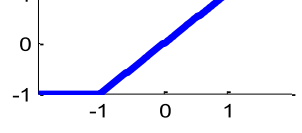
La fonction d'activation calcul l'état du neurone, cette valeur sera transmise aux neurones avals. Il existe de nombreuses formes possibles de la fonction d'activation, les plus courantes sont présentées dans le tableau 1.1 [Zil01, Spo02, Sar06].

Comme conclusion, le résultat de cet automate s'exprime par un simple produit scalaire entre le vecteur d'entrées (X) et le vecteur des poids (V), ensuite le seuillage par un biais b et la fonction d'activation σ sont appliqués au résultat obtenu et la sortie finale du neurone sera :

$$Y = \sigma(V^T X + b) \quad (1.1)$$

Il est à noter que le neurone biologique a une sortie de nature binaire, c'est-à-dire le neurone peut prendre deux états seulement (active ou désactivé), par contre l'utilisation des fonctions d'activation continues ci-dessous permet d'avoir une infinité de valeurs possibles pour la sortie de neurone formel.

Tableau 1.1. Fonctions d'activations

| Fonction d'activation | Modèle mathématique | Graphe |
|-----------------------|--|---|
| Fonction signe | $\sigma(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{si } x \leq 0 \end{cases}$ |  |
| Fonction seuil | $\sigma(x) = \begin{cases} 1 & \text{si } x > s \\ 0 & \text{si } x \leq s \end{cases}$ |  |
| Fonction linéaire | $\sigma(x) = x$ |  |
| Fonction sigmoïde | $\sigma(x) = \frac{1}{1 + \exp(-x)}$ |  |
| Fonction gaussienne | $\sigma(x) = \frac{1}{\sqrt{2\pi\sigma'^2}} e^{-\frac{(x-c)^2}{2\sigma'^2}}$ |  |
| Fonction saturation | $\sigma(x) = \begin{cases} +1 & \text{si } x \geq +1 \\ x & \text{si } -1 \leq x \leq +1 \\ -1 & \text{si } x \leq -1 \end{cases}$ |  |

3. Développement des réseaux de neurones artificiels et les règles d'apprentissage

Par définition, un réseau de neurones artificiels est un ensemble de neurones formels connectés entre eux. Le développement des RNAs se résume comme suit :

- En 1949, le physiologiste américain D. O. Hebb dans son ouvrage *'The Organization of Behavior'* [Rum86] a introduit la première règle d'apprentissage des coefficients synaptiques. Cette règle s'exprime comme suit : *'Si deux cellules sont activées au même temps alors la force de la connexion augmente, sinon elle reste constante'*.

Cette règle a connu beaucoup de succès, dont, plusieurs fonctions logiques et arithmétiques ont été réalisées par le neurone formel en utilisant cette règle.

- En 1958, F. Rosenblatt propose le premier réseau de neurone artificiel pratique pour la reconnaissance des formes, c'est le Perceptron. Ce modèle est composé de deux couches, la couche d'entrée et la couche de sortie comme il est présenté par la Figure 1.3 :

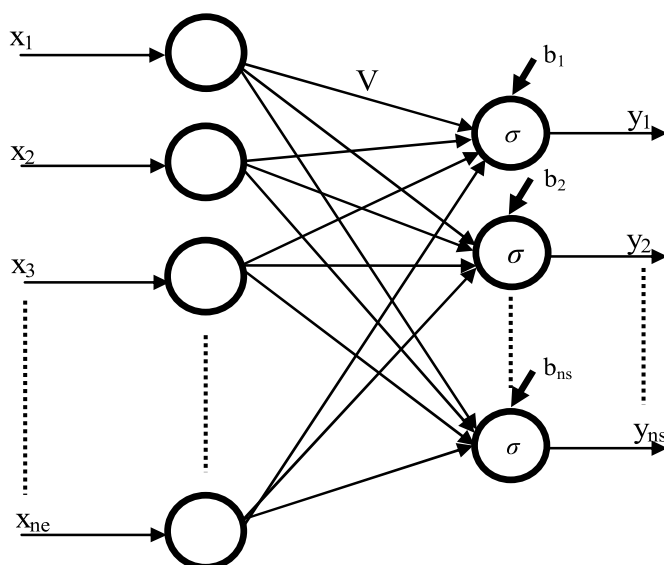


Figure 1.3. Perceptron à deux couches.

Avec : σ est la fonction seuil (modèle de F. Rosenblatt), $X = [x_1 \ \dots \ x_{ne}]^T \in \mathfrak{R}^{ne}$ est le vecteur d'entrée, $Y = [y_1 \ \dots \ y_{ns}]^T \in \mathfrak{R}^{ns}$ est le vecteur de sortie, $V \in \mathfrak{R}^{ns \times ne}$ est la matrice des poids et ne et ns sont le nombre des neurones dans la couche d'entrée et de sortie respectivement.

La loi d'apprentissage de perceptron est inspirée de la règle de D. O. Hebb : 'En cas d'erreur en sortie le vecteur d'entrée correspondant est ajouté ou retranché au vecteur des poids *synaptiques*'.

- En 1960, l'automaticien B. Widrow développe l'Adaline (**Ad**aptative **L**inear **E**lement), Ce modèle à la même structure de Perceptron mais la fonction d'activation est linéaire et la règle d'apprentissage est supervisée dite 'règle Delta' ou 'règle de Widrow' [Rum86, Zil01]. L'ajustement des poids ici est basé sur la minimisation de l'erreur quadratique moyenne entre la sortie de réseau et la sortie désirée. L'Adaline est connu comme le modèle de base des RNAs actuels (réseaux de neurones artificiel multicouche).

- En 1969, M. Minsky et S. Papert montrent les limites des modèles perceptrons et l'Adaline pour le cas des problèmes non linéairement séparables (Voir Figure 1.4) [Zil01] tel que : la fonction logique 'XOR'.

Après cette publication la notion des réseaux de neurones artificiels est disparue pendant près de vingt ans.

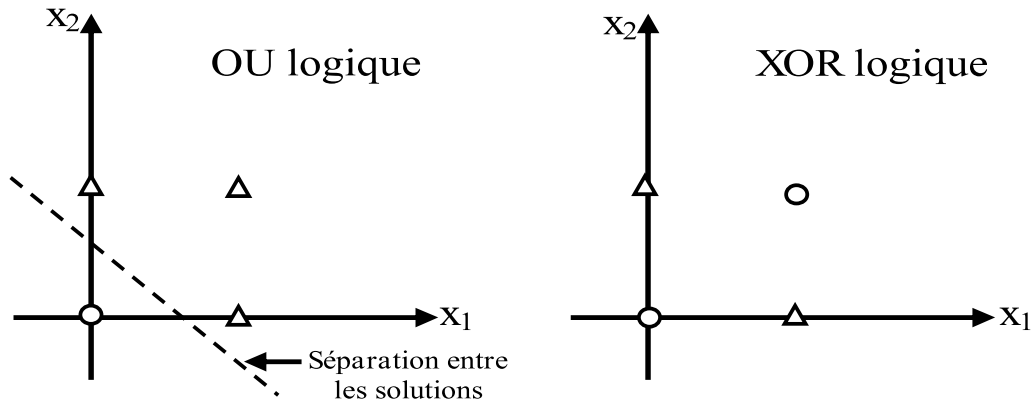


Figure 1.4. Problèmes linéairement et non linéairement séparable.

3.1. Perceptron multicouche

En 1986 D. Rumelhart montre que le problème non linéairement séparable est surmonté par l'utilisation d'un perceptron multicouche [Rum86]. Ce dernier est composé d'une couche d'entrée, une couche de sortie et une/ ou plusieurs couches cachées, comme il est illustré sur la Figure 1.5. Ce réseaux est capable de traiter tous les problèmes (linéairement séparable ou non). La fonction logique 'XOR' par exemple peut être réalisée par un réseau à trois couches : une couche d'entrée à deux neurones, une couche de sortie à un seul neurone et une couche cachée à deux neurones [Rum86].

Le perceptron multicouche est un réseau à propagation avant (feed-forward), dont, l'information se propage dans un sens unique, de la couche d'entrée vers la couche de sortie. La fonction d'activation σ est située dans les couches cachées.

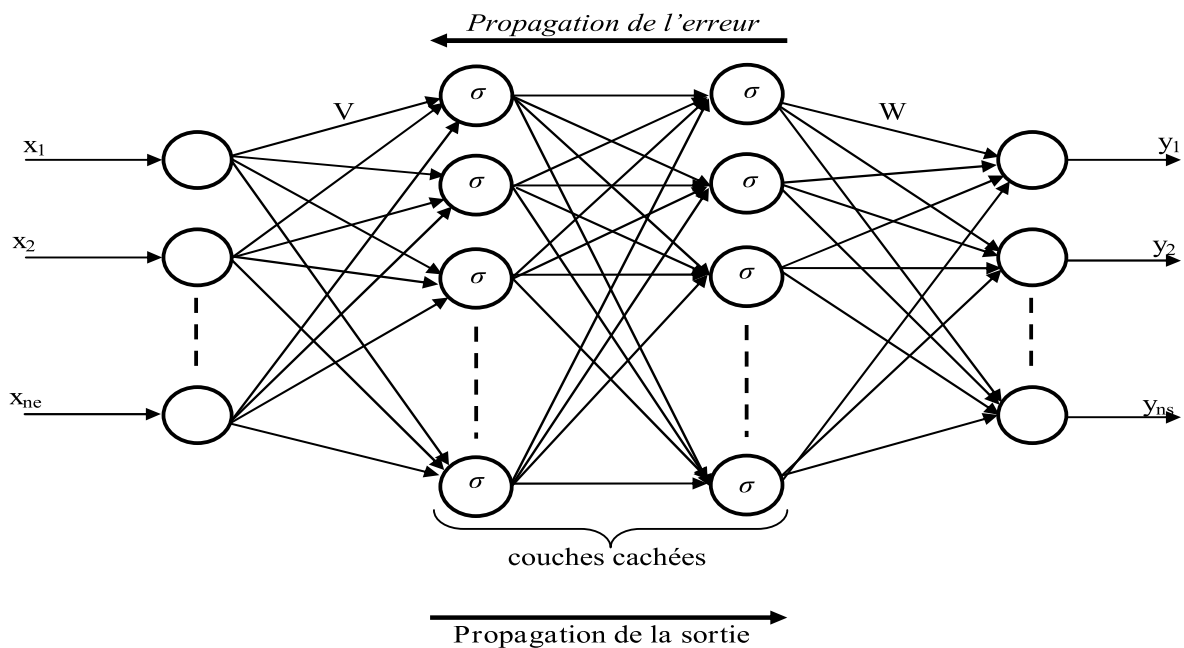


Figure 1.5. Perceptron multicouche.

L'apprentissage de ce RNA se fait par l'algorithme de rétro-propagation (backpropagation) [Zil01, Sar06]. Le principe de cet algorithme est donné dans la partie suivante.

3.1.1. Algorithme de rétro-propagation

L'algorithme de rétro-propagation est basé sur la méthode de gradient descendant [Rum86, Zil01], dont, l'apprentissage est supervisé. L'ajustement des poids se fait dans le sens inverse de la couche de sortie vers la couche d'entrée d'où l'appellation rétro propagation (backpropagation). Au début, la sortie du réseau est calculée et comparée avec la sortie désirée, ensuite l'erreur résultante est utilisée pour l'apprentissage des poids de la couche de sortie et de la dernière couche cachée. De la même manière, l'erreur du réseau se propage pour l'ajustement des poids des autres couches cachées jusqu'à la couche d'entrée.

L'opération d'apprentissage supervisé d'un réseau de neurones artificiels est résumée par la figure suivante :

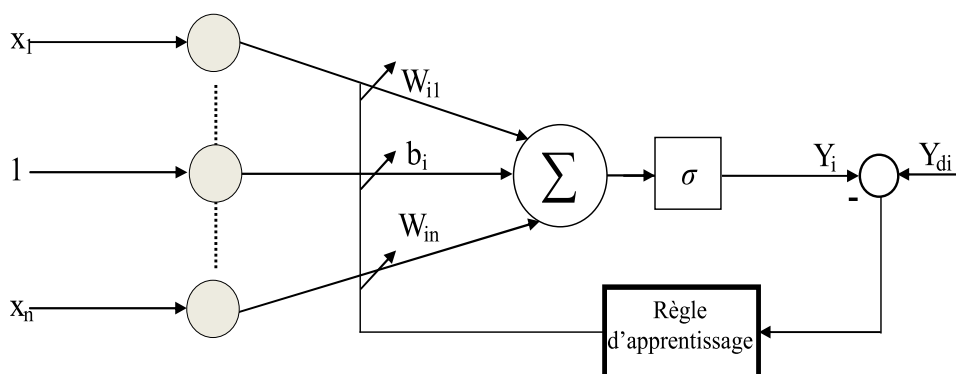


Figure 1.6. Mécanisme d'apprentissage supervisé.

Selon la Figure 1.6, l'ajustement des poids minimise l'erreur entre les sorties désirée et réelle du RNA.

3.1.1.a. Algorithme du gradient

L'algorithme de gradient est basé sur la minimisation une fonction coût, pour un apprentissage supervisé des réseaux de neurones artificiels cette fonction est donnée par:

$$J_i = \frac{1}{2} E_i^2 = \frac{1}{2} (y_{di} - y_i)^2 \quad (1.2)$$

Avec : E_i : est l'erreur du neurone de sortie i , y_{di} : est la sortie désirée de neurones de sortie i et y_i : est la sortie réelle de neurone de sortie i .

L'algorithme du gradient pour l'ajustement des poids et de biais est donnée par :

$$\begin{cases} w_{ij}(t+1) = w_{ij}(t) - \mu \frac{\partial J_i}{\partial w_{ij}(t)} \\ b_i(t+1) = b_i(t) - \mu \frac{\partial J_i}{\partial b_i(t)} \end{cases} \quad (1.3)$$

La constante $0 < \mu < 1$ représente le gain d'apprentissage.

3.1.1.b Etapes de rétro-propagation

La sortie d'un réseau de neurones artificiel multicouche (Figure 1.5) se calcul de la couche d'entrée vers la couche de sortie. Par contre, la minimisation de l'erreur entre la sortie réelle et la sortie désirée du RNA par la méthode de gradient est obtenue par la propagation du signal d'erreur dans le sens inverse c'est-à-dire de la couche de sortie vers la couche d'entrée. Alors, l'ajustement des poids se fait dans le même sens que le signal d'erreur et l'algorithme de gradient qui réalise cette tâche est appelé la rétro-propagation.

Les différentes étapes à suivre lors de l'apprentissage d'un réseau de neurones à propagation avant avec cet algorithme sont comme suit :

- **Étape 1.** Initialiser de tous les poids par des valeurs aléatoires.
- **Étape 2.** Utiliser des flèches entrées-sorties pour l'apprentissage.
- **Étape 3.** Calculer le signal de sortie du RNA comme suit :

- La sortie de chaque neurone j de la couche cachée k est donnée par :

$$O_j^k = \sigma \left(\sum_{i=1}^{nc_{k-1}} w_{ij}^{k-1} O_i^{k-1} \right), \quad \begin{cases} k = \overline{1:p}, \\ j = \overline{1:nc_k} \end{cases} \quad (1.4)$$

Avec : σ est la fonction d'activation, w_{ij}^{k-1} sont les poids liant la couche $k-1$ et la couche k , nc_{k-1} est le nombre de neurones dans la couche cachée $k-1$, p est le nombre de couches cachées. Pour la première couche cachée ($k=1$), on note nc_0 par ne c'est-à-dire le nombre de neurones dans la couche d'entrée et $O_i^0 = x_i$ désigne l'entrée du RNA.

- La sortie de chaque neurone de sortie l , est donnée par :

$$y_l = \sum_{i=1}^{nc_p} w_{ij}^p O_i^p, \quad \begin{cases} l = \overline{1:ns} \\ j = \overline{1:l} \end{cases} \quad (1.5)$$

Avec ns est le nombre de neurones dans la couche de sortie.

• **Étape 4.** Calculer les erreurs des neurones de la couche de sortie et par la suite toutes les erreurs des neurones des couches cachées comme suit :

- L'erreur de chaque neurone de sortie l est donnée par :

$$e_l = y_{dl} - \sum_{i=1}^{nc_p} w_{ij}^p O_i^p, \begin{cases} l = \overline{1:ns} \\ j = \overline{1:1} \end{cases} \quad (1.6)$$

Avec y_{dl} est la sortie désirée de neurone de sortie l .

- L'erreur de chaque neurone caché j dans la couche cachée k est calculée par :

$$e_j^k = \sum_{i=1}^{nc_{k+1}} w_{ij}^{k+1} e_i^{k+1}, \begin{cases} k = \overline{p:1}, \\ j = \overline{1:nc_k} \end{cases} \quad (1.7)$$

Avec e_i^{p+1} est l'erreur e_l définie dans (1.6).

• **Étape 5.** en utilisant l'algorithme de gradient, l'ajustement des poids est effectuée de la couche de sortie jusqu'à la couche d'entrée comme suit :

$$w_{ij}^{k-1}(t+1) = w_{ij}^{k-1}(t) + \mu e_j^k \sigma' \left(\sum_{i=1}^{nc_{k-1}} w_{ij}^{k-1}(t) O_i^{k-1} \right) O_i^{k-1}, \begin{cases} k = \overline{p+1:1}, \\ i = \overline{1:nc_{k-1}} \\ j = \overline{1:nc_k} \end{cases} \quad (1.8)$$

Avec :

$$\sigma'(x) = \frac{d \sigma(x)}{d x} \quad (1.9)$$

• **Étape 6.** refaire les étapes de 2 à 5 jusqu'à l'obtention d'une fonction coût proche de zéro (inférieur à la valeur désirée).

3.2. Réseau de neurone à fonctions de base radiale

L'emploi d'une fonction d'activation sigmoïdale a permis une large utilisation des RNA. Cependant, les fonctions sigmoïdes ne sont pas adoptés pour tous les domaines à titre d'exemple le traitement de signaux [Hus09]. Pour cette raison d'autre type de réseaux ont été développés, à savoir entre autres : le réseau à Fonction de Base Radiale (FBR) [San92]. Comme les perceptrons multicouches, les réseaux de neurones à fonctions de base radiale sont des réseaux à propagation avant avec une fonction d'activation de type gaussienne (voir Tableau 1.1). La Figure 1.7 montre, un réseau à fonction de base radiale à deux entrées et quatre neurones dans la couche cachée.

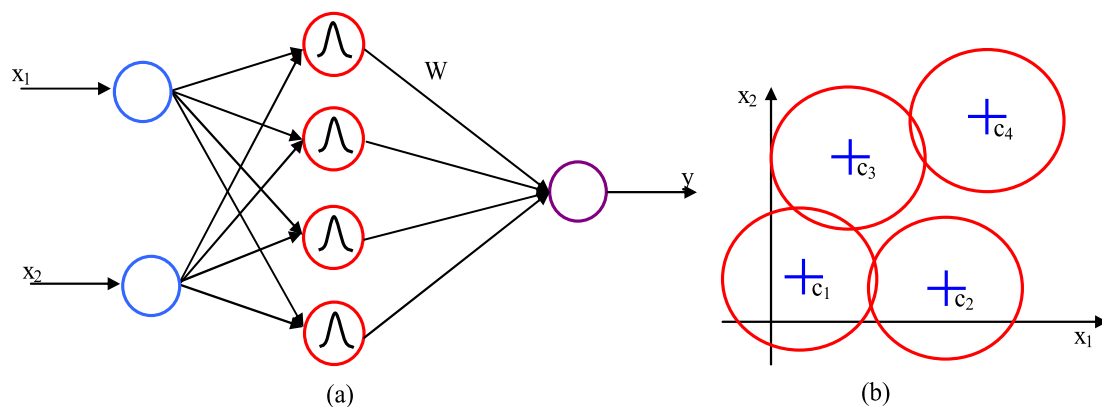


Figure 1.7. Réseau à fonction de base radiale, (a) : structure FBR, (b) : distribution des centres.

Contrairement aux réseaux de neurones multicouches, les neurones cachés d'un réseau de neurones à fonction de base radiale sont activés au voisinage des centres des fonctions gaussiennes seulement de plus, les connexions provenant des neurones d'entrée ne sont pas pondérées.

Il existe deux types de ce réseau, le premier est à centres fixes [Ond01], mais, l'inconvénient majeur de ce type est la nécessité d'utiliser un nombre très important de neurones cachés pour palier tous l'espace d'état. Le deuxième type est les FBRs à centres variables [Bou10]. Avec ce type, on minimise le nombre de neurones des couches cachés. Alors, l'apprentissage des FBRs implique l'ajustement des poids de la couche de sortie et l'ajustement des centres des fonctions gaussiennes supposées variables et ajustées par un algorithme approprié.

4. Approximation neuronale des fonctions

Le fondement théorique de l'approximation des fonctions inconnues par les RNAs est présenté dans cette section.

Le théorème d'approximation universel consiste que toute fonction continue $f(x)$ définie sur un ensemble compact Ω peut être approximée par un réseau de neurone à trois couches avec un ajustement approprié des poids [Sar06], où :

$$\sup_{x \in \Omega} |f(x) - \hat{f}(x)| \leq \varepsilon \quad (1.10)$$

Avec : $\hat{f}(x)$ est l'approximation neuronale de $f(x)$ et ε est une petite constante positive définie l'erreur d'approximation neuronale.

En exploitant les propriétés d'approximation universel, un RNA est généralement implémenté dans un schéma de commande adaptative directe ou indirecte [Ioa06] pour

approximer la commande elle-même ou bien le modèle dynamique du système à commander. L'adaptation des coefficients synoptiques pour un contrôle adaptatif est effectué en ligne en utilisant une de ces deux méthodes : par la minimisation d'une fonction coût comme dans [Hun04] ou bien par l'utilisation des algorithmes issus de l'étude de la stabilité au sens de Lyapunov [Bou11b].

Dans cette thèse, les RNAs sont utilisés dans plusieurs schémas de commande adaptative. Le but est d'approximer en ligne les termes inconnus des systèmes dynamiques à commander. En plus, cet approximateur est utilisé pour pallier les problèmes des observateurs d'états.

5. Identification par les réseaux de neurones artificiels

L'efficacité de l'algorithme de rétro-propagation a permis l'utilisation des RNAs dans plusieurs domaines tels que : l'identification des processus. Dans cette section, nous présentons les schémas d'identification directe et inverse. A la fin de cette section, un exemple d'application est donné.

5.1. Identification directe

Dans un schéma d'identification direct (voir Figure 1.8), le RNA est en parallèle au système à identifier, dont, l'erreur entre la sortie du système (Y) et celle de réseau (Y_N) est utilisée pour l'apprentissage du réseau. Après cette phase, le RNA peut simuler le comportement du système.

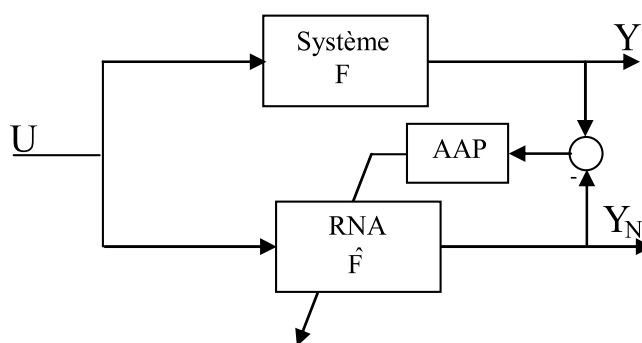


Figure 1.8. Identification directe par le RNA.

5.2. Identification inverse

Dans un schéma d'identification inverse le RNA est en cascade avec le système, dont, l'entrée de RNA est la sortie du système. L'objectif est d'utiliser le modèle obtenu comme

contrôleur. Sa mise en cascade avec le système permet d'avoir une fonction de transfert unitaire entre la trajectoire désirée et la sortie réelle du système.

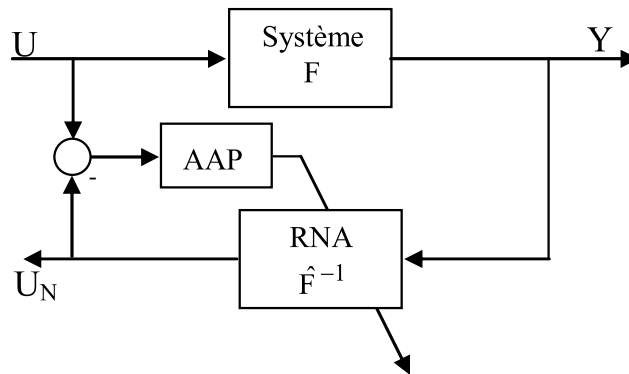


Figure 1.9. Identification inverse par les RNAs.

5.3. Application de l'algorithme rétro-propagation

Dans cette section, l'implémentation des RNAs pour l'identification du modèle directe et inverse d'un système quelconque est présentée. Les réseaux de neurones utilisés sont à propagation avant composé de trois couches : couche d'entrée, couche de sortie et une seule couche cachée.

5.3.1. Identification neuronale directe

L'objectif est d'obtenir le modèle neuronale d'un système dynamique quelconque. La sortie du réseau de neurone utilisé pour l'identification est choisie comme suit :

$$Y_N = W^T \sigma(V^T \chi) \quad (1.11)$$

Avec Y_N est la sortie du RNA, W est la matrice des poids de la couche cachée/ couche sortie, V est la matrice de la couche d'entrée/ couche cachée, la fonction d'activation $\sigma(\cdot)$ est définie par :

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1.12)$$

Et sa dérivée par :

$$\sigma'(x) = \frac{d}{dx} \sigma(x) = \sigma(x)(1 - \sigma(x)) \quad (1.13)$$

En se basant sur le schéma d'identification directe donnée par la Figure 1.8, l'entrée de RNA est choisie $\chi = [Y \ U]^T$ avec Y et U sont les signaux de sortie et entrée utilisés pour l'identification (Voir Figure 1.10).

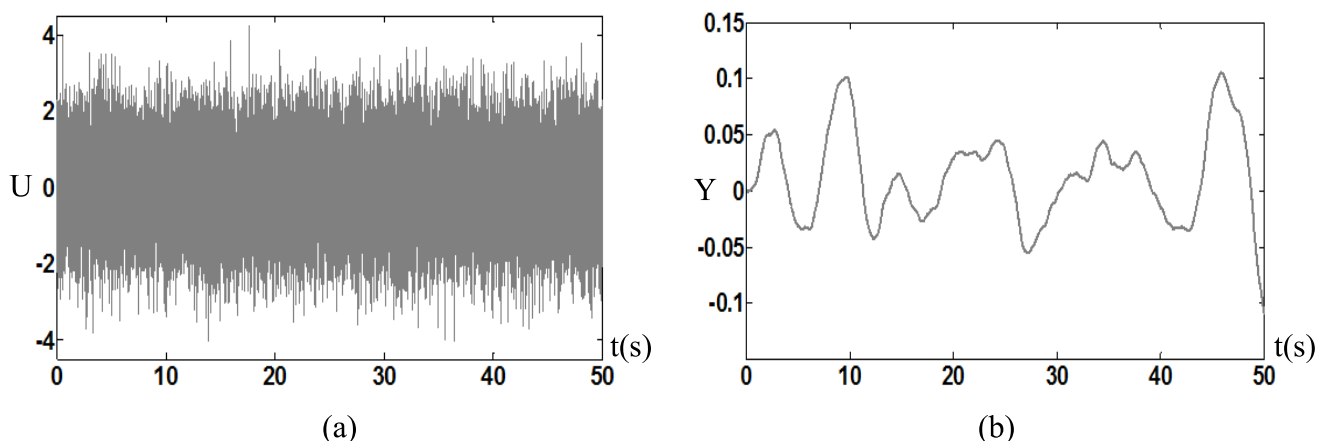


Figure 1.10. Signaux utilisés pour l'identification : (a) : signal d'entrée, (b) : signal de sortie.

L'apprentissage est effectué par l'algorithme de rétro-propagation, dont le RNA utilisé dans l'identification est composé de deux neurones dans la couche d'entrée, 18 neurones dans la couche cachée et un seul neurone dans la couche de sortie, le gain d'apprentissage $\mu=0.1$. L'initialisation aléatoire des poids et les poids finals après 80 itérations identifiant le système sont donnés par (1.14).

Pour valider les résultats d'identification, nous avons appliqué au système original et le RNA obtenu une entrée sinusoïdale, la Figure 1.11 montre les signaux obtenus :

$$\begin{matrix}
 \begin{matrix}
 \left[\begin{matrix} 1 \\ 3.4 \\ 2 \\ 0.1 \\ 1.2 \\ 3 \\ 0.1 \\ 1 \\ 3.2 \\ 1.1 \\ 1 \\ 1 \\ 0.1 \\ 2.1 \\ 6.2 \\ 0 \\ 0.1 \\ 9.1 \\ 0.1 \end{matrix} \right] \\
 W =
 \end{matrix}
 &
 , V^T = \begin{matrix}
 \left[\begin{matrix} 1 & 2 \\ 0.3 & 0.1 \\ 0.4 & 0 \\ 1.1 & 0.1 \\ 0.3 & 2 \\ 0.1 & 2 \\ 1 & 1 \\ 11 & 11 \\ 3.2 & 3.2 \\ 0.1 & 0.1 \\ 1 & 1 \\ 1 & 1 \\ 0.2 & 0.4 \\ 1.3 & 2.5 \\ 3.2 & 1.3 \\ 0 & 0 \\ 3.4 & 2.1 \\ 1.2 & 1.1 \\ 1.5 & 1.3 \end{matrix} \right] \\
 \end{matrix}
 &
 \xrightarrow{\text{après apprentissage}}
 &
 W = \begin{matrix}
 \left[\begin{matrix} -1.0973 \\ 1.2293 \\ -0.1485 \\ -1.9828 \\ -0.8973 \\ 0.9027 \\ -1.9892 \\ -1.0981 \\ 1.1019 \\ -0.9599 \\ -1.0890 \\ -1.0890 \\ -1.9717 \\ 0.0021 \\ 4.1064 \\ -0.2221 \\ -1.9975 \\ 7.0099 \\ 0.1000 \end{matrix} \right] \\
 \end{matrix}
 &
 , V^T = \begin{matrix}
 \left[\begin{matrix} 1.0000 & 2.0021 \\ 0.3000 & 1.4070 \\ 0.4000 & 3.0362 \\ 1.1000 & 0.9239 \\ 0.3000 & 2.0018 \\ 0.1000 & 1.9983 \\ 1.0000 & 1.0449 \\ 11.000 & 11.000 \\ 3.2000 & 3.1999 \\ 0.1001 & 0.3096 \\ 1.0000 & 1.0250 \\ 1.0000 & 1.0250 \\ 0.2001 & 0.5858 \\ 1.3000 & 2.5000 \\ 3.2000 & 1.2539 \\ -0.0001 & 64.294 \\ 3.4000 & 2.1030 \\ 1.2000 & 0.9619 \\ 1.5000 & 1.3000 \end{matrix} \right] \\
 \end{matrix}
 \end{matrix}
 \quad (1.14)$$

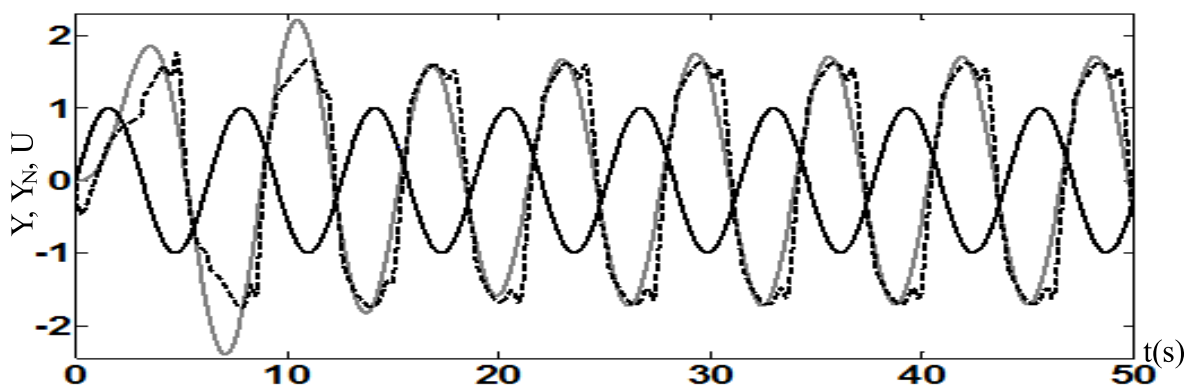


Figure 1.11. Résultat de l'identification directe (— : entrée, — : sortie de système, - - : sortie de RNA).

Selon les résultats ci-dessus, la réponse de RNA suit la réponse de système dynamique, ce qui montre la capacité d'identification des RNA.

5.3.2. Identification neuronale inverse

Un réseau de neurones à une couche cachée est utilisé pour identifier le modèle inverse, et la sortie de RNA sera donnée par :

$$U_N = W^T \sigma(V^T \chi) \quad (1.15)$$

Avec U_N est la sortie du RNA, W est la matrice des poids de la couche cachée/ couche de sortie, V est la matrice de la couche d'entrée/ couche cachée, La fonction d'activation $\sigma(\cdot)$ est sa dérivée sont données par (1.11) et (1.12) respectivement. Les signaux d'identification sont représentés par la Figure (1.10). Le RNA est composé de 18 neurones dans la couche cachée, un seul neurone dans la couche de sortie et deux neurones dans la couche d'entrée, avec le vecteur d'entrée $\chi = [Y \ U \ U_N]^T$ où Y et U sont la sortie et l'entrée du système respectivement.

Le gain d'apprentissage $\mu=0.11$. L'initialisation aléatoire des poids est donnée par l'équation (1.15) Après 80 itérations d'apprentissage par l'algorithme de rétro-propagation les poids W et V obtenues sont données par (1.15).

De la même manière comme avec l'identification directe, un signal sinusoïdal est utilisé comme entrée de système dynamique pour la validation de l'identification. La Figure 1.12 représente le signal d'entrée de système et le signal de sortie de RNA implémenté.

$$\begin{array}{c}
 \begin{bmatrix} 1 \\ 3.4 \\ 2 \\ 0.1 \\ 1.2 \\ 3 \\ 0.1 \\ 1 \\ 3.2 \\ 1.1 \\ 0 \\ 0.2 \\ 9 \\ 7.1 \\ 1.1 \\ 2.1 \\ 0.2 \\ 0.9 \end{bmatrix}, V^T = \begin{bmatrix} 1 & 2 & 1 \\ 0.3 & 0.1 & 0.3 \\ 0.4 & 0 & 0.4 \\ 1.1 & 0.1 & 1.1 \\ 0.3 & 2 & 0.3 \\ 0.1 & 2 & 0.1 \\ 1 & 1 & 1 \\ 11 & 11 & 11 \\ 3.2 & 3.2 & 3.2 \\ 0.1 & 0.1 & 0.1 \\ 11 & 11 & 11 \\ 3.2 & 3.2 & 3.2 \\ 0.1 & 0.1 & 0.1 \\ 11 & 11 & 11 \\ 3.2 & 3.2 & 3.2 \\ 0.1 & 0.1 & 0.1 \\ 11 & 11 & 11 \\ 3.2 & 3.2 & 3.2 \end{bmatrix} \xrightarrow{\text{après apprentissage}} \begin{bmatrix} -0.1374 \\ 0.7584 \\ 0.0538 \\ -2.2768 \\ 0.2518 \\ 1.3403 \\ -1.6615 \\ -0.8048 \\ 1.6422 \\ -0.8335 \\ -1.7732 \\ -1.3377 \\ 4.5950 \\ 5.1852 \\ -0.4344 \\ 0.2220 \\ -1.5800 \\ -0.6317 \end{bmatrix}, V^T = \begin{bmatrix} 1.0000 & 1.9160 & 1.3239 \\ 0.2999 & 0.4555 & 1.0907 \\ 0.4000 & 0.2353 & 0.8585 \\ 1.1001 & -0.8818 & 1.9582 \\ 0.3000 & 2.1301 & 0.4831 \\ 0.0999 & 2.2618 & 0.9591 \\ 1.0001 & -0.0212 & 1.9009 \\ 11.0000 & 10.8857 & 11.1353 \\ 3.1999 & 3.4625 & 2.8764 \\ 0.1001 & -0.3023 & 0.3676 \\ 11.0001 & 10.8763 & 11.1462 \\ 3.2001 & 2.6738 & 3.7956 \\ 0.0999 & 1.1034 & 11.2667 \\ 10.9999 & 11.0764 & 10.8846 \\ 3.2001 & 2.9970 & 3.4857 \\ 0.1000 & 0.4512 & 0.8418 \\ 11.0001 & 10.8797 & 11.1429 \\ 3.2001 & 2.9417 & 3.5430 \end{bmatrix} \quad (1.16)
 \end{array}$$

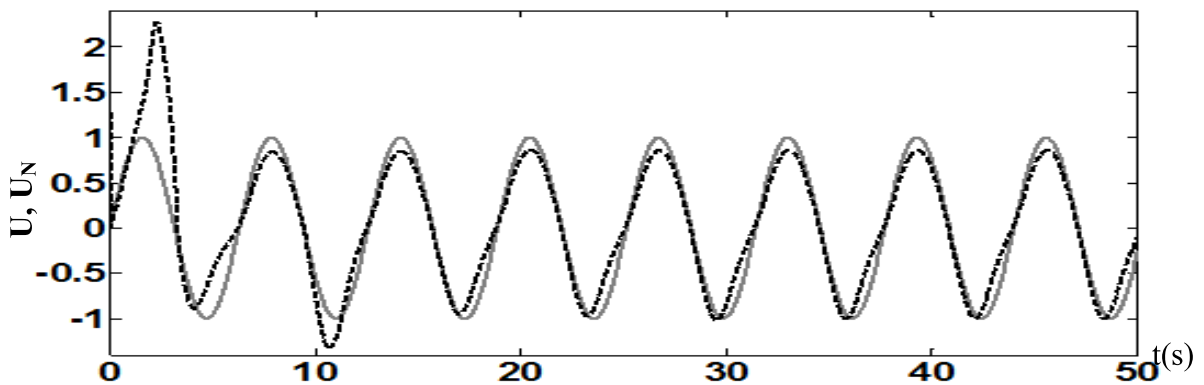


Figure 1.12. Résultat de l'identification de système inverse (— : entrée, - - : sortie de RNA).

Selon le résultat obtenu, la sortie de RNA est très proche du signal d'entrée, ce qui prouve l'efficacité de RNA dans l'identification de la dynamique inverse des processus.

Il est à noter que l'amélioration des résultats d'identification (directe et inverse) est possible : si le gain d'apprentissage est adaptatif, par le changement de nombre de couches cachés et le nombre de leurs neurones, on peut même utiliser des critères pour déterminer le nombre approprié de neurones et de couche cachées.

6. Conclusion

Ce chapitre a été consacré pour la description de l'approximateurs universel appelé réseaux de neurones artificiels. L'origine biologique de cet approximateur est présentée, où l'analogie avec la cellule nerveuse a donné naissance au neurone formel qu'est l'unité de base des réseaux de neurones artificiels actuels. Le développement des RNAs et les règles d'apprentissage ont été présentés brièvement. Le théorème de l'approximation universel par les RNAs des fonctions inconnues est donné. L'algorithme d'apprentissage, rétro-propagation et son utilisation dans des schémas d'identification directe et inverse des systèmes dynamiques ont été exposés..

Chapitre II

Notions sur les commandes stabilisantes des systèmes non linéaires

1 Introduction

Dans la synthèse des lois de commande, l'objectif est d'obtenir les performances désirées. La stabilisation du processus est la première performance à satisfaire par une technique de commande. La stabilité des systèmes linéaires dépend des valeurs propres de la matrice fondamentale. Cependant, l'étude de la stabilité pour les systèmes non linéaires présente un grand challenge. La théorie développée par le mathématicien russe Alexandr Mikhailovich Lyapunov à la fin de 19^{ième} siècle est la plus utilisée pour l'étude de la stabilité [Slo91]. Cette théorie présente deux méthodes ; la méthode indirecte (où de linéarisation) et la méthode directe [Slo91]. La première méthode donne des conclusions sur la stabilité locale autour des points d'équilibre à partir de l'approximation linéaire du système. La deuxième méthode est basée sur un principe physique, de l'énergie d'un système (linéaire ou non linéaire) se dissipe avec le temps pour qu'il puisse se stabiliser à son point d'équilibre [Kha96]. Cette propriété, permet l'étude de la stabilité à partir l'étude des variations d'une fonction définie positive représentative de l'énergie.

Dans ces dernières décennies, plusieurs techniques de commande sont développées à la base de la méthode directe de Lyapunov. L'objectif est de synthétiser une commande qui garantit la décroissance de la fonction candidate de Lyapunov [Bou07, Kha96, Slo91]. Le même principe est utilisé avec la technique de backstepping [Cha96], toutefois, la fonction de Lyapunov est construite récursivement par l'intermédiaire des commandes virtuelles afin d'obtenir l'expression finale de la loi de commande. Cependant, ces méthodes utilisent le modèle dynamique du système à commander, ce qui provoque des dégradations dans la présence des

incertitudes. Les techniques de commandes robustes telles que la commande par mode de glissement [Buh86] donnent une solution à ce genre de problème malgré l'utilisation de modèle dynamique. Puisque le terme discontinu permet de compenser l'effet des incertitudes et rend la commande robuste. Cependant, cette technique présente quelques problèmes, comme le phénomène de réticence dans la commande et la sortie et la nécessité de connaître le modèle au moins partiellement. La commande adaptative est une technique donnant une solution aux problèmes des techniques classiques. Le choix d'une loi de commande et d'un mécanisme d'adaptation approprié, permet la compensation des incertitudes et des perturbations, ce qui donne une robustesse à cette technique. Néanmoins, cette technique présente quelques inconvénients à savoir l'explosion des paramètres estimés sur les longues durées d'adaptation et la singularité de la commande dans la commande adaptative indirecte.

L'objectif de ce chapitre est de donner quelques notions de base et les théories utilisées pour étudier la stabilité des systèmes non linéaires. En plus, la méthode directe de Lyapunov est présentée. Ensuite, trois types de commandes stabilisantes sont exposés et leurs avantages et inconvénients sont énumérés. La première technique n'est pas robuste vis-à-vis les variations paramétriques et les erreurs de modélisation, c'est la commande par backstepping. La deuxième approche concerne la commande par mode de glissement. Malgré qu'elle utilise le modèle dynamique dans sa synthèse, cette technique présente une robustesse. Cependant, la commande par mode de glissement provoque le phénomène de réticence qui peut détruire les actionneurs. La troisième technique concerne l'objet de cette thèse est la commande adaptative. Une recherche bibliographique est effectuée sur les techniques et les méthodes proposées pour surmonter les problèmes de chaque techniques.

Ce chapitre est organisé comme suit : la section 2 comporte quelques définitions et notions sur la stabilité des systèmes non linéaire et l'énoncer de la méthode directe de Lyapunov. La synthèse d'une loi de commande par la technique de backstepping est présentée dans la section 3. La commande par mode glissant est présentée dans la section 4. Les notions de base de la commande adaptative sont données dans la section 5. Et la conclusion de chapitre est donnée dans la dernière section.

2 Stabilité des systèmes non linéaires

Contrairement aux systèmes linéaires où la stabilité est étudiée à partir des valeurs propres de la matrice fondamentale, les systèmes non linéaires présentent un grand challenge dans l'étude de stabilité. Depuis le 19^{ième} siècle, la théorie développée par le mathématicien russe Lyapunov la plus utilisée dans l'étude de la stabilité des systèmes non linéaires.

2.1. Théorèmes de stabilité

L'étude de stabilité au sens de Lyapunov des systèmes physiques [Slo91] utilise les définitions suivantes :

L'expression d'un système non linéaire est comme suit :

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

Avec : x est le vecteur d'état, t : est le temps, f est une fonction non linéaire de même dimension que x . Le point d'équilibre est noté x_e avec $f(x_e, t) = 0$.

Etudier la stabilité de point d'équilibre x_e est l'objectif principal dans chaque schéma de commande. Il est à noter que toutes les relations qu'on va exposer sont pour le cas d'un point d'équilibre nul et donc en utilisant un simple changement de variable les relations restent valables pour un point d'équilibre non nul.

• **Définition 2.1.** Stabilité et instabilité [Kha96],

1. L'origine est un point d'équilibre **stable** pour le système non linéaire (2.1) au sens de Lyapunov si : $\forall t > t_0, \forall \varepsilon > 0, \exists \delta(\varepsilon, t_0) > 0$ tel que : $\|x_0\| < \delta(\varepsilon, t_0) \Rightarrow \|x(t)\| < \varepsilon$.

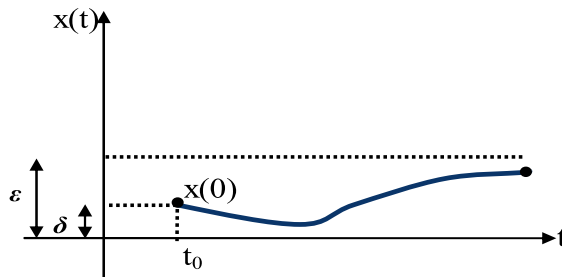


Figure 2.1. point d'équilibre stable.

Alors, la stabilité au sens de Lyapunov signifie que la trajectoire de système est arbitrairement proche du point d'équilibre. La Figure 2.1 illustre cette notion.

2. L'origine est un point d'équilibre **instable** au sens de Lyapunov s'il n'est pas stable au sens de Lyapunov comme il est montré sur la Figure 2.2.

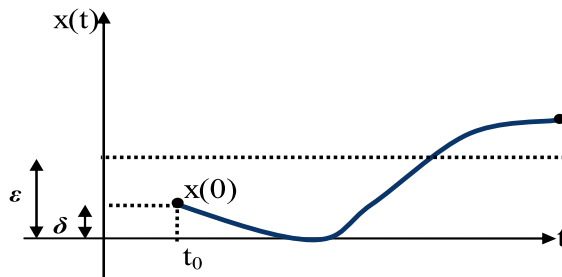


Figure 2.2. point d'équilibre instable.

• **Définition 2.2.** Stabilité asymptotique et exponentielle [Slo91],

La nécessité de la convergence vers l'origine (point d'équilibre) en automatique rend la stabilité au sens de Lyapunov insuffisante, alors, des nouvelles notions sont introduites telles que la stabilité asymptotique et exponentielle.

1. L'origine est un point d'équilibre **asymptotiquement stable** (Figure 2.3.a) s'il est stable au sens de Lyapunov et si : $\exists \delta(t_0) > 0$ tel que : $\|x_0\| < \delta(t_0) \Rightarrow \|x(t)\| \rightarrow 0$ quand $t \rightarrow \infty$.

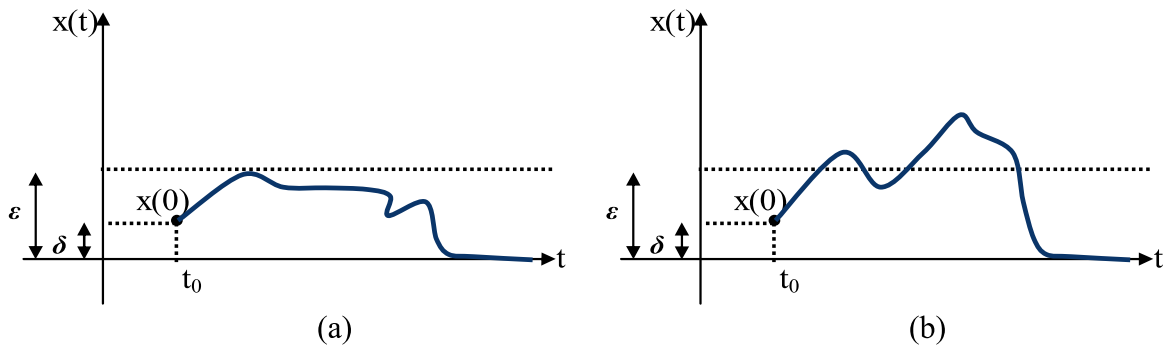


Figure 2.3. stabilité asymptotique, (a) :point d'équilibre asymptotiquement stable, (b) : point d'équilibre instable.

Cependant, la convergence du système vers le point d'équilibre, n'implique pas forcément la stabilité au sens de Lyapunov. La Figure 2.3.b, présente un exemple typique. Alors, si le système converge vers son point d'équilibre alors, on ne peut pas conclure sur son stabilité, pour cela, la notion suivante est introduite :

2. L'origine est un point d'équilibre **exponentiellement stable** (Figure 2.4) s'il existe deux constantes α et β positives telles que : $\forall t > t_0, \|x(t)\| < \alpha \|x_0\| e^{-\beta(t-t_0)}$.

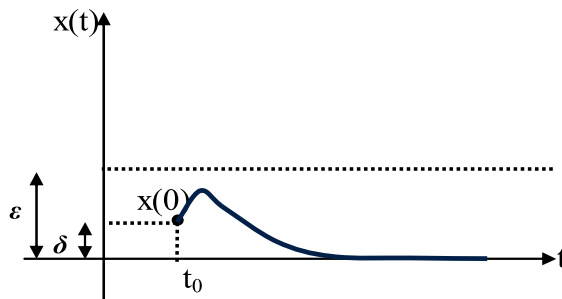


Figure 2.4. point d'équilibre exponentiellement stable.

• **Définition 2.3.** Stabilité uniforme [Slo91],

Ce type de stabilité est indépendant du temps initial t_0 d'où :

1. L'origine est un point d'équilibre **uniformément stable** pour le système non linéaire (2.1) si :
 $\forall \varepsilon > 0, \exists \delta(\varepsilon) > 0$ tel que : $\|x_0\| < \delta(\varepsilon) \Rightarrow \|x(t)\| < \varepsilon$ (Voir Figure 2.5).
2. Le point d'équilibre x_e de système non linéaire (2.1) est **uniformément asymptotiquement stable** au sens de Lyapunov s'il est uniformément stable et asymptotiquement stable.

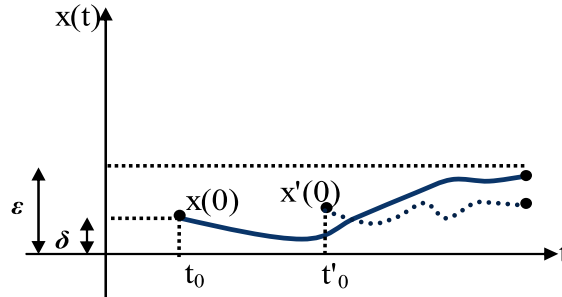


Figure 2.5. point d'équilibre uniformément stable.

• **Définition 2.4.** Bornitude [Lia07],

1. Le point d'équilibre de système non linéaire (2.1) est borné si :
 $\forall t > t_0, \forall \delta > 0, \exists \varepsilon(r, t_0) < \infty$ tel que : $\|x_0\| < \delta \Rightarrow \|x(t)\| < \varepsilon(r, t_0)$.
2. Le point d'équilibre de système non linéaire (2.1) est uniformément borné si :
 $\forall t > t_0, \forall \delta > 0, \exists \varepsilon(r) < \infty$ tel que : $\|x_0\| < \delta \Rightarrow \|x(t)\| < \varepsilon(r)$.
3. Le point d'équilibre de système non linéaire (2.1) est uniformément ultime borné si les états commençant au voisinage de points d'équilibre seront convergent vers une région, c'est-à-dire : $\forall \delta > 0, \forall \varepsilon > 0, \forall t > T(\delta, \varepsilon)$, tel que : $\|x_0\| < \delta \Rightarrow \|x(t)\| \leq \varepsilon$.

2.2. Stabilité local et stabilité global

Toutes les définitions données sont pour le cas local seulement, car les conditions initiales sont au voisinage de l'origine. Dans le cas où les conditions initiales sont éloignées de l'origine le comportement est inconnu, alors le concept de stabilité globale est nécessaire.

• **Définition 2.5.** [Slo91]

Si la stabilité asymptotique (où bien exponentielle) donnée par la définition 2.2 est vérifiée pour chaque condition initiale x_0 , le point d'équilibre est globalement asymptotiquement (où bien exponentiellement) stable.

La figure suivante illustre cette notion :

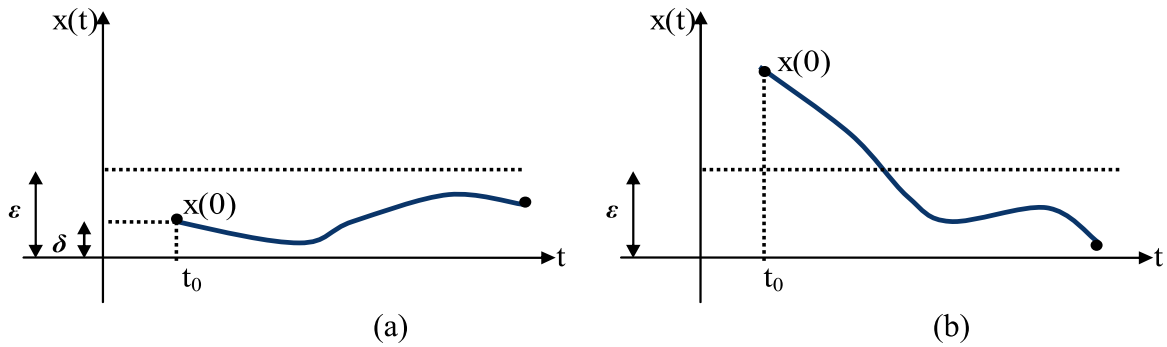


Figure 2.6. Stabilité (a) locale (b) globale.

2.3. Méthodes de Lyapunov

Les définitions présentées ne sont pas utilisables pour l'étude de la stabilité des systèmes physiques car un système physique est décrit généralement par un ensemble des équations différentielles complexes. Ses différents sous-systèmes sont fortement interconnectés qui rend la détermination de la solution analytique est une tâche très délicate, et plusieurs points d'équilibre peuvent exister.

Les deux méthodes développées par Lyapunov permettent l'étude de la stabilité et surmontent ces problèmes.

2.3.1. Méthode indirecte de Lyapunov

Cette méthode est basée sur le concept, que le comportement d'un système physique est similaire à celui de son approximation linéaire autour de ses points d'équilibre et donc étudier la stabilité locale.

Le développement en série de Taylor au voisinage du point d'équilibre x_e de système non linéaire (2.1) donne :

$$\dot{x} = \frac{\partial f(x,t)}{\partial x} \Big|_{x=x_e} (x - x_e) + f(x_e, t) + O(x) = A(t)(x - x_e) + f(x_e, t) + O(x) \quad (2.2)$$

Avec $O(x)$ regroupe les termes d'ordre supérieure à un.

La forme de l'équation (2.2) est semblable à celle du système linéaire et la stabilité de système (2.1) autour de son point d'équilibre, dépend de la matrice $A(t)$.

Le théorème de linéarisation de Lyapunov à une importance limitée car, il ne permet que d'étudier la stabilité locale et ne donne aucune information sur la stabilité globale de plus dû aux approximations du premier ordre il n'est pas possible de tenir compte de toutes les dynamiques non linéaires.

2.3.2. Méthode directe de Lyapunov

Cette méthode est basée sur le principe physique, que l'énergie d'un système quelconque se dissipe avec le temps quand il se stabilise à son point d'équilibre. En automatique ce principe est appliqué par l'utilisation d'une fonction définie positive ($L(x) > 0$) de l'énergie du système dite fonction candidate de Lyapunov. Ensuite, on doit vérifier que cette fonction est décroissante c'est-à-dire que sa dérivée est négative $\dot{L}(x) < 0$.

• **Définition 2.6.** [Slo91]

Le système non linéaire donné par (2.1) son point d'équilibre est l'origine, est globalement asymptotiquement stable s'il existe une fonction définie positive L telle que :

- $L(0) = 0, \quad L(x) > 0, \quad \forall x \neq 0,$
- $L(x) \rightarrow 0,$ quand $x \rightarrow \infty, L(x),$ radialement non bornée,
- $\dot{L}(0) = 0, \quad \dot{L}(x) < 0, \quad \forall x \neq 0.$

Dans le cas où $\dot{L}(x) \leq 0$, on peut conclure que le système est stable uniquement.

3 Commande par backstepping

Le Backstepping est une technique de détermination de la loi de commande. Il permet la reconstruction systématique et récursive de la fonction de Lyapunov et en conséquence la conception itérative de la loi de commande stabilisante. Cette technique est destinée pour une classe bien spécifique des systèmes non linéaires dites triangulaire inférieures [Cha96, Kha96], son modèle d'état est donné par les équations suivantes :

$$\begin{cases} \dot{x}_1 = f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)x_3 \\ \vdots \\ \dot{x}_i = f_i(x_1, x_2, \dots, x_i) + g_i(x_1, x_2, \dots, x_i)x_{i+1} \\ \vdots \\ \dot{x}_n = f_n(x_1, x_2, \dots, x_n) + g_n(x_1, x_2, \dots, x_n)u \end{cases} \quad (2.3)$$

Avec : f_i sont des fonctions non linéaires, g_i sont des gains de commande non nuls, x_i sont les variables d'états, u : est l'entrée de système et n est l'ordre de système.

3.1. Algorithme backstepping

L'algorithme backstepping et les étapes qu'on doit effectuer pour construire la loi de commande stabilisante sont présentés dans cette section.

La variable à contrôler y est définie par :

$$y = x_1 \quad (2.4)$$

Et l'erreur de poursuite e_1 par :

$$e_1 = y_d - y \quad (2.5)$$

Avec y_d est la sortie désirée de processus. Le choix de sortie implique que le système à un degré relatif r égale à son ordre n , alors, la conception de la loi de commande u nécessite n itérations, le détail de la $i^{\text{ème}}$ itération est comme suit :

- **Itération i .** avec $i = \overline{1:n}$, l'objectif de cette itération est de stabiliser le sous-système i donné par l'équation suivante :

$$\dot{x}_i = f_i(x_1, \dots, x_i) + g_i(x_1, \dots, x_i)x_{i+1}, \quad (2.6)$$

Selon la technique backstepping, le sous-système i est stabilisé par une commande virtuelle notée α_i . Son expression est choisie comme suit :

$$\alpha_i = \frac{1}{g_i} [\dot{\alpha}_{i-1} - f_i + g_{i-1}e_{i-1} + k_i e_i], \quad k_i > 0, \quad (2.7)$$

La fonction de Lyapunov candidate donnée par (2.8) est utilisée pour étudier la stabilité, donc :

$$L_i(e_1, \dots, e_i) = L_{i-1}(e_1, \dots, e_{i-1}) + \frac{1}{2} e_i^2 = \frac{1}{2} \sum_{j=1}^i e_j^2, \quad (2.8)$$

Dans ce cas, la première dérivée temporelle de L_i est calculée :

$$\dot{L}_i(e_1, \dots, e_i) = \dot{L}_{i-1}(e_1, \dots, e_{i-1}) + e_i \dot{e}_i = - \sum_{j=1}^{i-1} k_j e_j^2 + e_i (\dot{\alpha}_{i-1} - f_i - g_i x_{i+1} + g_{i-1} e_{i-1}) \quad (2.9)$$

Remplaçant l'expression de la commande virtuelle α_i dans l'équation (2.9) on obtient :

$$\dot{L}_i(e_1, \dots, e_i) = - \sum_{j=1}^i k_j e_j^2 + g_i e_i (\alpha_i - x_{i+1}), \quad (2.10)$$

Il est à noter que $x_{n+1} = u$, $\dot{\alpha}_0 = \dot{x}_{1d}$, $g_0 = 0$, $e_0 = 0$ et $L_0 = 0$; la $n^{\text{ème}}$ itération donne l'expression de la loi de commande stabilisante de système complet u , dont :

$$u = \alpha_n = \frac{1}{g_n} [\dot{\alpha}_{n-1} - f_n + g_{n-1}e_{n-1} + k_n e_n], \quad k_n > 0 \quad (2.11)$$

Cette commande assure que :

$$\dot{L}_n(e_1, \dots, e_n) = -\sum_{i=1}^n k_i e_i^2 \leq 0 \quad (2.12)$$

Selon la méthode directe de Lyapunov, le système (2.3) est globalement asymptotiquement stable par l'application de la loi de commande (2.11) [Slo91].

• **Remarque 2.1.** Il est à noter, que le choix de la sortie, implique le degré relatif de système et par conséquent le nombre des itérations effectuer. Dans le cas où la sortie y est une variable d'état :

$$y = x_i, \quad i = \overline{1:n} \quad (2.13)$$

Le degré relatif de système r est égale à :

$$r = n - i, \quad i = \overline{1:n} \quad (2.14)$$

Ce qui implique le passage par $n-i$ itérations.

3.2. Robustesse de backstepping

La loi de commande synthétisée par la technique de backstepping donnée par (2.11), n'est pas robuste car elle dépend principalement de modèle dynamique ce qui implique l'apparition des dégradations dans la présence des incertitudes. Des extensions robustes à la technique backstepping sont proposées ces dernières années. Dans [Bou07], un terme intégral a été ajouté au terme stabilisant $k_i e_i$ pour annuler les erreurs de poursuite. [Leu09] a utilisé les systèmes flous pour approximer les incertitudes de processus trouvées dans la dernière étape de backstepping. Dans [Zon12], le design de la loi de commande est basé sur les informations reçues d'un observateur d'état, et dans chaque étape, une loi d'adaptation pour les paramètres inconnus est déterminée. La conception d'une loi de commande par la technique de backstepping dans [Kos13] est associée avec la détermination récursive d'un mécanisme d'adaptation pour les paramètres inconnus. De plus, la robustesse de système est augmentée par l'ajout d'un terme discontinu dans l'étape finale, dont, la surface de glissement est la somme des termes proportionnels des erreurs produites dans chaque étape. Dans [Sun13], la technique backstepping a été développée pour une classe des systèmes non affine à la commande où un réseau de neurones artificiels de type fonction à base radiale a été utilisée pour estimer les incertitudes. Cette technique n'utilise pas les commandes virtuelles comme intermédiaires entre les différentes étapes et la commande de système est déterminée dans la dernière étape avec un mécanisme d'adaptation.

4 Commande par mode glissant

La commande par mode de glissement est une commande robuste. Pour cela, cette technique est appliquée à plusieurs types de systèmes tels que : les systèmes mécaniques, et électrotechniques.

Une commande par mode glissant typique est composée d'une partie nominale et une autre additionnelle utilisée pour compenser les incertitudes de modèle dynamique.

La conception de la loi de commande par mode de glissement, commence par la définition d'une surface de glissement représentative à dynamique désirée de processus. L'objectif est de synthétiser une loi de commande qui garantit la convergence vers la surface de glissement.

4.1. Conception de la loi de commande par mode de glissant

Les étapes les plus importantes dans la conception de la loi de commande par mode glissant sont données dans cette partie. Le modèle utilisé est de type SISO (Single Input Single Output) d'ordre n dont, son équation différentielle est donnée par :

$$y^{(n)} = f(X) + g(X)U \quad (2.15)$$

Avec : $X = [y \quad \dot{y} \quad \dots \quad y^{(n-1)}]^T = [x_1 \quad x_2 \quad \dots \quad x_n]^T \in \mathfrak{R}^n$ est le vecteur d'état, y : est la sortie de système, u : est l'entrée de système, $f(X)$: est une fonction non linéaire et $|g(X)| \geq \underline{g}$: est un gain de commande non linéaire et non nul.

4.1.1 Surface de glissement

La première étape dans la conception d'une loi de commande par la technique de mode de glissant consiste à définir une surface de glissement s , et pour cela l'erreur de poursuite sera définie en premier lieu par :

$$e = y_d - y \quad (2.16)$$

Avec y_d est la sortie désirée. Selon [Slo91], la surface de glissement est donnée par :

$$s = \left(\frac{d}{dt} + \lambda \right)^{n-1} e, \quad \lambda > 0 \quad (2.17)$$

Dans ce cas, la première dérivée de s par rapport au temps est donnée par :

$$\dot{s} = v - f(X) - g(X)u \quad (2.18)$$

Avec :

$$\begin{cases} v = y_d^{(n)} + \sum_{i=1}^{n-1} v_i e^{(i)} \\ v_i = \frac{(n-1)!}{(n-i)!(i-1)!} \lambda^{n-i} \end{cases}, i = \overline{1:n-1} \quad (2.19)$$

Selon l'expression (2.15), on déduit que la convergence de l'erreur de poursuite e et de toutes ses dérivées $e^{(i)}, i = \overline{1:n-1}$ vers zéros quand $t \rightarrow \infty$ est assurée si l'erreur filtrée s est nulle [Slo91]. De plus, la bornitude de cette dernière implique la bornitude de l'erreur de poursuite et ses dérivées comme suit [Slo91].

$$|s| \leq \Omega \Rightarrow |e^{(i)}| \leq 2^i \lambda^{i-n+1}, \quad i = \overline{0:n-1} \quad (2.20)$$

En utilisant cette propriété, le problème de commande de dimension n devient un problème de dimension 'une'. Alors, l'objectif est de synthétiser une loi de commande annulant l'erreur filtrée.

En utilisant la technique de mode glissant cet objectif est achevé par la commande u suivante :

$$u = u_{\text{gliss}} + u_{\text{eq}}, \quad (2.21)$$

Cette commande atteindre l'objectif souhaité en deux phases :

4.1.2 Commande glissante

La première phase consiste le mode de convergence vers la surface. La commande glissante u_{gliss} définie par l'équation (2.22) force le système à rejoindre la surface de glissement s durant un temps fini (Voir Figure 2.1).

$$u_{\text{gliss}} = k \text{signe}(s), \quad g(X)k > 0 \quad (2.22)$$

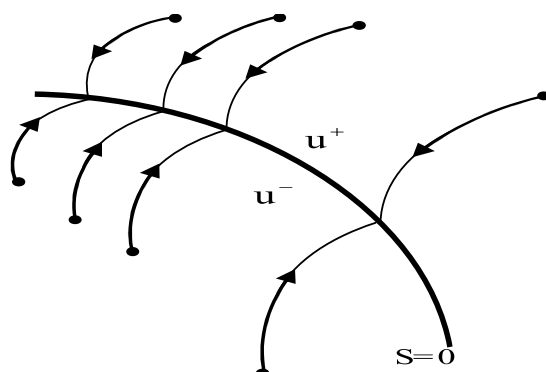


Figure 2.7. mode de convergence.

La robustesse de la commande par mode glissant vis-à-vis les incertitudes est due principalement au terme u_{gliss} .

4.1.3 Commande équivalente

Après la convergence vers la surface de glissement, le terme u_{gliss} est désactivé ce qui implique la fin de la première phase, et le début de la deuxième phase où la commande équivalente u_{eq} garantie le maintien sur cette surface jusqu'à atteindre l'origine [Hus09, Ram06].

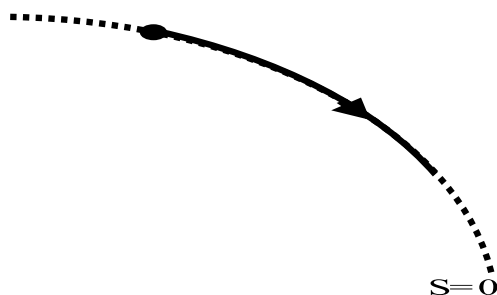


Figure 2.8. mode de glissement.

L'expression vérifiant cette condition est déterminée à partir de la dynamique de l'erreur filtrée, donnée par :

$$u_{\text{éq}} = g^{-1}(X)(v - f(X)) \quad (2.23)$$

4.2. Analyse de la stabilité

La méthode directe de Lyapunov est utilisée pour l'étude de la stabilité de système (2.15) après l'application de la commande par mode glissant donnée par (2.21)-(2.23), donc la fonction candidate de Lyapunov est comme suit :

$$L = \frac{1}{2} s^2 \quad (2.24)$$

La première dérivée temporelle de L est donnée par :

$$\dot{L} = s\dot{s} \quad (2.25)$$

Remplaçant la dynamique de la surface de glissement, la commande glissante et la commande équivalente par leurs expressions données par (2.18), (2.22) et (2.23) respectivement dans (2.25), on obtient :

$$\dot{L} = -g(X)k|S| \leq -\kappa|S| \leq 0 \quad (2.26)$$

Avec : $\kappa = k \underline{g}$, la surface de glissement est attractive dans un temps fini [Slo91] sa valeur est déterminée comme suit :

A partir (2.24) et (2.26), on peut écrire :

$$\dot{L} \leq -\kappa\sqrt{2L} \tag{2.27}$$

La résolution de cette équation différentielle donne :

$$\sqrt{L} \leq \sqrt{L(t_0)} - \frac{\kappa(t-t_0)}{\sqrt{2}} \tag{2.28}$$

Alors :

$$|s(t)| \leq |s(t_0)| - \kappa(t-t_0) \tag{2.29}$$

Selon l'équation ci-dessus, l'erreur filtrée s'annule à l'instant :

$$t = \frac{|s(t_0)|}{\kappa} + t_0 \tag{2.30}$$

4.3. Problèmes de la commandes par mode glissant

Malgré la robustesse de la technique de commande par mode de glissement, elle comporte trois inconvénients qui sont :

- La commande équivalente nécessite au moins une connaissance partielle de modèle dynamique pour le maintien sur la surface de glissement.
- Le phénomène de réticence est le deuxième inconvénient de cette technique, il est causé par le terme discontinu de la commande glissante car la commutation entre deux valeurs n'est pas instantanée en pratique (hystérésis), (Voir Figure 2.9).
- La compensation des incertitudes est garantie non seulement par le terme discontinu mais par la valeur élevée de son amplitude k. Cependant, le phénomène de réticence est amplifié, ce qui pose un problème de choix optimale de l'amplitude de la commande glissante.

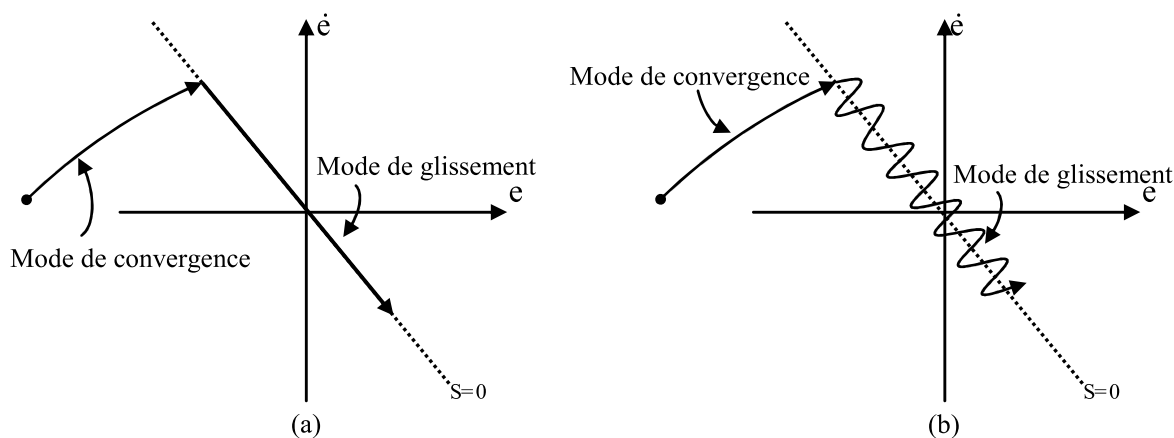


Figure 2.9. Phénomène de réticence : (a) : Mode de glissement idéal, (b) : Mode de glissement réel.

En se basant sur les approximateurs universel plusieurs solutions ont été proposées dans la littérature pour traiter les problèmes de la commande par mode glissant.

Dans [Cha01, Hua08, Ond01, Sef12], la commande équivalente est supposée totalement inconnue d'où : les réseaux de neurones artificiels et les systèmes flous ont été employés pour l'approximer en ligne par un mécanisme d'adaptation adéquat.

Le phénomène de réticence causé par la commande glissante est généralement résolu par l'implémentation d'une fonction approximative à la fonction signe telles que saturation, tanh,...etc., comme dans [Ha99, Kha96, Slo91, Top07],

Dans [Ais09, Bec05, Bou10], les systèmes flous de type Mamdani [Bou00] ont été utilisés pour fournir une valeur appropriée au gain de glissement, et dans [Bou12, Hun04, Hua08], l'approximateur universel RNA a été employé pour approximer la commande glissante idéale.

5 Commande adaptative

La modélisation des systèmes utilise souvent des simplifications et des hypothèses, ce qui résulte un modèle dynamique qui ne copie pas fidèlement le processus physique. De plus, la présence des perturbations internes et externes, rend les techniques de commande à gain fixe inefficace en termes de performances. La commande adaptative traite se problème par l'emploi des mécanismes d'estimation pour les termes et les dynamiques incertains.

5.1. Commande adaptative à modèle de référence

La commande adaptative à modèle de référence (CAMR) est composée principalement de système à commander, un modèle de référence qui décrit les performances désirées et un contrôleur à paramètres variables ajustés par un mécanisme d'adaptation, dont, le signal d'entraînement est la différence entre la sortie de système et la sortie de modèle de référence. Le schéma représentatif de la CAMR est représenté dans la Figure 2.10.

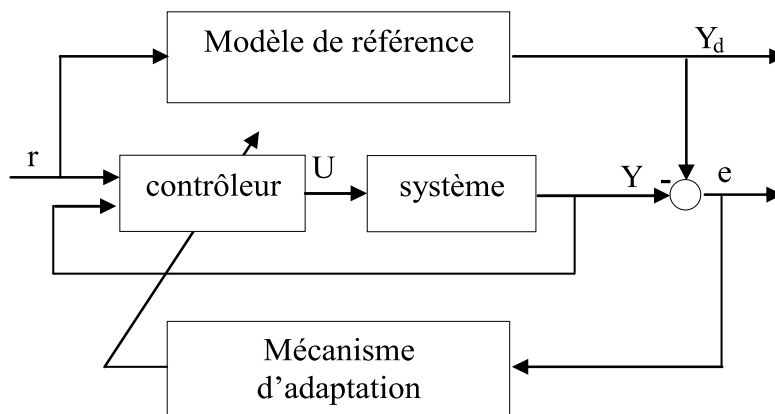


Figure 2.10. Commande adaptative à modèle de référence.

5.2. Commande adaptative auto ajustable

La commande adaptative auto ajustable (CAAA) utilise les paramètres de système. Ces derniers sont inconnus, donc ils sont remplacés par leurs estimés. Le schéma bloc de la commande adaptative auto ajustable est présenté sur la Figure 2.11.

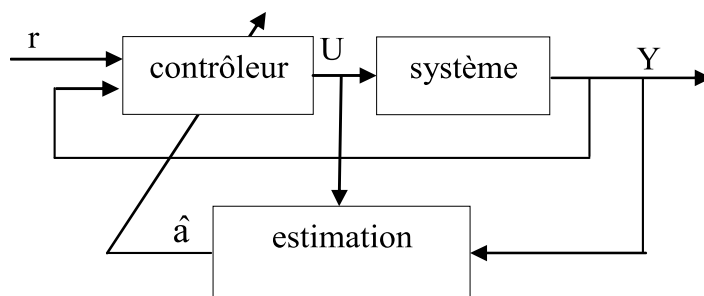


Figure 2.11. Commande adaptative auto ajustable.

5.3. Concept de la commande adaptative

La synthèse d'une loi de commande commence toujours par le choix de la structure du contrôleur et par la suite les paramètres de ce dernier sont calculés à partir des paramètres de système à commander. Le même principe est appliqué pour la commande adaptative sauf que les paramètres inconnus de système sont estimés par un mécanisme d'adaptation. Les étapes de la conception d'une loi de commande adaptative sont :

- Choisir une loi de commande contenant des paramètres variables,
- Choisir une loi d'adaptation pour ajuster les paramètres,
- Analyser les propriétés de la convergence du système de contrôle résultant.

5.4. Commande adaptative directe et commande adaptative indirecte

La technique de commande adaptative peut être classée en deux catégories selon, les connaissances disponibles et la structure de la loi de commande choisie. A partir de schéma de commande adaptative auto ajustable donné par Figure 2.11, le calcul de la loi de commande est fait après l'estimation des paramètres inconnus de système. Ce type de schéma représente à la première catégorie qu'est la commande adaptative indirecte. Toutefois, la synthèse d'une loi de commande adaptative peut être effectuée autrement, où les paramètres de contrôleur sont utilisés pour reparamétriser le modèle dynamique de processus et un mécanisme d'adaptation est implémenté pour approximer le contrôleur. Cette technique est la deuxième catégorie qu'est la

commande adaptative directe. Le même principe est appliqué pour une commande adaptative à modèle de référence [Slo91].

Cependant, la singularité de la commande quand le gain de commande estimé prend la valeur zéro est un problème souvent rencontré avec la commande adaptative indirecte. Pour traiter ce problème [Xu01a, Xu01b, Kos99] ont modifiés la structure de la loi de commande, où, le gain de commande estimé est remplacé par un autre terme qui non s'annule pas et par conséquent un terme robuste est associé pour compenser l'effet supplémentaire de cette modification. Les algorithmes de projection offrent une solution à la singularité de la commande comme dans [Cha01]. Le principe est de modifier les lois d'adaptation quand les paramètres estimés de gain de commande sont sur les bornes d'une zone désirée et leurs dérivées sont dirigées à l'extérieur de cette zone, alors, les dérivés sont projetées sur un hyperplan tangent au bornes de cette zone. Dans [Tin07] le problème est surmonté par l'estimation de l'inverse de gain de commande.

5.5. Commande adaptative par les approximateurs universels

Les techniques de commande adaptative conventionnelles sont généralement applicables pour les systèmes linéaires [Ioa84] où bien pour les systèmes linéairement paramétrés [Slo91]. D'où, l'objectif est de chercher un mécanisme d'adaptation pour les paramètres incertains en fonction des non linéarités connues du système à commander. Cependant, les systèmes physiques sont généralement fortement interconnectés et non linéaires, ce qui rend l'application des techniques adaptatives conventionnelles basées sur les contrôleurs linéaires inefficace et dans certains cas impossible. Les approximateurs universels tels que les systèmes flous [Des95] et les réseaux de neurones artificiels [Ioa06] donnent une solution appropriée à ce genre d'applications. Les approximateurs universels sont implémentés soit en ligne ou hors ligne pour l'approximation des non linéarités [Ioa06].

L'approximateur universel utilisé dans cette thèse est le réseau de neurones artificiel. Il a été utilisé pour la première fois au début des années quatre-vingt-dix [Ioa06]. Cet approximateur a attiré l'attention des chercheurs pour les raisons de cette attractivité sont résumées dans les points suivantes [Bro92] :

- Il a une grande capacité d'apprentissage en utilisant les données entrées/ sorties,
- Il peut approximer n'importe quelle fonction non linéaire continue d'après le théorème d'approximation universel,
- Il a une capacité de généralisation, c'est-à-dire extrapolé pour des nouvelles entrées.

L'importance de ces caractéristiques augmente avec l'augmentation des complexités et le besoin de synthétiser un contrôleur avec le minimum d'information sur le processus à commander et son environnement.

5.6. Divergence des paramètres

La commande adaptative souffre de problème de divergence des paramètres estimés (explosions) en longues durées d'adaptation si le processus est sous l'effet des incertitudes non paramétriques telles que : les bruits de mesure et les perturbations extérieures [Slo91].

Ce problème dans, [Goa02, Ioa06], est traité par l'utilisation des algorithmes de projection pour garantir que les paramètres estimés restent à l'intérieur d'une zone désirée. L'utilisation de la zone morte [Far06, Ioa06] est une autre solution pour ce genre de problèmes, où, l'adaptation est arrêtée si l'erreur d'estimation est relativement petite. Avec σ -switching [Ioa84, Ioa06], un terme supplémentaire est utilisé en cas où l'erreur d'estimation est à l'extérieur d'un intervalle donné. Cependant, ces trois techniques nécessitent la connaissance a priori de certaines informations qui sont difficiles ou même impossible de les déterminer en pratique. La technique σ -modification [Die08, Die09], comporte un terme additionnel dans le mécanisme d'adaptation, qu'est proportionnel aux paramètres estimés. Cependant l'inconvénient de cette méthode qu'elle n'assure pas une erreur de poursuite nulle. L' e -modification [Das09, Rai08] est une extension de σ -modification mais avec un terme additionnel dépendant de l'erreur de poursuite. Néanmoins, la valeur choisie pour le terme additionnel pose un compromis entre la poursuite et la divergence des paramètres. Les petites valeurs garantissent une meilleure poursuite, mais le risque de divergence des paramètres estimés est toujours grand. Dans [Bou10, Bou12a, Coz06], ce problème a été résolu par l'implémentation de deux approximateurs de même type pour estimer les mêmes incertitudes. Le premier approximateur est entraîné par l'erreur de poursuite et l'apprentissage est de type supervisé et le deuxième est entraîné par la différence entre les deux approximateurs.

6 Conclusion

Dans ce chapitre quelques définitions et notions importantes sur la stabilité ont été présentées. La méthode directe de Lyapunov pour l'analyse de la stabilité a été donnée. Trois techniques de commande stabilisantes des systèmes non linéaires ont été présentées. La première, est dédiée aux systèmes triangulaires inférieurs, c'est la commande par backstepping. Nous avons constaté que le modèle dynamique de système à commander est utilisé dans la loi commande par backstepping ce qui limite cette technique à la commande des systèmes connus.

La deuxième technique est la commande par mode de glissement. Malgré sa robustesse vis-à-vis les perturbations, cette technique comporte quelques inconvénients tels que : le phénomène de réticence et l'exigence de connaître le modèle dynamique au moins partiellement. La troisième technique est la commande adaptative ; les deux schémas conventionnels et les étapes de synthèse ainsi que les limitations de cette technique ont été présentées. Nous constatons que la commande adaptative est adaptée à la commande des systèmes linéaires et non linéaires inconnus, complexes et soumis à des perturbations internes et externes, de plus, elle donne une solution aux limitations des techniques de commandes classiques.

Chapitre III

Commande adaptative d'un système non linéaire sous-actionné (Quadrotor)

1. Introduction

L'avancement technologique en aéronautique et la commande des systèmes autonomes a permis une vaste utilisation des drones dans les domaines civils et militaire [Bal12, Erg12, Zem09]. Ces robots volant peuvent transporter des caméras, des équipements de communication ou d'autres dispositifs. Afin d'achever des missions de reconnaissance, surveillance, de recherche d'informations voire des opérations de combat et de réaliser des tâches pénibles et répétitives et l'accès à des zones non accessibles et dangereuses pour l'être humain [Adi07, Zem09].

L'hélicoptère à quatre rotors ou le quadrotor est composé de quatre rotors placés aux extrémités d'une croix. Le quadrotor est un système sous-actionné à six degrés de liberté et quatre entrées seulement [Bou07]. Pour surmonter ce manque, plusieurs travaux ont été proposés des techniques afin de pouvoir commander chaque sortie indépendamment de l'autre, dont, l'idée principale était basée sur l'utilisation indirecte des angles de roulis et tangage comme contrôleurs pour le mouvement dans l'espace [Bou11b, Cas04, Die08, Mad07, Ond07]. D'autres types de commandes ont été synthétisées pour le quadrotor telles que : mode glissant et backstepping [Bou11a, Nag13]. Une autre technique par mode glissant mais basée sur l'observation des bruits est développée dans [Ond07]. Dans cette technique le mouvement vertical et le déplacement en 2D sont commandés par la force de portance et les angles de roulis et tangage respectivement. Une étude comparative entre le PID et la technique LQ pour la stabilisation d'attitude est faite dans [Bou04]. Les résultats de cette étude ont montré l'avantage de la commande PID par rapport à la commande LQ si le système est sous une perturbation mineure comparant à la technique LQ.

Une autre application du PID est effectuée dans [Oos12] pour la commande de l'attitude et l'altitude. Dans [Cas04], les mouvements, vertical et lacet sont commandés par la technique backstepping par contre les autres sorties ont été contrôlées par l'algorithme des saturations imbriquées [Tee92]. Cependant, toutes les techniques citées nécessitent la disponibilité du vecteur d'état, les phénomènes physiques agissant sur lui et par conséquent la connaissance exacte du modèle dynamique qui est difficile ou impossible à déterminer en pratique.

La synthèse des commandes adaptatives pour la stabilisation de quadrotor a été le sujet de plusieurs travaux de recherche. Le principe consiste à remplacer les termes inconnus par leurs estimés adaptatif [Mor06, Lee09]. Cependant, il faut prendre en considération que le quadrotor, est sous-actionné. Le travail développé dans [Lee09] a choisi les trajectoires désirées des angles de roulis et de tangage comme des surfaces de glissement pour le plan. En se basant sur le travail dans [Tee92], [Cas04] a proposé la simplification de modèle à un ensemble des intégrateurs en cascade. Dans [Bou07], deux commandes virtuelles ont été ajoutées pour commander le déplacement en 2D. Même idée est appliquée dans [Die08], mais avec des commandes virtuelles adaptatives, dont, l'approximation était avec un réseau de neurones artificiels. Des trajectoires désirées en roulis et en tangage dans [Das09] sont dérivées à partir de la force de portance agissante sur les trois axes.

Dans ce chapitre, nous proposons une commande adaptative indirecte pour le système sous actionné « quadrotor ». Le schéma de commande contient trois phases importantes, dans la première, deux commandes virtuelles sont utilisées pour rendre le gain de commande inversible. La deuxième phase, comporte le mécanisme d'adaptation utilisé pour estimer le modèle dynamique inconnu, et les commandes virtuelles. Dans la troisième phase, nous proposons des contraintes holonomes pour définir les trajectoires désirées en roulis et en tangage. Ces contraintes sont basées sur les relations physiques entre les sorties du quadrotor. L'analyse de la stabilité en boucle fermée et les lois d'adaptations utilisant la théorie de Lyapunov. Ensuite, des simulations sont réalisées pour valider la stratégie de commande proposée.

Ce chapitre est organisé comme suit : le modèle dynamique et le principe de fonctionnement du quadrotor sont donnés dans la section 2. Position de problème est donnée dans la section 3. La commande adaptative indirecte utilisée, les contraintes holonomes proposées et la preuve de stabilité sont présentés dans la section 4. Les résultats de simulation sont présentés dans la section 5 et nous terminons par une conclusion.

2. Principe de fonctionnement et dynamique du quadrotor

Un quadrotor est un véhicule aérien possède, quatre propulseurs installés aux extrémités d'une croix, et une boîte de contrôle placée au centre de cette croix. Chaque propulseur est composé d'un moteur à courant continu et des hélices bipales, identiques, à pas fixe et leurs taille est petite par rapport à la taille de l'appareil [Bou07, Zem09]. La rotation des rotors produit une force verticale dite force de portance ou de poussée responsable sur la sustentation de l'appareil [Bou07].

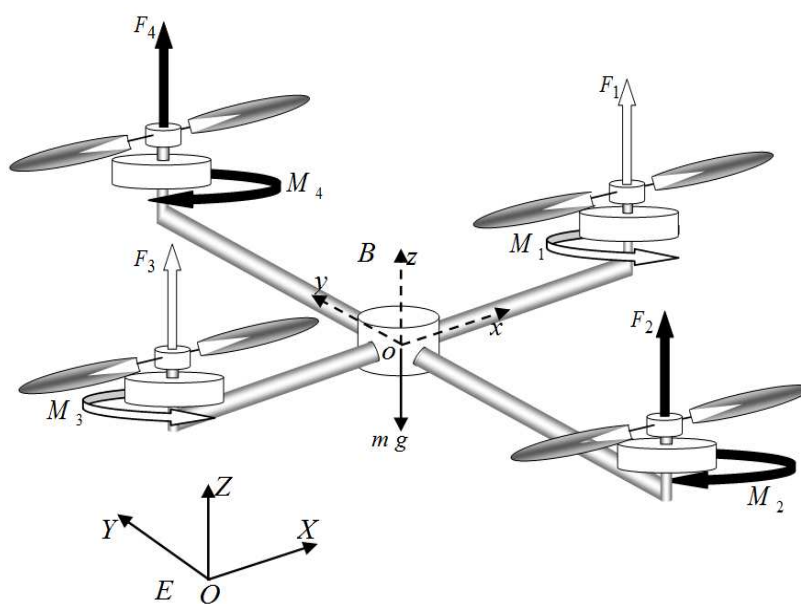


Figure 3.1. Configuration d'un quadrotor conventionnel.

La rotation des rotors avant et arrière (M_1 et M_3 respectivement) dans un quadrotor conventionnel est au sens des aiguilles d'une montre et les moteurs droit et gauche (M_2 et M_4 respectivement) tournent dans le sens opposé (Voir Figure 3.1), dont, l'objectif est d'éviter la rotation de quadrotor sur lui-même.

Il existe d'autre version des quadrotors comme dans [Zem06]. Dans cette structure les quatre rotors tournent dans la même direction mais, deux rotors sont inclinés par rapport à l'horizontal de manière à créer un couple de lacet.

La variation dans la vitesse de rotation dans chaque rotor est responsable sur les différents mouvements provoqués par le quadrotor. Le vol stationnaire est obtenu lorsque la somme des quatre forces de portance est égale au poids de quadrotor dans le module. Le mouvement vertical ascendant (descendant) est produit par l'augmentation (diminution) des forces de portance au même temps. L'existence d'une différence dans les vitesses de rotation des rotors latéraux (M_2 et M_4) est responsable sur le mouvement de roulis. Ce dernier est défini par une rotation autour

l'axe X couplé avec une translation selon l'axe Y. Le même principe est appliqué entre les rotors avant et arrière (M_1 et M_3) pour l'obtention du tangage qu'est un mouvement de rotation autour de l'axe Y couplé avec un mouvement de translation selon l'axe X. Le mouvement de Lacet c'est le mouvement de rotation de l'appareil sur elle-même. Ce type de mouvement est créé par l'augmentation (diminution) des vitesses de rotation des rotors avant et arrière tout en réduisant (amplifiant) les vitesses des rotors latéraux.

La Figure 3.2 montre les différents mouvements d'un quadrotor.

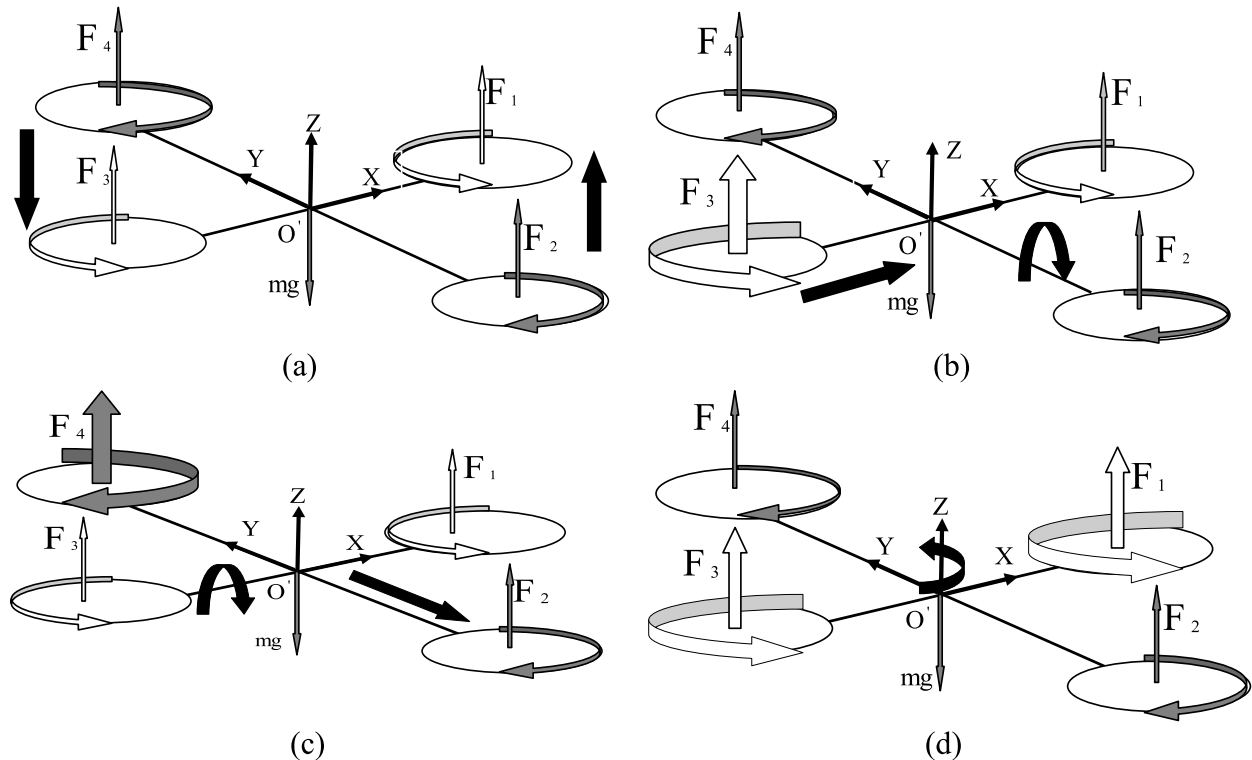


Figure 3.2. Mouvement du quadrotor : (a) : mouvement vertical, (b) : mouvement de tangage, (c) : mouvement de roulis, (d) : mouvement de lacet.

2.1. Modélisation du quadrotor

En utilisant le formalisme de Newton-Euler [Bou04], et en prenant les hypothèses suivantes en considération: « Les forces de portances et de frottement sont proportionnelles aux carrés des vitesses de rotation des rotors, le quadrotor est un corps rigide et symétrique, les hélices sont rigides et le centre de masse coïncide avec l'origine », les équations des mouvements linéaires et rotatifs sont formulées comme suit :

$$\begin{cases} m\ddot{\xi} = RF_p + P - F_f \\ J\dot{\eta} = M_p - M_{gh} - M_{gq} - M_f \end{cases} \quad (3.1)$$

D'où : $\xi = [x \ y \ z]^T$: est le vecteur de déplacement dans l'espace, $\eta = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$: sont les dérivées des angles d'Euler (angles de roulis, tangage et lacet respectivement), $J = \text{diag}(I_x, I_y, I_z)$: est la matrice d'inertie, m : est la masse de quadrotor, R : est la matrice de rotation ou bien la matrice d'Euler, qui sert à représenter l'orientation du quadrotor (voir figure 3.3), elle est donnée par :

$$R = \begin{bmatrix} C_\psi C_\theta & S_\theta C_\psi S_\phi - C_\phi S_\psi & C_\psi S_\theta C_\phi + S_\phi S_\psi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - S_\phi C_\psi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \in \mathbb{R}^{3,3} \quad (3.2)$$

Les expressions C_ϕ et S_ϕ sont une abréviation du $\cos(\phi)$ et $\sin(\phi)$ respectivement. Il est à noter que les angles en roulis et en tangage sont entre 0^0 à 20^0 [Zem06] c'est-à-dire on peut approximer : $\sin(\phi) = S_\phi \approx \phi$ et $\cos(\phi) = C_\phi \approx 1$.

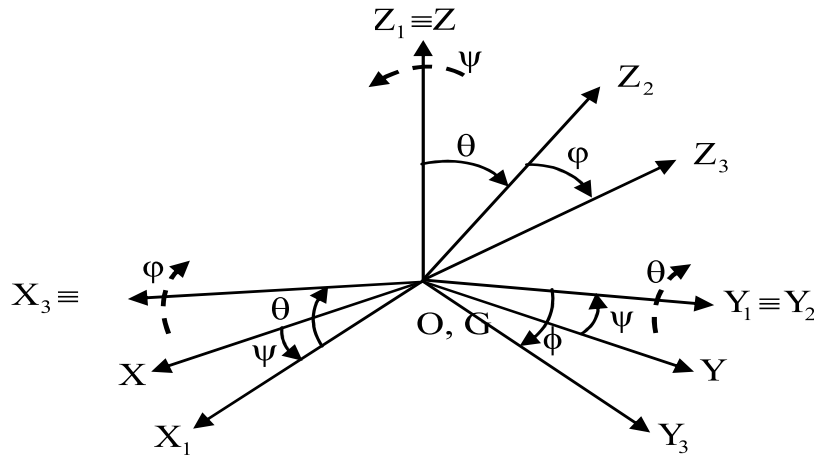


Figure 3.3. Angles d'Euler.

Les termes F_p , P et F_f présentent la force de portance, le poids du quadrotor et la force de frottement respectivement, leurs expressions sont données par :

$$\begin{cases} F_p = \begin{bmatrix} 0 & 0 & k_p \sum_{i=1}^4 \omega_i^2 \end{bmatrix}^T \\ P = \begin{bmatrix} 0 & 0 & -mg \end{bmatrix}^T, F_f = K_{ft} \dot{\xi} \end{cases} \quad (3.3)$$

Avec k_p : est le coefficient de portance, $K_{ft} = \text{diag}(k_{ftx}, k_{fity}, k_{ftz})$: est la matrice de frottement pour le mouvement de translation et ω_i est la vitesse de rotation de chaque rotor i .

Et les termes M_p et M_f sont les moments provoqués par la rotation des rotors et celui de frottement, ils sont donnés par :

$$\begin{cases} \mathbf{M}_p = \left[\begin{array}{ccc} lk_p(\omega_4^2 - \omega_2^2) & lk_p(\omega_3^2 - \omega_1^2) & k_d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{array} \right]^T \\ \mathbf{M}_f = \mathbf{K}_{fr} \begin{bmatrix} \dot{\phi}^2 & \dot{\theta}^2 & \dot{\psi}^2 \end{bmatrix}^T \end{cases} \quad (3.4)$$

Avec : k_d : est le coefficient de frottement, l : est la moitié de la longueur du bras de quadrotor et $\mathbf{K}_{fr} = \text{diag}(k_{frx}, k_{fry}, k_{frz})$: est la matrice de frottement pour le mouvement de rotation. Les moments \mathbf{M}_{gh} et \mathbf{M}_{gq} : sont les moments gyroscopiques des hélices et du quadrotor respectivement, ils sont définis comme suit :

$$\begin{cases} \mathbf{M}_{gq} = \boldsymbol{\eta} \wedge \mathbf{J} \cdot \boldsymbol{\eta} = \left[\begin{array}{ccc} (I_z - I_y)\dot{\psi}\dot{\theta} & (I_x - I_y)\dot{\phi}\dot{\psi} & (I_y - I_x)\dot{\phi}\dot{\theta} \end{array} \right]^T \\ \mathbf{M}_{gh} = \sum_{i=1}^4 \boldsymbol{\eta} \wedge \mathbf{J}_r \begin{bmatrix} 0 & 0 & \omega_i \end{bmatrix}^T = \left[\begin{array}{ccc} \Omega_r \mathbf{J}_r \dot{\theta} & -\Omega_r \mathbf{J}_r \dot{\phi} & 0 \end{array} \right]^T \end{cases} \quad (3.5)$$

Avec : \mathbf{J}_r est l'inertie des moteurs et Ω_r est :

$$\Omega_r = \sum_{i=1}^4 (-1)^{i+1} \omega_i \quad (3.6)$$

En remplaçant les expressions (3.2)-(3.6), dans le système d'équation (3.1), on trouve le modèle dynamique du quadrotor comme suit :

$$\begin{cases} \ddot{\phi} = \frac{u_\phi}{I_x} + \frac{(I_y - I_z)}{I_x} \dot{\psi}\dot{\theta} - \frac{\mathbf{J}_r \Omega_r}{I_x} \dot{\theta} - \frac{k_{frx}}{I_x} \dot{\phi}^2 \\ \ddot{\theta} = \frac{u_\theta}{I_y} + \frac{(I_z - I_x)}{I_y} \dot{\psi}\dot{\phi} + \frac{\mathbf{J}_r \Omega_r}{I_y} \dot{\phi} - \frac{k_{fry}}{I_y} \dot{\theta}^2 \\ \ddot{\psi} = \frac{u_\psi}{I_z} + \frac{(I_x - I_y)}{I_z} \dot{\phi}\dot{\theta} - \frac{k_{frz}}{I_z} \dot{\psi}^2 \\ \ddot{z} = \frac{1}{m} \left((C_\theta C_\phi) u_1 - k_{frz} \dot{z} \right) - g \\ \ddot{x} = \frac{1}{m} \left((C_\psi S_\theta C_\phi + S_\phi S_\psi) u_1 - k_{frx} \dot{x} \right) \\ \ddot{y} = \frac{1}{m} \left((S_\psi S_\theta C_\phi - S_\phi C_\psi) u_1 - k_{fry} \dot{y} \right) \end{cases} \quad (3.7)$$

Les entrées réelles de quadrotor $u_\phi, u_\theta, u_\psi, u_1$ sont en fonction des vitesses de rotation des quatre moteurs :

$$\mathbf{U} = \begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \\ u_1 \end{bmatrix} = \begin{bmatrix} 0 & -lk_p & 0 & lk_p \\ -lk_p & 0 & lk_p & 0 \\ k_d & -k_d & k_d & -k_d \\ k_p & k_p & k_p & k_p \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (3.8)$$

Selon (3.7), le quadrotor est un système sous actionné composé de six sorties et quatre entrées seulement (voir Figure 3.4).

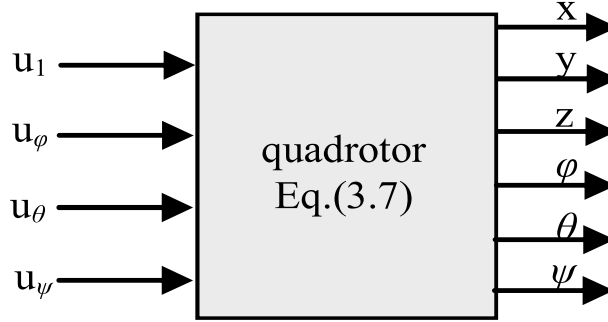


Figure 3.4. Quadrotor (système sous-actionné).

3. Position du problème

La forme matricielle du quadrotor donnée par l'équation (3.7), est comme suit :

$$\ddot{Y} = F(X) + G(X)U \quad (3.9)$$

Avec :

$$\begin{cases} Y = [\varphi \ \theta \ \psi \ z \ x \ y]^T \in \mathfrak{R}^{dl.1} \\ U = [u_\varphi \ u_\theta \ u_\psi \ u_1]^T \in \mathfrak{R}^{ne.1} \\ X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T \in \mathfrak{R}^{2dl.1} \\ \quad = [\varphi \ \dot{\varphi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ z \ \dot{z} \ x \ \dot{x} \ y \ \dot{y}]^T \end{cases} \quad (3.10)$$

Avec $dl=6$ est le degré de liberté de quadrotor, $ne=4$ est le nombre des entrées, $F(X)$ est une fonction non linéaire lisse est supposée inconnue, et elle est donnée par :

$$F(X) = \begin{bmatrix} F_\varphi(X) \\ F_\theta(X) \\ F_\psi(X) \\ F_z(X) \\ F_x(X) \\ F_y(X) \end{bmatrix} = \begin{bmatrix} \frac{(I_y - I_z)}{I_x} \dot{\psi} \dot{\theta} - \frac{J_r \Omega_r}{I_x} \dot{\theta} - \frac{k_{frx}}{I_x} \dot{\varphi}^2 \\ \frac{(I_z - I_x)}{I_y} \dot{\psi} \dot{\varphi} + \frac{J_r \Omega_r}{I_y} \dot{\varphi} - \frac{k_{fry}}{I_y} \dot{\theta}^2 \\ \frac{(I_x - I_y)}{I_z} \dot{\varphi} \dot{\theta} - \frac{k_{frz}}{I_z} \dot{\psi}^2 \\ -\frac{k_{ftz}}{m} \dot{z} - g \\ -\frac{k_{ftx}}{m} \dot{x} \\ -\frac{k_{fry}}{m} \dot{y} \end{bmatrix} \in \mathfrak{R}^{dl.1} \quad (3.11)$$

Et $G(X)$ est le gain de commande :

$$G(X) = \begin{bmatrix} \frac{1}{I_x} & 0 & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 & 0 \\ 0 & 0 & \frac{1}{I_z} & 0 \\ 0 & 0 & 0 & \frac{1}{m} C_\theta C_\varphi \\ 0 & 0 & 0 & \frac{1}{m} (C_\psi S_\theta C_\varphi + S_\varphi S_\psi) \\ 0 & 0 & 0 & \frac{1}{m} (S_\psi S_\theta C_\varphi - S_\varphi C_\psi) \end{bmatrix} \in \mathfrak{R}^{dl.ne} \quad (3.12)$$

Pour synthétiser une loi de commande U du système, les hypothèses suivantes sont considérées :

- **Hypothèse 3.1.** les vecteurs de sortie Y et l'état X sont mesurables.
- **Hypothèse 3.2.** la trajectoire désirée $Y_{di} = [x_{1d} \quad x_{3,5,7,9d} \quad x_{11d}]^T$ et ces dérivées sont lisses et bornées.

L'objectif principal de ce chapitre est de commander le système sous actionné quadrotor (les six sorties) en utilisant les quatre entrées seulement, en garantissant les performances du système bouclé même si le modèle dynamique est inconnu et/ou soumis sous l'effet des perturbations.

L'erreur de poursuite est définie par :

$$E = Y_d - Y = \begin{bmatrix} x_{1d} - x_1 \\ x_{3,5,7,9d} - x_{3,5,7,9} \\ x_{11d} - x_{11} \end{bmatrix} \in \mathfrak{R}^{dl.1} \quad (3.13)$$

Et l'erreur filtrée par:

$$S = [s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5 \quad s_6]^T = \dot{E} + \lambda E \quad (3.14)$$

Dont la matrice λ est donnée par :

$$\lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6), \lambda_i > 0 \quad (3.15)$$

Selon [Slo91], l'équation différentielle $S=0$ dispose une solution unique, qu'est, $E=0$, c'est-à-dire la convergence de l'erreur filtrée vers zéro garantit la convergence de l'erreur de poursuite et sa dérivée vers zéro aussi. De plus, la bornitude de l'erreur filtrée S implique la bornitude de l'erreur de poursuite E et \dot{E} [Slo91]. Alors, la synthèse d'une loi de commande U qui donne une

erreur filtrée S nulle sera l'objectif de commande. Dans ce contexte, la loi de commande idéale U^* garantissant cet objectif a la forme suivante :

$$U^* = G^{-1}(X)(v - F(X) + U^{pd}) \quad (3.16)$$

Avec :

$$\begin{cases} U^{pd} = K S \\ K = \text{diag}(k_1, k_2, k_3, k_4, k_5, k_6) \quad k_i > 0 \end{cases} \quad (3.17)$$

La dérivée temporelle de S est donnée par :

$$\dot{S} = v - F(X) - G(X)U \quad (3.18)$$

Avec v est un vecteur défini comme suit:

$$v = [v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6]^T = Y_d + \lambda \dot{E} \quad (3.19)$$

En remplaçant la loi (3.16) dans la dynamique de l'erreur filtrée (3.18), on aura:

$$\dot{S} = -K S \quad (3.20)$$

L'équation différentielle ci-dessus, a une solution unique qui est $S \rightarrow 0$, quand $t \rightarrow \infty$, et par conséquent $E \rightarrow 0$, et $\dot{E} \rightarrow 0$ quand $t \rightarrow \infty$ [Slo91].

Alors, la loi de commande dans (3.16) garantit la convergence de l'erreur de poursuite et ses dérivées vers zéro. Cependant, son implémentation est impossible pour deux raisons : l'incertitude de la fonction $F(X)$ et le gain de commande $G(X)$ n'est pas inversible dû à la nature sous actionnée. Pour surmonter ces inconvénients deux commandes virtuelles et des contraintes holonomes seront utilisées dans le schéma de commande adaptatif proposé dans la section suivante.

4. Synthèse d'une commande adaptative pour un système sous-actionné (quadrotor)

On, s'intéresse à la synthèse d'une loi de commande adaptative qui donne des solutions aux problèmes cités dans la section 3 toute en garantissant les performances en boucle fermée. La nouvelle loi de commande idéale est notée U_n^* et elle est proposée comme suit :

$$U_n^* = G_n^{-1}(X)(v - F(X) + g_v u_v + U^{pd}) \quad (3.21)$$

Avec : $G_n(X)$ est le nouveau gain de commande, défini par :

$$G_n(\mathbf{X}) = [G(\mathbf{X}) \quad \mathbf{g}_v] = \begin{bmatrix} \frac{1}{I_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{m} C_\theta C_\varphi & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{m} (C_\psi S_\theta C_\varphi + S_\varphi S_\psi) & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{m} (S_\psi S_\theta C_\varphi - S_\varphi C_\psi) & 0 & 1 \end{bmatrix} \in \mathfrak{R}^{dl,dl} \quad (3.22)$$

Avec :

$$\mathbf{g}_v = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (3.23)$$

Et l'inverse de ce nouveau gain de commande est formulé comme suit :

$$G_n^{-1}(\mathbf{X}) = \begin{bmatrix} I_x & 0 & 0 & 0 & 0 & 0 \\ 0 & I_y & 0 & 0 & 0 & 0 \\ 0 & 0 & I_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{m}{C_\theta C_\varphi} & 0 & 0 \\ 0 & 0 & 0 & -\frac{C_\psi S_\theta C_\varphi + S_\varphi S_\psi}{C_\theta C_\varphi} & 1 & 0 \\ 0 & 0 & 0 & -\frac{S_\psi S_\theta C_\varphi - S_\varphi C_\psi}{C_\theta C_\varphi} & 0 & 1 \end{bmatrix} \in \mathfrak{R}^{dl,dl} \quad (3.24)$$

Le vecteur \mathbf{u}_v contient les commandes virtuelles suivantes :

$$\mathbf{u}_v = [\mathbf{u}_x \quad \mathbf{u}_y]^T \quad (3.25)$$

Le vecteur \mathbf{U}_n^* présente la nouvelle commande idéale, qui est composé de six éléments comme suit :

$$\mathbf{U}_n^* = [\mathbf{u}_\varphi^* \quad \mathbf{u}_\theta^* \quad \mathbf{u}_\psi^* \quad \mathbf{u}_l^* \quad \mathbf{u}_x^* \quad \mathbf{u}_y^*]^T \in \mathfrak{R}^{dl,1} \quad (3.26)$$

Les commandes virtuelles \mathbf{u}_x et \mathbf{u}_y n'ont pas un sens physique et leurs utilisation est pour des raisons mathématiques pur. Dans le chapitre 4 nous utilisons des commandes virtuelles qui ont une signification physique expliquent les relations d'intersection entre l'orientation et le déplacement dans l'espace.

Pour monter la validité de cette loi de commande, \dot{S} est calculée en se basant sur l'expression dans (3.18) et en ajoutant près l'ajout des commandes virtuelles u_x et u_y , \dot{S} est formulée comme suit :

$$\dot{S} = v - F(X) - G_n(X)U_n + g_v u_v \quad (3.27)$$

En remplaçant la loi (3.21) dans (3.31), on aura :

$$\dot{S} = -K S \quad (3.28)$$

Donc, l'implémentation de U_n^* conduit à une erreur filtrée nulle quant $t \rightarrow \infty$ et par conséquent la convergence de E et sa dérivée \dot{E} vers zéro quand $t \rightarrow \infty$ [Slo91]. Cependant, cette loi a donné une solution pour le gain de commande de quadrotor qui n'est pas inversible, mais le problème d'incertitude de la fonction $F(X)$ est toujours posé, de plus, avec U_n^* même le vecteur u_v est inconnu. Alors, un mécanisme d'estimation en ligne, est utilisé ce qui donne un schéma de commande adaptative indirecte.

4.1. Commande adaptative indirecte

Tant que l'application de la loi de commande (3.21) est impossible, elle sera donc remplacée par la loi de commande réelle suivante :

$$U_n = G_n^{-1}(X) \left(v - \hat{F}(X) + g_v \hat{u}_v + U^{pd} \right) \quad (3.29)$$

Avec U_n est la commande appliquée $\hat{F}(X)$ et \hat{u}_v sont les estimées des fonctions $F(X)$ et u_v respectivement, dont :

$$\begin{cases} F(X) = \hat{F}(X) + \tilde{F}(X) \\ u_v = \hat{u}_v + \tilde{u}_v \end{cases} \quad (3.30)$$

Les termes $\tilde{F}(X)$ et \tilde{u}_v sont les erreurs d'estimation adaptative des fonctions $F(X)$, u_v respectivement.

Les lois d'adaptation proposées pour l'estimation en ligne de la fonction $F(X)$ et le vecteur des commandes virtuelles u_v sont données par :

$$\begin{cases} \dot{\hat{F}}(X) = -F_F S \\ \dot{\hat{u}}_v = F_v g_v^T S \end{cases} \quad (3.31)$$

Dont, les matrices $F_F = F_F^T > 0$, $F_v = F_v^T > 0$ sont les gains d'adaptation.

4.2. Contraintes holonomes

La loi de commande physique (3.29) donne des solutions aux problèmes d'incertitude de modèle dynamique et au gain de commande qui n'est pas inversible. Mais, les trois mouvements de déplacement (selon X, Y et Z) dépendent de la force de portance u_1 (voir équation (3.7)). Alors que, la loi de commande proposée U_n est contrôlé l'altitude seulement. Pour remédier à ce problème, des contraintes holonomes sont proposées.

Physiquement, le mouvement de quadrotor dans l'espace dépend des angles de roulis φ et de tangage θ . Dont, l'entrée u_φ produit un mouvement de rotation autour l'axe X, et ce mouvement provoque un autre qu'est le déplacement latéral (selon l'axe Y). Le même principe est appliqué pour l'obtention de mouvement selon l'axe X, dont la rotation autour de Y par un angle θ provoqué par l'entrée u_θ est responsable sur ce mouvement. Ces relations physiques forment les contraintes holonomes du système.

En se basant sur la technique de linéarisation entrée-sortie [Kha96, Slo91], les trajectoires désirées idéales en roulis et en tangage qui garantit la poursuite dans le plan sont proposées comme suit :

$$\begin{cases} \theta_d^* = \frac{m}{u_1^*} (\ddot{x}_d + a_1^x \dot{e}_x + a_2^x e_x - F_x(X)) \\ \varphi_d^* = -\frac{m}{u_1^*} (\ddot{y}_d + a_1^y \dot{e}_y + a_2^y e_y - F_y(X)) \end{cases} \quad (3.32)$$

Avec : \ddot{x}_d et \ddot{y}_d sont la deuxième dérivée des trajectoires désirées, e_x et e_y sont les erreurs de poursuites et a_1^x , a_2^x , a_1^y et a_2^y sont des constantes positives choisis par l'utilisateur.

En utilisant la propriété que les angles d'Euler φ et θ sont très petites, à partir la partie sous actionnée dans (3.7) devient :

$$\frac{m}{u_1^*} = \ddot{z}_d - F_z(X) \quad (3.33)$$

Cependant, l'utilisation directe de (3.36) est impossible car les fonctions $F_x(X)$, $F_y(X)$ et l'entrée u_1^* sont inconnues. Alors, les trajectoires désirées réelles en roulis et en tangage sont proposées comme suit :

$$\begin{cases} \theta_d = \frac{\ddot{x}_d + a_1^x \dot{e}_x + a_2^x e_x - \hat{F}_x(\mathbf{X})}{\ddot{z}_d - \hat{F}_z(\mathbf{X})} \\ \varphi_d = -\frac{\ddot{y}_d + a_1^y \dot{e}_y + a_2^y e_y - \hat{F}_y(\mathbf{X})}{\ddot{z}_d - \hat{F}_z(\mathbf{X})} \end{cases} \quad (3.34)$$

Le système d'équation ci-dessus forme les contraintes holonomes du quadrotor. Il est à noter qu'on peut trouver plusieurs types de contraintes holonomes dans la littérature comme dans [Bou07, Bou11b, Das09, Die08, Die09, Lee09]. Cependant, les expressions mathématiques changent d'un travail à un autre, mais elles sont toujours basées sur la dynamique de système.

- **Remarque 3.1.** L'entrée u_1 peut être aussi utilisée comme contrôleur pour le déplacement selon l'axe X ou bien Y et dans ce cas l'expression des contraintes holonomes va changer.

La Figure 3.5 résume le schéma de commande proposé dans ce chapitre.

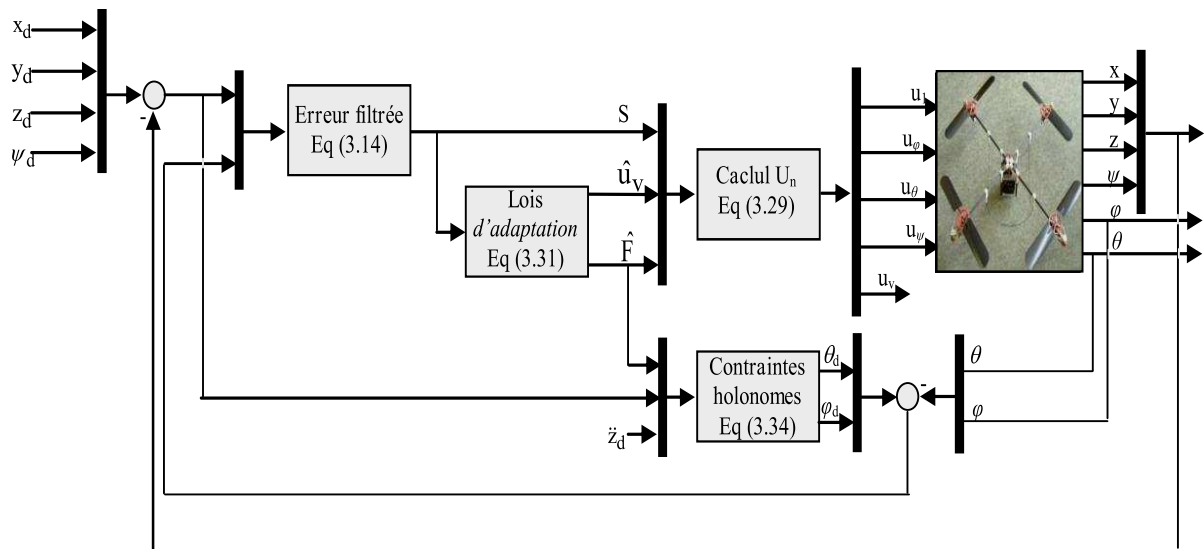


Figure 3.5. Schéma de commande adaptative pour un système non linéaire sous-actionné (Quadrotor).

4.3. Analyse de la stabilité

Dans cette partie, la stabilité du système bouclé est démontrée par la méthode directe de Lyapunov. L'expression dans (3.27) après l'implémentation de la loi de commande (3.29) devient :

$$\dot{S} = -K S - \tilde{F}(\mathbf{X}) + g_v \tilde{u}_v \quad (3.35)$$

• **Théorème 3.1.** Considérons le système (3.7), la loi de commande définie par (3.29), avec les lois d'adaptation (3.31), les contraintes holonomes données par (3.34) avec les hypothèses 3.1-3.2, le système bouclé est stable et tous les signaux sont bornés.

• **Preuve du Théorème 3.1.** pour la démonstration du théorème 3.1, la méthode directe de Lyapunov est utilisée. La fonction de Lyapunov candidate choisi est définie comme suit :

$$L = \frac{1}{2} S^T S + \frac{1}{2} \tilde{F}^T(X) F_F^{-1} \tilde{F}(X) + \frac{1}{2} \tilde{u}_v^T F_v^{-1} \tilde{u}_v \quad (3.36)$$

La première dérivée de (3.36) par rapport au temps est donnée par :

$$\dot{L} = S^T \dot{S} + \tilde{F}^T(X) F_F^{-1} \dot{\tilde{F}}(X) + \tilde{u}_v^T F_v^{-1} \dot{\tilde{u}}_v \quad (3.37)$$

Remplaçant l'expression donnée par (3.35) dans (3.37) on trouve :

$$\dot{L} = \left(-S^T K S + \tilde{F}^T(X) \left(F_F^{-1} \dot{\tilde{F}}(X) - S \right) + \tilde{u}_v^T \left(F_v^{-1} \dot{\tilde{u}}_v + g_v^T S \right) \right) \quad (3.38)$$

Et l'implémentation des lois d'adaptation (3.31) donne :

$$\dot{L} = -S^T K S \quad (3.39)$$

Selon le théorème de stabilité de Lyapunov [slo91], le système bouclé est asymptotiquement stable et que tous les signaux du système bouclé sont bornés.

5. Résultats de simulation

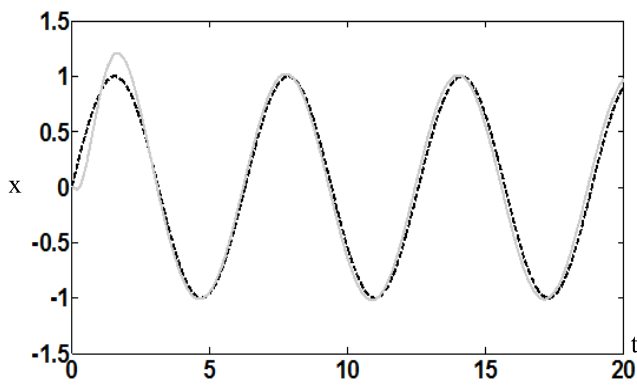
Afin de monter les performances de schéma de commande proposé, des simulations sont effectuées sur le modèle dynamique du quadrotor, dont les paramètres de ce dernier et de contrôleur sont donnés par les tableaux (3.1 et 3.2).

Tableau 3.1. Paramètres du quadrotor

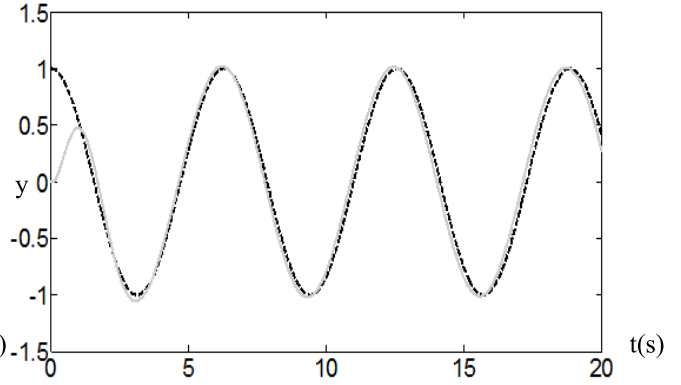
| Symbol | Valeur | Symbol | Valeur |
|---------------------------------|----------------|-------------------------------------|----------------|
| m | 0.46 (Kg) | Iz | 7.65e-3 (KgM2) |
| l | 0.25 (m) | g | 9.81 (Ms2) |
| k _d | 3.23e-7 (NMs2) | k _{ftx} , k _{fty} | 5.56e-4 (Ns/M) |
| k _p | 2.98e-5 (Ns2) | k _{ftz} | 6.35e-4 (Ns/M) |
| Jr | 2.83e-5 (KgM2) | k _{frφ} , k _{frθ} | 5.56e-4 (Ns/r) |
| I _x , I _y | 3.82e-3 (KgM2) | k _{frψ} | 6.35e-4 (Ns/r) |

Tableau 3.2. Paramètres Contrôleur.

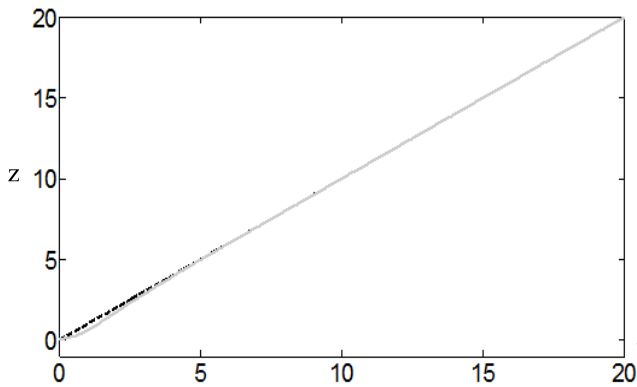
| Symbol | Valeur | Symbol | Valeur |
|----------------|----------------|-----------------|----------------|
| λ_η | diag(50,50,5) | λ_ζ | diag(1,10,10) |
| K_η | diag(50,50,10) | K_ζ | diag(10,10,10) |
| $F_{F\eta}$ | diag(50,50,10) | $F_{F\zeta}$ | diag(10,10,10) |
| F_v | diag(50,50) | F_m | 10 |



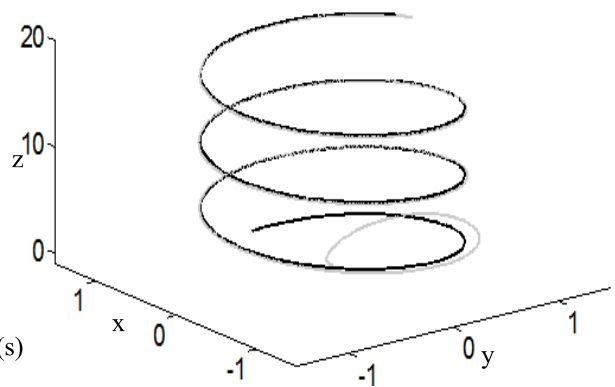
(a)



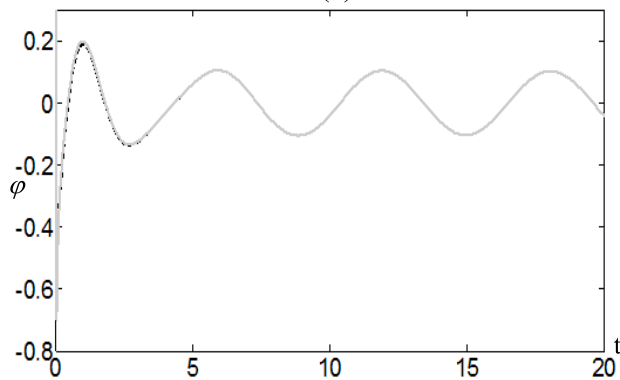
(b)



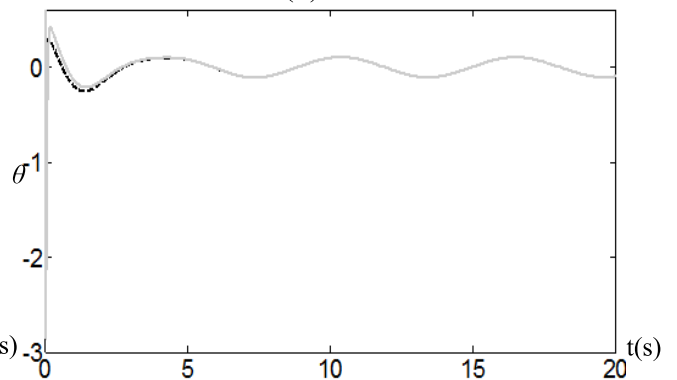
(c)



(d)



(e)



(f)

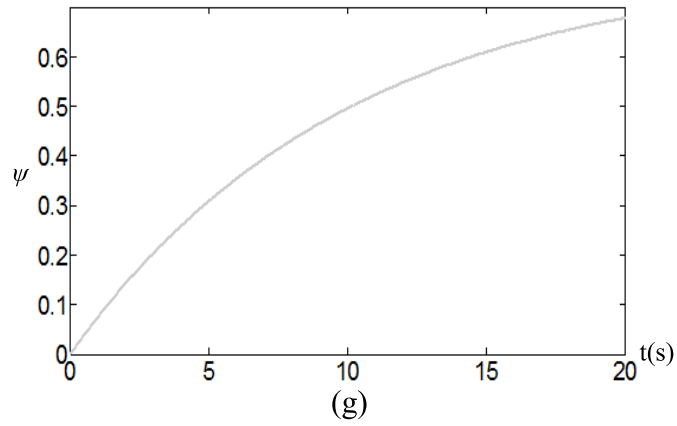
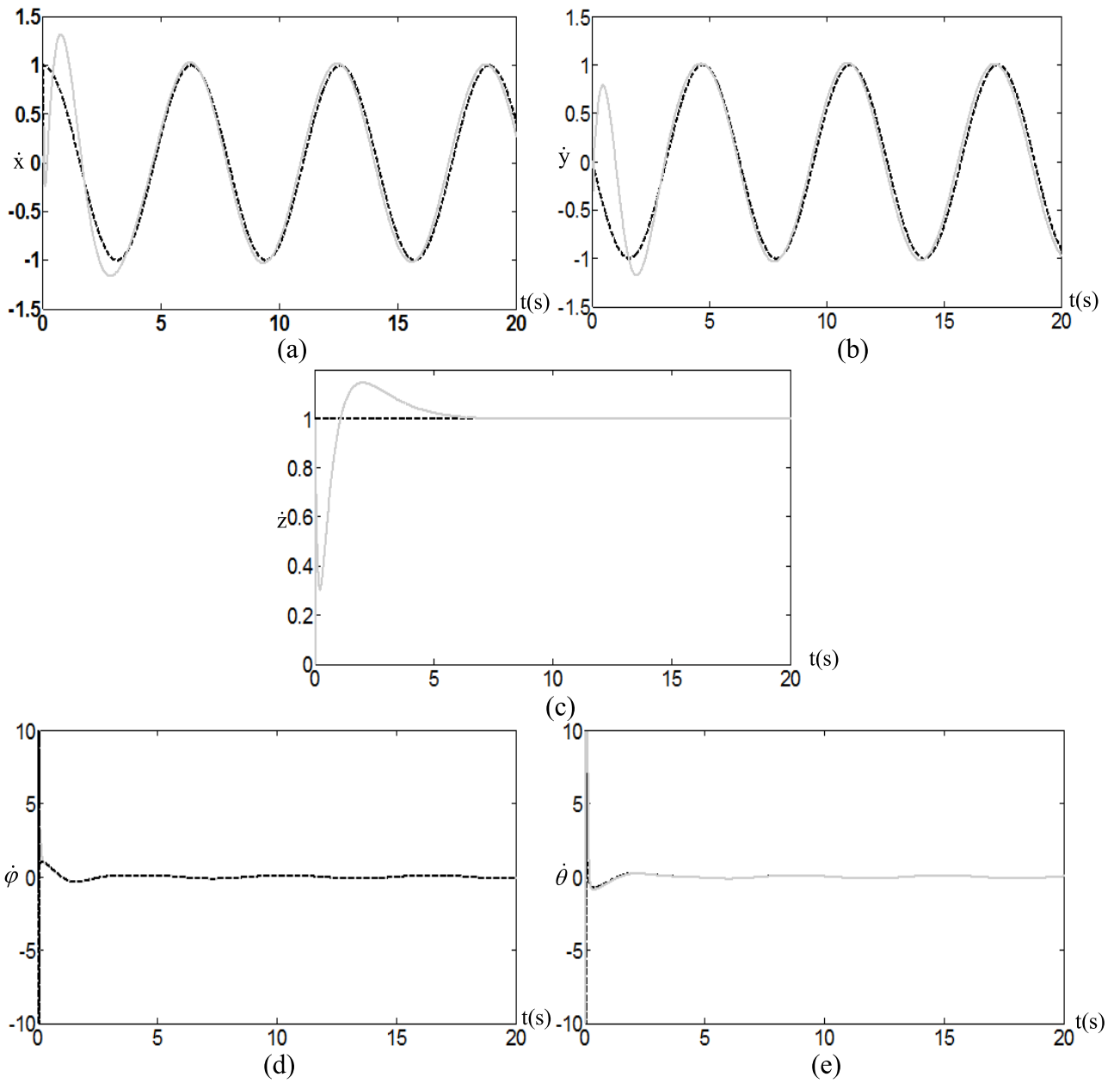


Figure 3.6. Position (- - - :trajectoire désirée, — :trajectoire réelle).



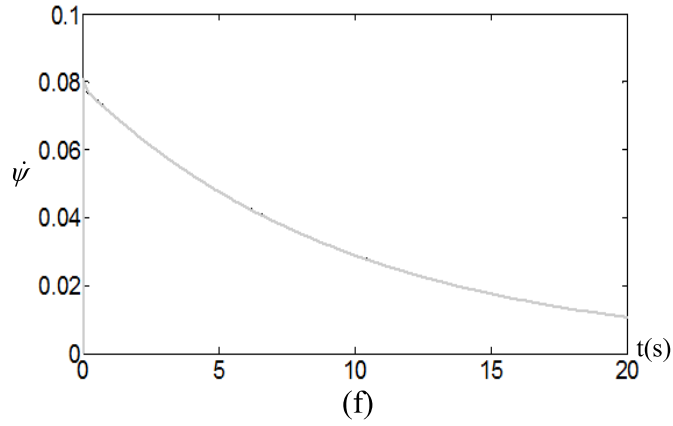
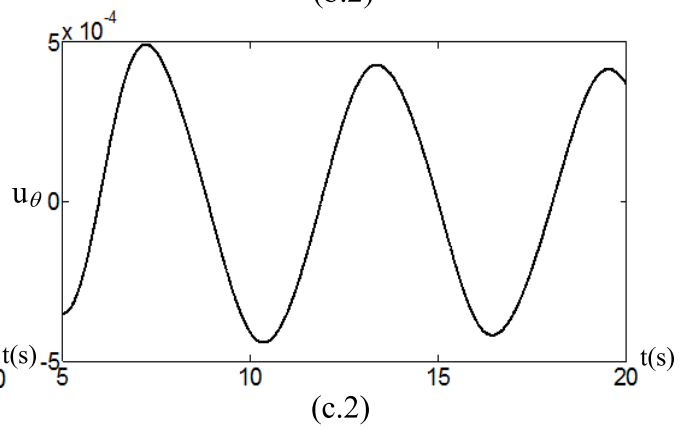
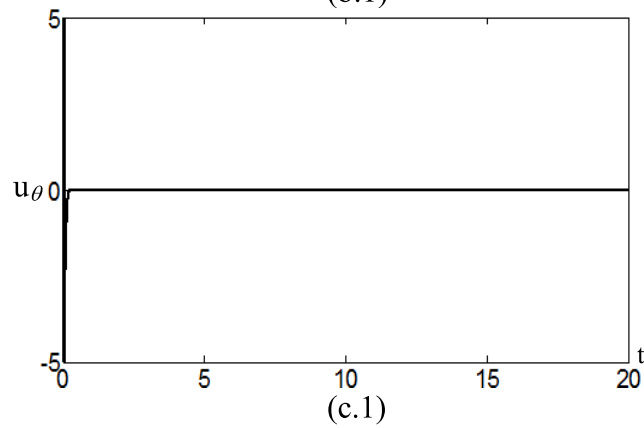
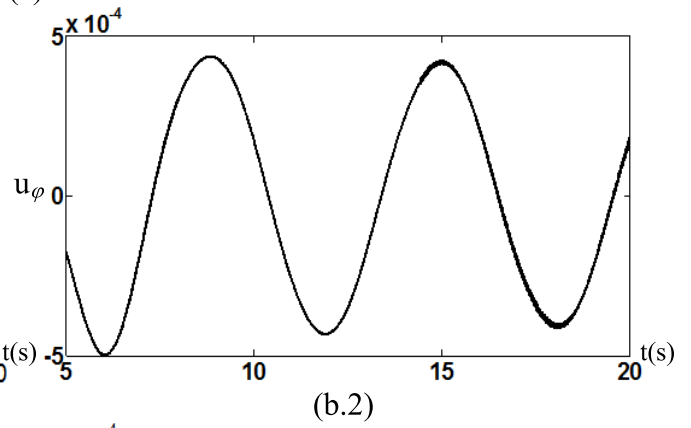
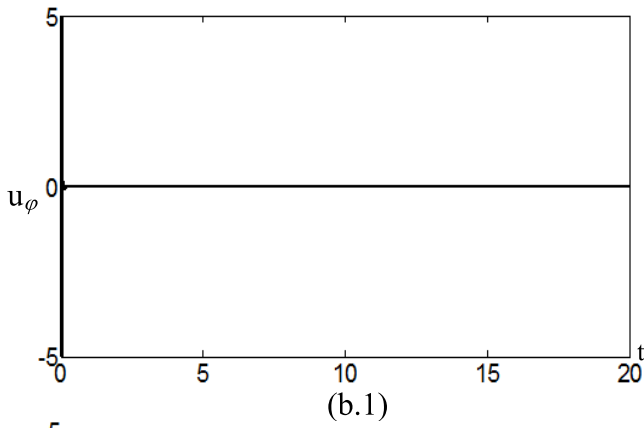
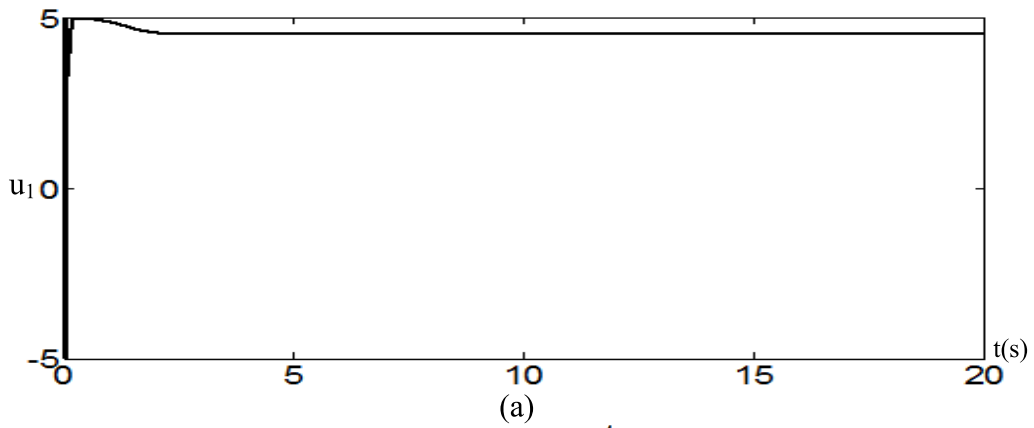


Figure 3.7. Vitesse (- - - :trajectoire désirée, — :trajectoire réelle).



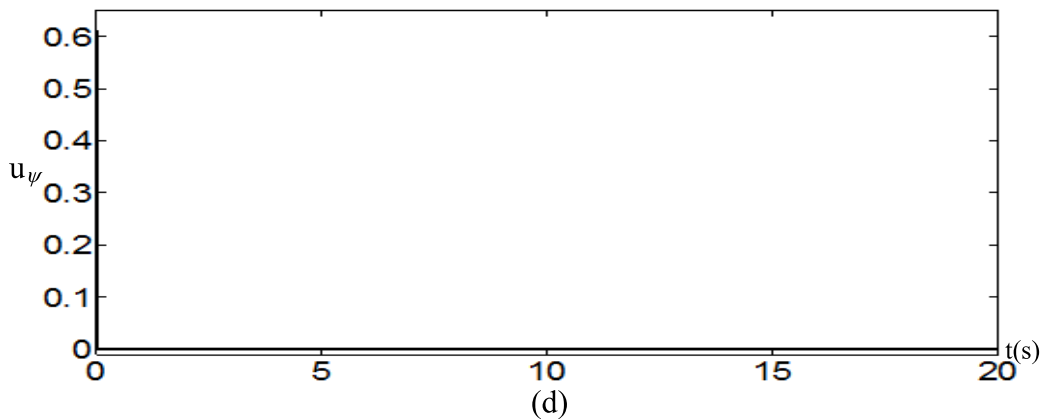


Figure 3.8. Signal de commande.

Les résultats de simulation (Figures. 3.6, 3.7 et 3.8) montrent l'efficacité de la technique proposée. Toutes les sorties de quadrotor ont été commandées par les quatre entrées seulement ce qui implique l'efficacité des contraintes holonomes proposées dans la commande de système sous actionnée. Ceci est dû à l'obtention d'un système équivalent possédant six entrées et six sorties.

6. Conclusion

Une technique de commande adaptative indirecte pour le système sous-actionné «quadrotor» est proposée dans ce chapitre. La méthode proposée comporte trois phases importantes, dans la première des commandes virtuelles ont été utilisées pour rendre le gain de commande inversible. La deuxième phase consiste l'estimation en ligne des fonctions non linéaires inconnues dans le modèle dynamique de quadrotor et les commandes virtuelles ajoutées par un mécanisme d'adaptation. Dans la dernière phase nous avons proposé des contraintes holonomes pour commander les six sorties par les quatre entrées seulement. Ces contraintes ont été déterminées à partir des liens physiques entre les différents sorties du quadrotor, et leurs formules ont été obtenues utilisant la technique de linéarisation entrée-sortie. La théorie de Lyapunov a été utilisée pour démontrer la stabilité du système bouclé. Des simulations numériques ont été effectuées à la fin de ce chapitre afin de monter les performances et l'efficacité de la technique proposée dans la commande des systèmes sous actionnés.

Dans le chapitre suivant, un schéma de commande adaptative basé sur les approximateurs universels, Réseaux de Neurones Artificiels RNA avec observateur d'état est proposé.

Chapitre IV

Commande adaptative par les réseaux de neurones artificiels utilisant un observateur adaptatif neuronal

1. Introduction

Une loi de commande dépend souvent de toutes les mesures essentiels telles que : position, orientation, vitesse, accélération ...etc. Autrement dit, la conception des lois de commande nécessite la disponibilité de toutes les variables d'état à chaque instant. Plusieurs capteurs ont été développés pour les mesures tridimensionnelles tels que : les capteurs ultrasons [Zem06] et l'Unité de Mesure Inertielle (UMI) [Cha10, Che11]. Le capteur ultrasons est composé d'un émetteur et un récepteur utilisé pour mesurer la distance entre le processus et les obstacles. L'inconvénient de ce capteur, est que l'onde émet peut tomber sur plusieurs objets à distances différentes au même temps. L'Unité de Mesure Inertielle (UMI) est un capteur petit, légère, puissant et à coût faible [Cha10] qui donne des mesures sur l'accélération linéaire et la vitesse angulaire en temps réel [Cha10, Che11]. Ce capteur regroupe un gyroscope et un accéléromètre à trois axes [Bat11, Cha10, Che11, Lan13, Lim12, Sti09, Zha11]. Néanmoins, le résultat des UMIs est couplé par des bruits de mesure [Cha10, Che11]. Dont, la vitesse linéaire est obtenue par l'intégration de la mesure de l'accéléromètre et la position est déterminée par une deuxième intégration, la mesure de gyroscope est obtenue aussi par intégration. Toutefois, l'intégration numérique produit des erreurs importantes notamment pour les longues durées. Ces erreurs sont dues principalement à l'existence des bruits de mesure provoqués par les vibrations des moteurs ou d'autres perturbations [Che11, Mat92, Mar10, Nag12]. Alors, la mesure des UMIs ne peut pas être utilisée directement pour la mise en œuvre de la loi de commande.

L'observateur est un logiciel qui peut reconstruire les variables d'états en utilisant les informations entrées/ sorties seulement. Plusieurs types d'observateur ont été développées à

savoir : observateur à grand gain, observateur à mode de glissement, observateur à base des systèmes flous et réseaux de neurones artificiels. Dans [Die08, Die09], un observateur basé sur ce dernier a été utilisé pour estimer les vitesses linéaires et angulaires d'un quadrotor. Le terme de correction dans ces travaux est composé de deux parties : la première est proportionnelle à l'erreur d'observation et la deuxième est l'estimée par le RNA des termes inconnus dans le modèle d'état. L'observation des angles d'Euler et des vitesses angulaire du même système a été effectuée par un observateur qui utilise le modèle flou Takagi – Sugeno (TS) dans [Len11]. Le filtre de kalman étendu a été largement implémenté pour le quadrotor tels que : [Ale12, Kam13, Lan13, Tan13, Zha11]. Dans [Gui09] et [Ale12], un observateur à grand gain (OGG) et à mode glissant (OMG) ont été utilisés respectivement pour estimer l'orientation et toutes les vitesses linéaires et angulaires. L'estimation des variables d'états et des perturbations externes dans [Ben08], a été effectuée par un observateur à mode glissant d'ordre supérieur. Ces derniers, (c'est-à-dire OMG et OGG) sont très robuste vis-à-vis les perturbations et les incertitudes [Lee07]. Cependant, l'inconvénient major de ces observateurs est leurs sensibilités aux bruits de mesure. Ce phénomène indésirable est causé principalement par la fonction signe et le gain d'observation si l'observateur à mode glissant est utilisé. Alors, des solutions ont été proposées telles que : l'implémentation d'un gain d'observation adaptatif [Che06, Che09] et l'utilisation des fonctions approximatives et lissent à la fonction signe [Kim11, Slo91] telles que : sigmoïde, saturation, tangent hyperbolique...etc. Néanmoins, L'OMG perd ses propriétés de convergences. Dans [Haj10, Vel11], un OMG et un OGG sont utilisés pour séparer les signaux rapides et les signaux lents.

Dans ce chapitre, un schéma de commande adaptative par les réseaux de neurones artificiels utilisant un nouvel observateur adaptatif neuronal est proposé pour une classe des systèmes non linéaires incertains. Le schéma proposé est appliqué sur le modèle dynamique d'un quadrotor. Ce schéma est composé de douze RNAs, six RNA pour le contrôleur adaptatif et six RNA pour l'observateur. L'objectif principal est de synthétiser un observateur insensible aux bruits de mesure. Le terme σ -modification est additionné à toutes les lois d'adaptation, et la méthode directe de Lyapunov est utilisée pour étudier la stabilité du système global et assurer la bornitude de tous les signaux en boucle fermée.

Ce chapitre est organisé comme suit : La section 2 présente la position de problème. La section 3 est divisée en quatre parties : la synthèse d'une loi de commande adaptative par le réseau de neurones artificiel, la structure conventionnelle d'un observateur à mode glissant, la conception de l'observateur adaptatif neuronal et l'étude de stabilité du système global. Les

résultats de simulation avec une étude comparative sont exposés dans la section 4, dont, l'application a été effectuée sur le modèle de quadrotor en utilisant des commandes virtuelles et de nouvelles contraintes holonomes. La conclusion est donnée dans la section 5.

2. Position du problème

Le système étudié dans ce chapitre est non linéaire, MIMO (Multi Input Multi Output), et affine à la commande, son équation différentielle est donnée comme suit :

$$Y^{(r)} = F(X) + G(X)U \quad (4.1)$$

Ce système est composé de p sous-systèmes $(\Sigma_i, i = \overline{1:p})$, dont, chacun a la forme :

$$\Sigma_i = \dot{Y}_i^{(r_i)} = f_i(X) + g_i(X)u_i, \quad i = \overline{1:p} \quad (4.2)$$

Ou bien :

$$\Sigma_i = \begin{cases} \dot{x}_{ij} = x_{ij+1}, \\ \dot{x}_{ir_i} = f_i(X) + g_i(X)u_i, \quad i = \overline{1:p}, \quad j = \overline{1:r_i-1} \\ Y_i = x_{i,1} + b_i \end{cases} \quad (4.3)$$

Avec p est le nombre des sous-systèmes ,

$$X = [Y_1 \quad \dots \quad Y_1^{(r_1-1)} \quad \dots \quad Y_p \quad \dots \quad Y_p^{(r_p-1)}]^T = [x_{11} \quad \dots \quad x_{1r_1} \quad \dots \quad x_{p1} \quad \dots \quad x_{pr_p}]^T \in \mathcal{R}^r$$

: est le vecteur d'état, $U = [u_1 \quad \dots \quad u_p]^T$: est le vecteur d'entrée, $Y = [Y_1 \quad \dots \quad Y_p]^T$: est le

vecteur de sortie, r_i : est le degré relatif de chaque sous-système i , les fonctions $f_i(X)$, $g_i(X)$ sont non linéaires, lissent et inconnues et b_i : présente le bruit de mesure qu'est supposé un bruit blanc gaussien [Abe11, Ale12, Bat11, Din13, Mok05, Rub13, Zha11]

Avant de commencer la synthèse du schéma de commande proposé, des hypothèses sont prises en considération dans ce chapitre :

- **Hypothèses 4.1:** le vecteur de sortie Y est mesurable.
- **Hypothèses 4.2:** la trajectoire désirée $Y_{di}, i = \overline{1:p}$ et ces dérivées sont lissent et bornées.
- **Hypothèses 4.3:** le gain de la commande $g_i(X)$ est positif, borné ($g_i(X) \leq \bar{g}, i = \overline{1:p}$) et lentement variable dans le temps.

L'objectif principal est de synthétiser une loi de commande U qui garantit la convergence de la trajectoire réelle Y vers la trajectoire désirée Y_d et la bornitude de tous les signaux en boucle fermée.

Pour synthétiser la loi de commande qui garantit cet objectif, l'erreur de poursuite pour chaque sous-système e_i sera définie par :

$$e_i = Y_{d_i} - Y_i, \quad i = \overline{1:p} \quad (4.4)$$

Et l'erreur filtrée par:

$$s_i = \left(\frac{d}{dt} + \lambda_i \right)^{r_i - 1} e_i = \Phi_i E_i, \quad \lambda_i > 0, \quad i = \overline{1:p} \quad (4.5)$$

Avec λ_i est un paramètre de conception, E_i est un vecteur d'état, regroupant l'erreur de poursuite et ses (r_i-1) dérivées :

$$E_i = \left[e_i \quad \dot{e}_i \quad \dots \quad e_i^{(r_i-1)} \right]^T \in \mathbb{R}^{r_i}, \quad i = \overline{1:p} \quad (4.6)$$

Et Φ_i est donné par :

$$\Phi_i = [v_{i,1} \quad \dots \quad v_{i,r_i-1} \quad 1] \quad i = \overline{1:p} \quad (4.7)$$

Selon l'expression (4.5), on déduit que la convergence de l'erreur de poursuite e_i et toutes ses dérivées $e_i^{(j)}$, $j = \overline{1:r_i-1}$ vers zéros est assurée si l'erreur filtrée s_i est nulle [Slo91]. De plus, la bornitude de cette dernière implique la bornitude de l'erreur de poursuite et ses dérivées [Slo91]. Alors, la conception d'une loi de commande assurant une erreur filtrée nulle est l'objectif principal.

Pour chaque sous-système i , la loi de commande garantissant ce but est formulée comme suit :

$$u_i^* = u_i^{eq^*} + u_i^{pd^*}, \quad i = \overline{1:p} \quad (4.8)$$

Avec: u_i^* est la loi de commande idéale, elle est composée de deux termes, le premier $u_i^{eq^*}$ présente la commande équivalente idéale définie comme suit :

$$u_i^{eq^*} = g_i^{-1}(X)(v_i - f_i(X)) \quad (4.9)$$

et le deuxième est la commande PD idéale donnée par :

$$u_i^{pd^*} = k_i s_i = k_i \Phi_i E_i, \quad k_i > 0 \quad (4.10)$$

Afin de montrer que la loi de commande idéale admet une erreur filtrée nulle et par conséquent, la convergence de l'erreur de poursuite e_i et ses dérivées vers zéro, étudiant la dynamique de l'erreur filtrée :

$$\dot{s}_i = v_i - f_i(X) - g_i(X)u_i = e_i^{(r_i)} + \Phi_{0i} E_i \quad i = \overline{1:p} \quad (4.11)$$

Avec :

$$\begin{cases} v_i = Y_{di}^{(r_i)} + \sum_{j=1}^{r_i-1} v_{ij} e_i^{(j)} \\ v_{ij} = \frac{(r_i-1)!}{(r_i-i)!(j-1)!} \lambda^{r_i-j} \end{cases} \quad i = \overline{1:p}, j = \overline{1:r_i-1} \quad (4.12)$$

Et le vecteur Φ_{0i} est définie par :

$$\Phi_{0i} = [0 \quad v_{i,1} \quad \cdots \quad v_{i,r_i-1}], \quad i = \overline{1:p} \quad (4.13)$$

En remplaçant la loi de commande idéale donnée par (4.8) dans (4.11), on obtient:

$$\dot{s}_i = -g_i(X)k_i s_i, \quad \Rightarrow s_i \dot{s}_i \leq 0, \quad i = \overline{1:p} \quad (4.14)$$

Ce qui implique $s_i \rightarrow 0$, quand $t \rightarrow \infty$, et par conséquent $e_i^{(j)} \rightarrow 0$ ($j = \overline{0:r_i-1}, i = \overline{1:p}$), quand $t \rightarrow \infty$ [Slo91].

Selon l'analyse ci-dessus, la loi de commande (4.8) est facilement obtenue si les fonctions non linéaire $f_i(X)$ et $g_i(X)$ sont connues et le vecteur d'état X est disponible pour la mesure. Mais, ces fonctions sont généralement totalement ou partiellement inconnues, et seules les grandeurs de sorties qui sont mesurables. Alors, la loi de commande idéale u_i^* doit être approximée par une autre basée principalement sur l'implémentation de l'approximateur universel RNA et un observateur d'état est proposé pour reconstruire le vecteur d'état.

3. Synthèse d'une commande adaptative neuronale avec observateur adaptatif neuronal

Cette section présente la conception proposée pour la loi de commande, un réseau de neurones artificiel avec une seule couche cachée [Sar06] est utilisé pour l'approximation de la commande équivalente définie dans (4.9). Un observateur adaptatif neuronal est proposé pour l'estimation du vecteur d'état E_i .

L'estimation de la commande idéale (4.8) est donnée comme suit :

$$u_i = u_i^{eq} + u_i^{pd} + u_i^r \quad (4.15)$$

Avec : u_i est la loi de commande appliquée au système, elle est composée de trois termes comme suit :

- u_i^{eq} : est la commande équivalente estimée par un réseau de neurones artificiel. Le détail de l'approximation par le RNA est présenté plus tard.
- u_i^{pd} : est la commande PD utilisée pour la stabilisation du système. Son expression est définie comme suit :

$$u_i^{pd} = k_i \hat{s}_i = k_i \Phi_i \hat{E}_i. \quad (4.16)$$

Avec \hat{E}_i est l'estimé du E_i par un observateur d'état et \hat{s}_i est l'estimé de s_i dont :

$$\hat{s}_i = \Phi_i \hat{E}_i. \quad (4.17)$$

- u_i^r : est un terme robuste de type mode glissant [Kha96, Slo91]. Ce terme est utilisé pour compenser les erreurs d'approximation, dont :

$$u_i^r = \hat{\rho}_i \text{sign}(\hat{s}_i) \quad (4.18)$$

Le paramètre $\hat{\rho}_i \in \mathfrak{R}$ est variable dans le temps et sa loi d'adaptation est donnée par :

$$\dot{\hat{\rho}}_i = \gamma_i |\hat{s}_i|, \quad \gamma_i > 0. \quad (4.19)$$

3.1. Commande adaptative neuronale

Selon le théorème d'approximation universelle [Sar06], il existe des poids idéaux W_i et V_i tel que le RNA avec une seule couche cachée (Figure 4.1), peut approximer la commande équivalente idéale dans (4.9), d'où :

$$u_i^{eq*} = W_i^T \sigma(V_i^T \chi_i) + \varepsilon_i \quad (4.20)$$

Avec: $\chi_i = \begin{bmatrix} Y_{di}^{(r)} & Y_{di}^{(r-1)} & \dots & Y_{di} & e_i^{(r-1)} & e_i^{(r-2)} & \dots & e_i \end{bmatrix}^T \in \mathfrak{R}^{n_{ei}}$: est le vecteur d'entrée de RNA , $\sigma()$: est la fonction d'activation, $W_i \in \mathfrak{R}^{n_{ci} \times n_{si}}$: sont les poids idéaux des couches cachées/sortie et $V_i \in \mathfrak{R}^{n_{ei} \times n_{ci}}$: sont les poids idéaux des couches entrée/cachée, ε_i : est l'erreur d'approximation idéale par le RNA dans la Figure 4.1, n_{ei} : est le nombre de neurones dans la couche d'entrée, n_{ci} : est le nombre de neurones dans la couche cachée, et n_{si} : est le nombre de neurones de la couche de sortie.

Toutefois, les poids idéaux (W_i et V_i) et l'entrée du RNA χ_i sont inconnus. Afin de remédier ce problème :

- les poids idéaux sont estimés par \hat{W}_i, \hat{V}_i , en utilisant un mécanisme d'adaptation,

- et le vecteur d'entrée χ_i est estimé par $\hat{\chi}_i$ en implémentant un observateur d'état.

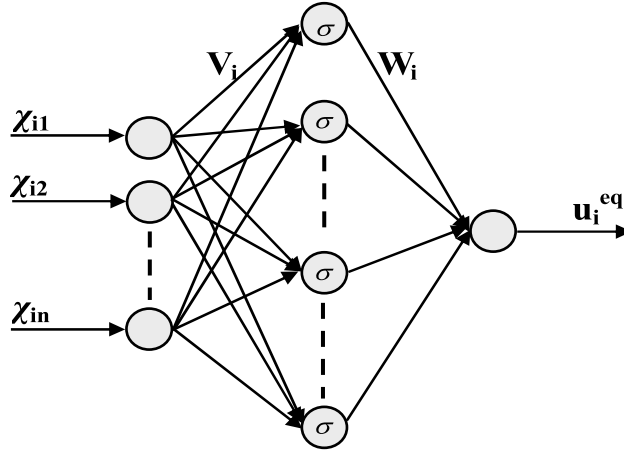


Figure 4.1. Réseau de neurones artificiel à une seule couche cachée.

Après l'utilisation de ces derniers, la loi de commande équivalente appliquée au système a la forme suivante :

$$u_i^{eq} = \hat{W}_i^T \sigma(\hat{V}_i^T \hat{\chi}_i) \quad (4.21)$$

Cette loi est la sortie adaptative du RNA présenté dans la Figure 4.1.

La commande équivalente dans (4.21) peut prendre une autre forme en utilisant le développement en série de Taylor de la fonction d'activation $\sigma(\hat{V}_i^T \hat{\chi}_i)$ comme suit :

$$\sigma(\hat{V}_i^T \hat{\chi}_i) = \sigma(\hat{V}_i^T \hat{\chi}_i) + \sigma'(\hat{V}_i^T \hat{\chi}_i) \tilde{V}_i^T \hat{\chi}_i + O(\tilde{V}_i^T \hat{\chi}_i)^2 \quad (4.22)$$

Tel que :

$$\sigma'(\hat{V}_i^T \hat{\chi}_i) = \frac{d(\sigma(\hat{V}_i^T \hat{\chi}_i))}{d(\hat{V}_i^T \hat{\chi}_i)} \Big|_{\hat{V}_i^T \hat{\chi}_i = \hat{V}_i^T \hat{\chi}_i} \quad (4.23)$$

En utilisant (4.22), la commande équivalente adaptative dans (4.21) devient :

$$u_i^{eq} = \hat{W}_i^T \sigma(\hat{V}_i^T \hat{\chi}_i) - \hat{W}_i^T \sigma'(\hat{V}_i^T \hat{\chi}_i) \tilde{V}_i^T \hat{\chi}_i - \hat{W}_i^T O(\tilde{V}_i^T \hat{\chi}_i)^2 \quad (4.24)$$

Par l'utilisation des formules de (4.20) et (4.24), l'erreur d'approximation de la commande équivalente idéale dans (4.9) par le RNA présenté dans la Figure 4.1 est formulée comme suit :

$$\tilde{u}_i^{eq} = u_i^{eq*} - u_i^{eq} = \tilde{W}_i^T \sigma(\hat{V}_i^T \hat{\chi}_i) + \hat{W}_i^T \sigma'(\hat{V}_i^T \hat{\chi}_i) \tilde{V}_i^T \hat{\chi}_i + w_i \quad (4.25)$$

Dont: $\tilde{W}_i = W_i - \hat{W}_i$, $\tilde{V}_i = V_i - \hat{V}_i$ sont les erreurs d'estimation des poids et w_i est définie par :

$$w_i = W_i^T (\sigma(V_i^T \chi) - \sigma(\hat{V}_i^T \hat{\chi}_i)) + \tilde{W}_i^T \sigma'(\hat{V}_i^T \hat{\chi}_i) \tilde{V}_i^T \hat{\chi}_i + W_i^T O(\tilde{V}_i^T \hat{\chi}_i)^2 + \varepsilon_i \quad (4.26)$$

• **Hypothèse 4.4** : les poids idéaux W_i et V_i et l'erreur d'estimation w_i sont bornés, c'est-à-dire il existe des constantes positives inconnues $W_{i,m}$, $V_{i,m}$ et ρ_i , tel que : $\|W_i\|_F \leq W_{i,m}$, $\|V_i\|_F \leq V_{i,m}$, $\|w_i\| \leq \rho_i$, $\|V_i\|_F \leq V_{i,m}$ et $\|w_i\| \leq \rho_i$.

Pour atteindre l'objectif de la commande le mécanisme d'adaptation implémenté pour l'adaptation des poids du RNA est donnée par [Bou14]:

$$\begin{cases} \dot{\hat{W}}_i = F_{W_i} \left(\sigma(\hat{V}_i^T \hat{\chi}_i) \hat{s}_i - \kappa_i \hat{W}_i \right) \\ \dot{\hat{V}}_i = F_{V_i} \left(\hat{\chi}_i \hat{s}_i \hat{W}_i^T \sigma'(\hat{V}_i^T \hat{\chi}_i) - \kappa_i \hat{V}_i \right) \end{cases} \quad (4.27)$$

Avec: $F_{W_i} > 0$, $F_{V_i} > 0$ sont les gains d'adaptation et κ_i est un paramètre constant, positif et petit. Ce dernier exprime le terme σ -modification [Bou12, Nag13, Spo06] pour éviter la divergence des poids estimés [Coz06, Nic08] ce qui implique la divergence des erreurs d'estimation \tilde{W}_i , \tilde{V}_i et par conséquent l'augmentation de la fonction candidate de Lyapunov [Far06] c'est-à-dire que le système en boucle fermée sera instable.

• **Remarque 4.1.** L'explosion des poids arrive si le système est soumis sous l'effet des perturbations non paramétriques pendant des longues durées. Ce problème est remédié par plusieurs types de solutions (Voir chapitre 2). Le schéma de commande proposé dans ce chapitre traite un problème différent, alors, le terme additionnel de la technique σ -modification a été utilisé dans les lois d'adaptation pour éviter l'explosion des poids.

3.2. Observateur à mode glissant

Toutes les lois du vecteur de commandes (équations 4.16, 4.18 et 4.221), les lois d'adaptation (équations 4.19 et 4.27) et l'entrée de RNA $\hat{\chi}_i$ dépendent du vecteur d'état estimé \hat{E}_i .

L'Observateur à Mode Glissant (OMG) [Ben08, Gui09, Haj10, Mok05, Vel11] est présenté dans cette partie. Le but, est de faire une étude comparative entre l'OMG et l'observateur proposé, dont, ce dernier est basé sur la structure d'un OMG.

Selon (4.11), on peut écrire :

$$e_i^{(r_i)} = \dot{s}_i - \Phi_{0i} E_i \quad (4.28)$$

En utilisant la dernière équation, la dynamique de vecteur d'état E_i est donnée par:

$$\begin{cases} \dot{E}_i = (A_i - B_i \Phi_{0i}) E_i + B_i \dot{s}_i \\ e_i = C_i E_i \end{cases} \quad (4.29)$$

Avec :

$$A_i = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \in \mathfrak{R}^{r_i \times r_i}, B_i = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in \mathfrak{R}^{r_i \times 1}, C_i^T = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \in \mathfrak{R}^{r_i \times 1}. \quad (4.30)$$

L'observateur candidat pour le système d'équation (4.29), est défini comme suit [Bou14] :

$$\begin{cases} \dot{\hat{E}}_i = (A_i - B_i \Phi_{0i}) \hat{E}_i + O_i(\tilde{e}_i) \\ \hat{e}_i = C_i \hat{E}_i \end{cases} \quad (4.31)$$

Avec $\hat{E}_i = [\hat{e}_i \quad \dot{\hat{e}}_i \quad \dots \quad \hat{e}_i^{(r_i-1)}]^T$ est l'estimé de E_i , par l'observateur ci-dessus, le terme \tilde{e}_i représente l'erreur d'observation, elle est définie comme la différence entre l'erreur de poursuite mesurée et celle estimée par l'observateur (4.31) ($\tilde{e}_i = e_i - \hat{e}_i$) et la fonction $O_i(\tilde{e}_i)$ est le terme de correction, utilisé pour compenser les non linéarités et les termes inconnues dans le modèle d'état.

Le choix du terme de correction $O_i(\tilde{e}_i)$ définit le type de l'observateur utilisé, pour un observateur à mode glissant, le terme $O_i(\tilde{e}_i)$ est donné par: [Bou14, Shi06, Zha10]

$$O_i(\tilde{e}_i) = \delta_i \Delta_i \text{signe}(\tilde{e}_i) \quad (4.32)$$

Avec \tilde{e}_i est la surface de glissement, $\delta_i \gg 1$ est le gain d'observation, $\mathcal{G}_{i,j} > 0$ ($i = \overline{1:p}, j = \overline{1:r_i}$) sont des paramètres de conception et Δ_i est donné par l'expression suivante :

$$\Delta_i = [\mathcal{G}_{i,1} \quad \mathcal{G}_{i,2} \delta_i \quad \dots \quad \mathcal{G}_{i,r_i} \delta_i^{r_i-1}]^T \quad (4.33)$$

• **Remarque 4.2.** on peut trouver d'autre types des observateurs à mode glissant (OMGs), dont, la fonction signe dans le terme de correction (équation (4.32)) est remplacée par d'autre fonctions lisses telles que : sat, tanh, sinh,...etc.

La robustesse de cet observateur est due à deux facteurs sont le gain d'observation δ_i et la fonction signe dans le terme de correction (équation (4.32)). Cependant ces derniers provoquent

un phénomène indésirable qui est : l'amplification des bruits de mesures. Il est à noter que ce phénomène apparait même si on approxime la fonction signe par une fonction lisse.

3.3. Conception d'un observateur neuronal adaptatif

Dans cette partie, un observateur neuronal adaptatif est proposé pour l'estimation du vecteur d'état E_i . L'objectif principal de cet observateur est de minimiser la sensibilité aux bruits de mesure.

En se basant sur l'expression donnée par (4.31), l'observateur neuronal idéal insensible aux bruits de mesure a l'expression suivante :

$$\begin{cases} \dot{\hat{E}}_i^* = (A_i - B_i \Phi_{0i}) \hat{E}_i^* + O_i^* (\tilde{e}_i^*) \\ \hat{e}_i^* = C_i \hat{E}_i^* \end{cases} \quad (4.34)$$

Avec : \hat{E}_i^* : est l'estimation idéale de E_i par l'observateur ci-dessus, $\tilde{e}_i^* = e_i - \hat{e}_i^*$: est l'erreur d'observation mesurée et $O_i^* (\tilde{e}_i^*)$ est le terme de correction idéal défini par :

$$O_i^* (\tilde{e}_i^*) = \Delta_i F^* (\tilde{e}_i^*) \quad (4.35)$$

La fonction $F^* (\tilde{e}_i)$ présente la sortie idéale d'un réseau de neurone composé d'un neurone de sortie, un neurone caché et deux neurones d'entrée (voir Figure 4.2), dont sa sortie est donnée par :

$$F^* (\tilde{e}_i) = \mu_i T(A_i^T \bar{E}_i^*) + \varepsilon_{i,0} \quad (4.36)$$

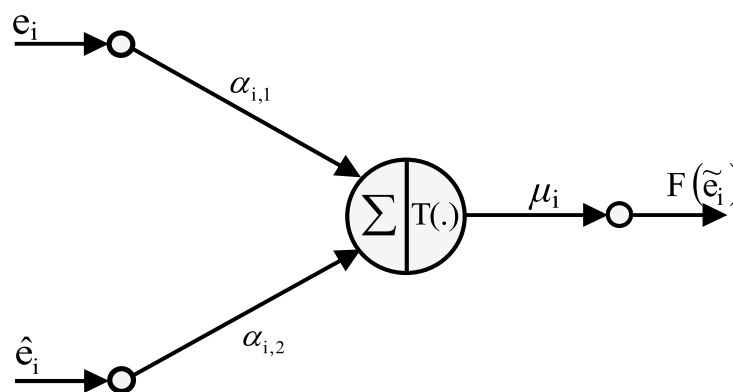


Figure 4.2. RNA utilisé dans l'observateur proposé.

Avec : $\bar{E}_i^* = [e_i \quad \hat{e}_i^*]^T$: est le vecteur d'entrée idéal, $A_i = [\alpha_{i,1} \quad \alpha_{i,2}]^T$: est le vecteur des poids idéaux des couches cachées/sortie, μ_i : est le poids idéal des couches entrée/cachée et $\varepsilon_{i,0}$:

est l'erreur d'approximation idéale par le RNA dans la figure 4.2 et $T(\tilde{e}_i^*)$ est la fonction d'activation choisi comme suit [Bou14] :

$$T(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}} \quad (4.37)$$

Les poids A_i et μ_i (et par conséquent le vecteur $\bar{E}_i^* = [e_i \ \hat{e}_i^*]^T$ est inconnu) sont inconnus ce qui implique que l'implémentation de l'observateur (4.34) est impossible. Alors, les poids idéaux A_i et μ_i sont approximés par \hat{A}_i et $\hat{\mu}_i$ en utilisant un mécanisme d'adaptation et l'observateur adaptatif neuronal proposé est défini par :

$$\begin{cases} \dot{\hat{E}}_i = (A_i - B_i \Phi_{0i}) \hat{E}_i + \Delta_i F(\tilde{e}_i) \\ \hat{e}_i = C_i \hat{E}_i \end{cases} \quad (4.38)$$

Avec la fonction $F(\tilde{e}_i)$ est l'approximée adaptative de la fonction $F^*(\tilde{e}_i^*)$ (équation (4.37)) par le RNA présenté dans la Figure 4.2., avec :

$$F(\tilde{e}_i) = \hat{\mu}_i T(\hat{A}_i^T \bar{E}_i) = \hat{\mu}_i T(A_i^T \bar{E}_i) - \hat{\mu}_i T'(\hat{A}_i^T \bar{E}_i) \tilde{A}_i^T \bar{E}_i - \hat{\mu}_i O(\tilde{A}_i^T \bar{E}_i)^2 \quad (4.39)$$

La deuxième expression est obtenue par le développement en série de Taylor de la fonction d'activation $T(\hat{A}_i^T \bar{E}_i)$ d'où :

$$T(A_i^T \bar{E}_i) = T(\hat{A}_i^T \bar{E}_i) + T'(\hat{A}_i^T \bar{E}_i) \tilde{A}_i^T \bar{E}_i + O(\tilde{A}_i^T \bar{E}_i)^2 \quad (4.40)$$

Avec :

$$T'(A_i^T \bar{E}_i) = \left. \frac{d(T'(\hat{A}_i^T \bar{E}_i))}{d(A_i^T \bar{E}_i)} \right|_{A_i^T \bar{E}_i = \hat{A}_i^T \bar{E}_i} \quad (4.41)$$

Le vecteur d'état $\tilde{E}_i = E_i - \hat{E}_i$ est défini comme étant le vecteur des erreurs d'observation, sa dynamique est donnée par :

$$\begin{cases} \dot{\tilde{E}}_i = (A_i - B_i \Phi_{0i}) \tilde{E}_i + B_i \dot{s}_i - \Delta_i F(\tilde{e}_i) \\ \tilde{e}_i = C_i \tilde{E}_i \end{cases} \quad (4.42)$$

Avec : $\tilde{\mu}_i = \mu_i - \hat{\mu}_i$, $\tilde{A}_i = A_i - \hat{A}_i$ sont les erreurs d'approximation neuronale, et l'expression de $\tilde{F}(\tilde{e}_i)$ est comme suit :

$$\tilde{F}(\tilde{e}_i) = F^*(\tilde{e}_i^*) - F(\tilde{e}_i) = \tilde{\mu}_i T(\hat{A}_i^T \bar{E}_i) + \hat{\mu}_i T'(\hat{A}_i^T \bar{E}_i) \tilde{A}_i^T \bar{E}_i + w_{i,o} \quad (4.43)$$

On a aussi :

$$\tilde{w}_{i,0} = \mu_i^T \left\{ T(A_i^T \bar{E}_i^*) - T(A_i^T \bar{E}_i) \right\} + \mu_i O(\tilde{A}_i^T \bar{E}_i)^2 + \tilde{\mu}_i^T T'(\hat{A}_i^T \bar{E}_i) \tilde{A}_i^T \bar{E}_i + \varepsilon_{i,0} \quad (4.44)$$

• **Hypothèse 4.5.** $\|A_i\| \leq A_{i,m}, |\mu_i| \leq \mu_{i,m}, \|\tilde{w}_{i,0}\| \leq \bar{w}_{i,0}, |\mu_i| \leq \mu_{i,m}$, et $\|\tilde{w}_{i,0}\| \leq \bar{w}_{i,0}$ avec $A_{i,m}, \mu_{i,m}$ et $\bar{w}_{i,0}$ sont les constantes positives inconnues.

Pour la mise en œuvre de l'observateur neuronal adaptatif proposé, les lois d'adaptation des poids estimés $\hat{\mu}_i, \hat{A}_i$ sont données par [Boul1b, Boul14]:

$$\begin{cases} \dot{\hat{\mu}}_i = F_{\mu_i} \left(T(\hat{A}_i^T \bar{E}_i) \tilde{e}_i + \kappa_{\mu_i} (\bar{\mu}_i - \hat{\mu}_i) \right) \\ \dot{\hat{A}}_i = F_{A_i} \left(\bar{E}_i \tilde{e}_i \hat{\mu}_i^T T'(\hat{A}_i^T \bar{E}_i) + \kappa_{A_i} (\bar{A}_i - \hat{A}_i) \right) \end{cases} \quad (4.45)$$

Avec : $F_{\mu_i} > 0, F_{A_i} > 0, \kappa_{\mu_i} > 0$, et $\kappa_{A_i} > 0$ sont des gains d'adaptation et $\bar{\mu}_i, \bar{A}_i$ sont choisis par le concepteur.

Le schéma de commande adaptative neuronale basée sur l'observateur adaptatif neuronal pour un système SISO (Single Input Single Output) de deuxième ordre est représenté sur la Figure 4.3.

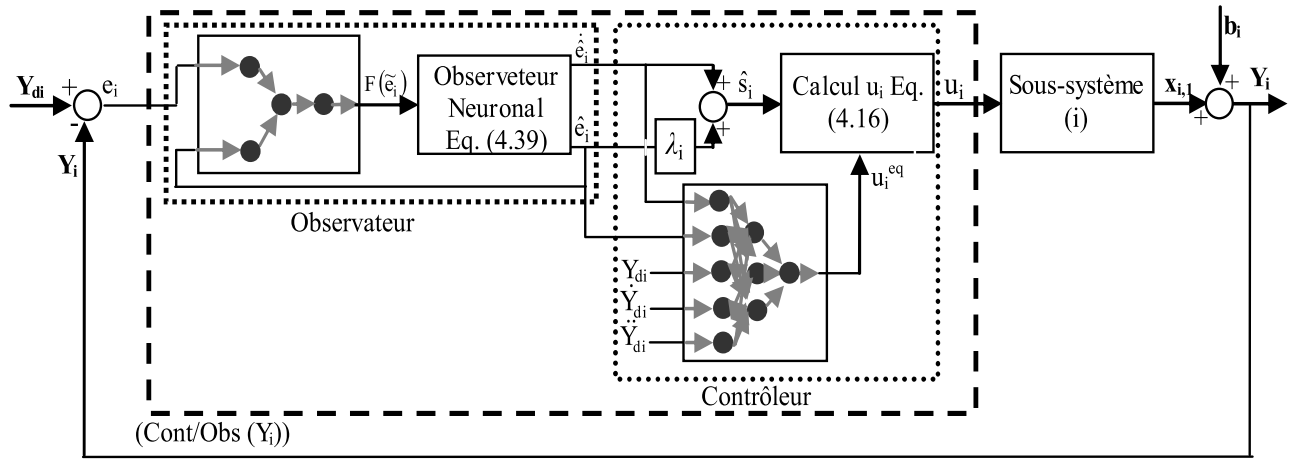


Figure 4.3. Configuration contrôleur/ observateur pour un système SISO de deuxième ordre.

3.4. Analyse de la stabilité

Afin d'examiner la stabilité du système global (contrôleur adaptatif / observateur proposé), et la bornitude de tous les signaux, la méthode directe de Lyapunov est utilisée. Pour cela, on utilise :

i). Soit $Q_i = Q_i^T > 0$ la solution algébrique unique de l'équation de Lyapunov définie par l'équation ci-dessous [Far09]:

$$Q_i + A_i^T Q_i + Q_i A_i = C_i^T C_i \quad (4.46)$$

Avec A_i et C_i sont définies dans l'équation (4.30) et les éléments de la matrice Q_i sont donnés par :

$$Q_i(l, k) = (-1)^{l+k} \frac{(l+k-2)!}{(l-1)!(k-1)!} \quad (4.47)$$

ii). Nous définissons le nouveau vecteur d'état Z_i , dont :

$$Z_i = \Delta_{di}^{-1} \tilde{E}_i \quad (4.48)$$

Avec :

$$\Delta_{di} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \delta_i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \delta_i^{r_i-1} \end{bmatrix} \quad (4.49)$$

Cependant, ce changement de variables implique les propriétés suivantes :

$$\begin{cases} \|Z_i\| \leq \|\tilde{E}_i\| \quad , \quad \|\tilde{E}_i\| \leq \delta_i^{r_i-1} \|Z_i\| \\ \tilde{e}_i = C_i \tilde{E}_i = C_i \Delta_{di} Z_i = C_i Z_i = z_i \\ \Delta_{di}^{-1} A_i \Delta_{di} = \delta_i A_i \quad , \quad \Delta_{di}^{-1} \Delta_i = \bar{\mathcal{G}}_i \end{cases} \quad (4.50)$$

Tel que :

$$\bar{\mathcal{G}}_i = [\mathcal{G}_{i,1} \quad \mathcal{G}_{i,2} \quad \dots \quad \mathcal{G}_{i,r_i}]^T \in \mathfrak{R}^{r_i} \quad (4.51)$$

Le vecteur $\bar{\mathcal{G}}_i$ doit vérifier la relation suivante :

$$\bar{\mathcal{G}}_i = Q_i^{-1} C_i^T \quad (4.52)$$

En utilisant, ce nouveau vecteur d'état, le système d'équation (4.42) peut prendre la forme suivante :

$$\begin{cases} \dot{Z}_i = (\delta_i A_i - \Delta_{di}^{-1} B_i \Phi_{0i} \Delta_{di}) Z_i + \Delta_{di}^{-1} B_i \dot{s}_i - Q_i^{-1} C_i^T F(\tilde{e}_i) \\ z_i = C_i Z_i \end{cases} \quad (4.53)$$

iii). L'utilisation des équations (4.8), (4.10), (4.15), (4.16) et (4.25), conduit au résultat suivant :

$$\dot{s}_i = v_i - f_i(X) - g_i(X) u_i^* + g_i(X) \tilde{u}_i = -g_i(k_i s_i - k_i \tilde{s}_i - w_{i,1}) \quad (4.54)$$

D'où:

$$\begin{cases} \tilde{u}_i = u_i^* - u_i = \tilde{u}_i^{eq} + \tilde{u}_i^{pd} - u_i^r \\ \tilde{u}_i^{pd} = u_i^{pd*} - u_i^{pd} = \kappa_i \tilde{s}_i \end{cases} \quad (4.55)$$

On a :

$$\tilde{s}_i = s_i - \hat{s}_i = \Phi_i \tilde{E}_i = \Phi_i \Delta_{di} Z_i \quad (4.56)$$

Et:

$$w_{i,1} = \tilde{W}_i^T \sigma(\hat{V}_i^T \hat{\chi}_i) + \hat{W}_i^T \sigma'(\hat{V}_i^T \hat{\chi}_i) \tilde{V}_i^T \hat{\chi}_i + w_i - u_i^r \quad (4.57)$$

• **Hypothèse 4.6.** $|w_{i,1}| \leq \frac{c_{i,1}}{\delta_i^{r_i-1}} \|\tilde{W}_i\| + \frac{c_{i,2}}{\delta_i^{r_i-1}} \|\tilde{V}_i\| + \frac{c_{i,3}}{\delta_i^{r_i-1}}$ avec $c_{i,1}$, $c_{i,2}$ et $c_{i,3}$ sont des constantes positives est inconnues.

iv). $\forall \bar{x} = [x_1 \ \dots \ x_{r_i}] \in \mathcal{R}^{r_i}$, l'inégalité suivante est vérifiée :

$$\bar{x}^T \left(C_i^T F^*(x_1) - \frac{\delta_i}{2} C_i^T C_i \bar{x} \right) \geq 0 \quad (4.58)$$

• **Théorème 4.1.** Considérons le système non linéaire, MIMO (4.2), la loi de commande (4.15), avec les termes définis par (4.16), (4.18), (4.19), (4.21), les lois d'adaptation (4.27), et l'observateur neuronal adaptatif (4.38) avec ses lois d'adaptation (4.48) dont la condition (4.58) est vérifiée. Supposons que les hypothèses 4.1-4.6 sont satisfaites et les paramètres de conception sont choisis tels que : $\delta_i > \max(\beta_7, \beta_8)$, $\kappa_i > \max(\frac{\beta_4}{\beta_3}, \frac{\beta_6}{\beta_5})$ avec : β_i sont des constantes positives, alors tous les signaux du système en boucle fermée sont Uniformément Ultiment Bornées (UUB).

• **Preuve du Théorème 4.1.** pour démontrer le théorème ci-dessus, la fonction de Lyapunov candidate suivante est utilisée :

$$L = L_1 + L_2 \quad (4.59)$$

Avec :

$$L_1 = \frac{1}{2} \sum_{i=1}^p \left(Z_i^T Q_i Z_i + F_{\mu_i}^{-1} \tilde{\mu}_i^2 + \tilde{A}_i^T F_{A_i}^{-1} \tilde{A}_i \right) \quad (4.60)$$

Et :

$$L_2 = \frac{1}{2} \sum_{i=1}^p \left(g_i^{-1} s_i^2 + \text{tr} \left(\tilde{W}_i^T F_{W_i}^{-1} \tilde{W}_i \right) + \text{tr} \left(\tilde{V}_i^T F_{V_i}^{-1} \tilde{V}_i \right) + \frac{1}{\gamma_i} \tilde{\rho}_i^2 \right) \quad (4.61)$$

Avec : $\tilde{\rho}_i = w_i - \hat{\rho}_i$ et la première dérivée de L_1 par rapport au temps est donnée par :

$$\dot{L}_1 = \sum_{i=1}^p \left(\frac{1}{2} Z_i^T Q_i \dot{Z}_i + \frac{1}{2} \dot{Z}_i^T Q_i Z_i + F_{\mu_i}^{-1} \tilde{\mu}_i \dot{\tilde{\mu}}_i + \tilde{A}_i^T F_{A_i}^{-1} \dot{\tilde{A}}_i \right) \quad (4.62)$$

En utilisant l'équation définie par (4.53) et les lois d'adaptation de l'observateur adaptatif neuronal données par (4.48), l'expression dans (4.62) devient :

$$\dot{L}_1 = \sum_{i=1}^p \left(\begin{aligned} & -Z_i^T Q_i \left(\frac{\delta_i}{2} I + \Delta_{di}^{-1} B_i \Phi_{0i} \Delta_{di} \right) Z_i - Z_i^T \left(C_i^T F^*(\tilde{e}_i) - \frac{\delta_i}{2} C_i^T C_i Z_i \right) \\ & + Z_i^T C_i^T \tilde{F}(\tilde{e}_i) + Z_i^T Q_i \Delta_{di}^{-1} B_i \dot{s}_i - \tilde{\mu}_i T(\hat{A}_i^T \bar{E}_i) \tilde{e}_i - \tilde{A}_i^T \bar{E}_i \tilde{e}_i \hat{\mu}_i T(\hat{A}_i^T \bar{E}_i) \\ & - \kappa_{\mu_i} \tilde{\mu}_i (\bar{\mu}_i - \hat{\mu}_i) - \tilde{A}_i^T \kappa_{A_i} (\bar{A}_i - \hat{A}_i) \end{aligned} \right) \quad (4.63)$$

Par l'implémentation des équations (4.43), (4.54), la condition dans l'équation (4.58) et par l'utilisation des hypothèses (4.5 et 4.6), dans l'équation (4.63) on aura :

$$\dot{L}_1 \leq \sum_{i=1}^p \left(\begin{aligned} & -Z_i^T Q_i \left(\frac{\delta_i}{2} I + \Delta_{di}^{-1} B_i \Phi_{0i} \Delta_{di} + \Delta_{di}^{-1} B_i g_i k_i \Phi_i \Delta_{di} \right) Z_i + \tilde{e}_i w_{i,o} \\ & + \left\| Q_i \Delta_{di}^{-1} B_i g_i k_i \right\| \|Z_i\| |s_i| + \left\| Q_i \Delta_{di}^{-1} B_i g_i \right\| \|Z_i\| (c_{i,1} \|\tilde{W}_i\| + c_{i,2} \|\tilde{V}_i\| + c_{i,3}) \\ & + \kappa_{\mu_i} \left(\bar{\mu}_i |\tilde{\mu}_i| + \mu_{i,m} |\tilde{\mu}_i| - |\tilde{\mu}_i|^2 \right) + \kappa_{A_i} \left(\bar{A}_i \|\tilde{A}_i\| + A_{i,m} \|\tilde{A}_i\| - \|\tilde{A}_i\|^2 \right) \end{aligned} \right) \quad (4.64)$$

Par l'utilisation de l'ensemble des inégalités suivantes :

$$\left\{ \begin{aligned} & \left\| Q_i \Delta_{di}^{-1} B_i g_i k_i \right\| \leq \frac{\bar{g}_i k_i}{\delta_i^{r_i-1}} \|Q_i\|, \left\| Q_i \Delta_{di}^{-1} B_i g_i \right\| \leq \frac{\bar{g}_i}{\delta_i^{r_i-1}} \|Q_i\|, \left\| Q_i \Delta_{di}^{-1} B_i \Phi_{0i} \Delta_{di} \right\| \leq \|Q_i\| \|\Phi_{0i}\| \\ & \left\| Q_i \Delta_{di}^{-1} B_i g_i k_i \Phi_i \Delta_{di} \right\| \leq \bar{g}_i k_i \|Q_i\| \|\Phi_i\|, \tilde{e}_i w_{i,o} \leq \bar{w}_{i,o} \|Z_i\|, -\frac{1}{2} |\tilde{\mu}_i|^2 + \bar{\mu}_i |\tilde{\mu}_i| \leq \frac{\bar{\mu}_i^2}{2} \\ & -\frac{1}{2} |\tilde{\mu}_i|^2 + \mu_{i,m} |\tilde{\mu}_i| \leq \frac{\mu_{i,m}^2}{2}, -\frac{1}{2} \|\tilde{A}_i\|^2 + \bar{A}_i \|\tilde{A}_i\| \leq \frac{\bar{A}_i^2}{2}, -\frac{1}{2} \|\tilde{A}_i\|^2 + A_{i,m} \|\tilde{A}_i\| \leq \frac{A_{i,m}^2}{2}. \end{aligned} \right. \quad (4.65)$$

dans (4.64), on obtient :

$$\dot{L}_1 \leq \sum_{i=1}^p \left(\begin{aligned} & -\left(\frac{\delta_i}{2} \|Q_i\|_{\min} - \|\Phi_{0i}\| \|Q_i\| - \bar{g}_i k_i \|\Phi_i\| \|Q_i\| \right) \|Z_i\|^2 + \frac{\bar{g}_i k_i}{\delta_i^{r_i-1}} \|Q_i\| \|Z_i\| |s_i| \\ & + c_{i,1} \frac{\bar{g}_i}{\delta_i^{2r_i-2}} \|Q_i\| \|Z_i\| \|\tilde{W}_i\| + c_{i,2} \frac{\bar{g}_i}{\delta_i^{2r_i-2}} \|Q_i\| \|Z_i\| \|\tilde{V}_i\| + \left(c_{i,3} \frac{\bar{g}_i}{\delta_i^{2r_i-2}} \|Q_i\| + \bar{w}_{i,o} \right) \|Z_i\| \\ & + \frac{\kappa_{\mu_i}}{2} (\bar{\mu}_i^2 + \mu_{i,m}^2) + \frac{\kappa_{A_i}}{2} (\bar{A}_i^2 + A_{i,m}^2) \end{aligned} \right) \quad (4.66)$$

Avec $\|\cdot\|_{\min}$ est la norme minimale et $\|\cdot\|$ est la norme maximale d'une matrice donnée.

Les mêmes étapes pour le deuxième terme de la fonction de Lyapunov (4.61), dont, sa première dérivée par rapport au temps est donnée par :

$$\dot{L}_2 = \sum_{i=1}^p \left(g_i^{-1} s_i \dot{s}_i + \text{tr} \left(\tilde{W}_i^T F_{W_i}^{-1} \dot{\tilde{W}}_i \right) + \text{tr} \left(\tilde{V}_i^T F_{V_i}^{-1} \dot{\tilde{V}}_i \right) + \frac{1}{\gamma_i} \tilde{\rho}_i \dot{\tilde{\rho}}_i \right) \quad (4.67)$$

En utilisant les lois de commandes dans (4.15), (4.16), (4.18), (4.19), (4.21), les lois d'adaptation (4.27), l'expression de l'erreur filtrée dans (4.54), et le fait que :

$$\begin{cases} \text{tr} \left(\tilde{W}_i^T \hat{W}_i \right) \leq \|\tilde{W}_i\| W_{i,m} - \|\tilde{W}_i\|^2 \\ \text{tr} \left(\tilde{V}_i^T \hat{V}_i \right) \leq \|\tilde{V}_i\| V_{i,m} - \|\tilde{V}_i\|^2 \end{cases} \quad (4.68)$$

L'équation (4.67) devient :

$$\dot{L}_2 \leq \sum_{i=1}^p \left(-k_i s_i^2 + k_i \delta_i^{r_i-1} \|\Phi_i\| |s_i| \|Z_i\| + c_{i,1} \|\Phi_i\| \|Z_i\| \|\tilde{W}_i\| + c_{i,2} \|\Phi_i\| \|Z_i\| \|\tilde{V}_i\| + c_{i,3} \|\Phi_i\| \|Z_i\| + \kappa_i \left(\|\tilde{W}_i\| W_{i,m} + \|\tilde{V}_i\| V_{i,m} - \|\tilde{W}_i\|^2 - \|\tilde{V}_i\|^2 \right) \right) \quad (4.69)$$

Alors, \dot{L} est :

$$\dot{L} \leq \sum_{i=1}^p \left(-k_i s_i^2 - \left(\frac{\delta_i}{2} \|Q_i\|_{\min} - c_{i,4} \right) \|Z_i\|^2 + 2c_{i,1} c_{i,5} \|Z_i\| \|\tilde{W}_i\| + 2c_{i,2} c_{i,5} \|Z_i\| \|\tilde{V}_i\| + 2c_{i,3} (c_{i,5} + c_{i,6}) \|Z_i\| + 2k_i \delta_i^{r_i-1} c_{i,5} \|Z_i\| |s_i| + \frac{\kappa \mu_i}{2} (\bar{\mu}_i^2 + \mu_{i,m}^2) + \frac{\kappa A_i}{2} (A_i^2 + A_{i,m}^2) + \kappa_i \left(\|\tilde{W}_i\| W_{i,m} + \|\tilde{V}_i\| V_{i,m} - \|\tilde{W}_i\|^2 - \|\tilde{V}_i\|^2 \right) \right) \quad (4.70)$$

Avec :

$$\left\{ c_{i,4} = \left(\|\Phi_{0i}\| + \bar{g}_i k_i \|\Phi_i\| \right) \|Q_i\|, c_{i,5} = \frac{\bar{g}_i}{2 \delta_i^{2r_i-2}} \|Q_i\| + \frac{\|\Phi_i\|}{2}, c_{i,6} = \frac{\bar{w}_{i,0}}{2 c_{i,3}} \right. \quad (4.71)$$

En remplaçant les inégalités suivantes dans (4.70) :

$$\begin{cases} 2\|Z_i\| |s_i| \leq \frac{\|Z_i\|^2}{\alpha_{i,1}} \delta_i^{r_i-1} + \frac{\alpha_{i,1}}{\delta_i^{r_i-1}} |s_i|^2, & 2\|Z_i\| \|\tilde{W}_i\| \leq \frac{\|Z_i\|^2}{\alpha_{i,2}} + \alpha_{i,2} \|\tilde{W}_i\|^2 \\ 2\|Z_i\| \|\tilde{V}_i\| \leq \frac{\|Z_i\|^2}{\alpha_{i,3}} + \alpha_{i,3} \|\tilde{V}_i\|^2, & 2\|Z_i\| \leq \frac{\|Z_i\|^2}{\alpha_{i,4}} + \alpha_{i,4} \\ 2\|\tilde{W}_i\| W_{i,m} \leq \frac{\|\tilde{W}_i\|^2}{\alpha_{i,5}} + \alpha_{i,5} W_{i,m}^2, & 2\|\tilde{V}_i\| V_{i,m} \leq \frac{\|\tilde{V}_i\|^2}{\alpha_{i,6}} + \alpha_{i,6} V_{i,m}^2 \end{cases} \quad (4.72)$$

avec α_i sont des constantes positives choisies suffisamment petites, nous obtenons :

$$\dot{L} \leq \sum_{i=1}^p \left(\begin{array}{l} -k_i (1 - \beta_1) s_i^2 - \delta_i^{r_i-1} \left(\frac{\|Q_i\|_{\min}}{2\delta_i^{2r_i-1}} - \beta_2 \right) \|\tilde{E}_i\|^2 - (\beta_3 \kappa_i - \beta_4) \|\tilde{W}_i\|^2 - (\beta_5 \kappa_i - \beta_6) \|\tilde{V}_i\|^2 \\ + \bar{\beta} \end{array} \right) \quad (4.73)$$

Avec :

$$\left\{ \begin{array}{l} \beta_1 = c_{i,5} \alpha_{i,1}, \beta_2 = c_{i,5} \left(\frac{k_i \delta_i^{2r_i-2}}{\alpha_{i,1}} + \left(\frac{c_{i,1}}{\alpha_{i,2}} + \frac{c_{i,2}}{\alpha_{i,3}} + \frac{c_{i,3}}{\alpha_{i,4}} \right) \right) + c_{i,4} + \frac{c_{i,3} c_{i,6}}{\alpha_{i,4}} \\ , \beta_3 = \left(1 - \frac{1}{2\alpha_{i,5}} \right), \beta_4 = c_{i,1} c_{i,5} \alpha_{i,2}, \beta_5 = \left(1 - \frac{1}{2\alpha_{i,6}} \right), \beta_6 = c_{i,2} c_{i,5} \alpha_{i,3}, \\ \bar{\beta} = c_{i,3} (c_{i,5} + c_{i,6}) \alpha_{i,4} + \frac{\kappa_i}{2} (\alpha_{i,5} W_{i,m}^2 + \alpha_{i,6} V_{i,m}^2) + \frac{\kappa \mu_i}{2} (\bar{\mu}_i^2 + \mu_{i,m}^2) + \frac{\kappa A_i}{2} (\bar{A}_i^2 + A_{i,m}^2) \end{array} \right. \quad (4.74)$$

L'expression suivante est valable :

$$\beta_2 \leq \bar{\sigma}_{i,1} \delta_i^{2r_i-1} + \bar{\sigma}_{i,2} \quad (4.75)$$

Avec $\bar{\sigma}_{i,1}, \bar{\sigma}_{i,2}$ sont des constantes positives.

En choisissant les paramètres de conception δ_i et κ_i à l'intérieur de l'intervalle

$$\delta_i > \max(\beta_7, \beta_8), \quad \kappa_i > \max\left(\frac{\beta_4}{\beta_3}, \frac{\beta_6}{\beta_5}\right) \text{ avec :}$$

$$\beta_7 = \left(\frac{2 - \alpha_{i,1}}{\alpha_{i,1} \bar{g}_i \|Q_i\|} \right)^{\frac{1}{2r_i-2}} \quad (4.76)$$

et

$$\beta_8 = \left(\frac{-\bar{\sigma}_{i,2} + \sqrt{\bar{\sigma}_{i,2}^2 + 2\bar{\sigma}_{i,1} \|Q_i\|_{\min}}}{2\bar{\sigma}_{i,1}} \right)^{\frac{1}{2r_i-1}} \quad (4.77)$$

nous pouvons conclure que $\dot{L} < 0$ est négatif, si les signaux $s_i, \tilde{E}_i, \tilde{W}_i$ et \tilde{V}_i sont en dehors des ensembles compacts $\Omega_{s_i}, \Omega_{\tilde{E}_i}, \Omega_{\tilde{W}_i}$ et $\Omega_{\tilde{V}_i}$ donnés par :

$$\left\{ \begin{array}{l} \Omega_{s_i} = \left\{ s_i / \|s_i\| \leq \sqrt{\frac{\bar{\beta}}{k_i (1 - \beta_1)}} \right\}, \quad \Omega_{\tilde{E}_i} = \left\{ \tilde{E}_i / \|\tilde{E}_i\| \leq \sqrt{\frac{2\bar{\beta} \delta_i^{2r_i-1}}{\|Q_i\|_{\min} - 2\bar{\sigma}_{i,1} \delta_i^{2r_i-1} - 2\bar{\sigma}_{i,2}}} \right\} \\ \Omega_{\tilde{W}_i} = \left\{ \tilde{W}_i / \|\tilde{W}_i\| \leq \sqrt{\frac{\bar{\beta}}{\beta_3 \kappa_i - \beta_4}} \right\}, \quad \Omega_{\tilde{V}_i} = \left\{ \tilde{V}_i / \|\tilde{V}_i\| \leq \sqrt{\frac{\bar{\beta}}{\beta_5 \kappa_i - \beta_6}} \right\} \end{array} \right. \quad (4.78)$$

Selon le théorème standard de Lyapunov [Ioa96], nous concluons que s_i , \tilde{E}_i , \tilde{W}_i , \tilde{V}_i sont bornés et convergent aux ensembles compacts Ω_{s_i} , $\Omega_{\tilde{E}_i}$, $\Omega_{\tilde{W}_i}$, $\Omega_{\tilde{V}_i}$. De plus, le rayon de ces ensembles peut être rendu arbitrairement petit si k_i , δ_i , κ_i sont choisis suffisamment grands.

4. Résultats de simulation

Pour valider le schéma de commande proposé des simulations ont été effectuées sur le modèle dynamique d'un quadrotor. Selon, le système d'équation présenté dans (3.7), il est clair que le quadrotor est un système sous actionné composé de six sorties $[\varphi \ \theta \ \psi \ x \ y \ z]^T$ contrôlées par quatre entrée seulement $[u_\varphi \ u_\theta \ u_\psi \ u_1]^T$. Pour commander chaque sortie, l'une indépendamment de l'autre, une solution basée sur les contraintes holonomes a été proposée dans le chapitre précédent. Dans le même contexte, et afin d'avoir la forme de l'équation (4.1) pour le modèle dynamique du quadrotor, trois commandes virtuelles u_x, u_y et u_z sont construites en utilisant les paramètres φ, θ et u_1 , ces commandes virtuelles sont définies par:

$$\begin{cases} u_x = (C_\psi S_\theta C_\varphi + S_\varphi S_\psi)u_1 \\ u_y = (S_\psi S_\theta C_\varphi - S_\varphi C_\psi)u_1 \\ u_z = (C_\theta C_\varphi)u_1 \end{cases} \quad (4.79)$$

Contrairement au chapitre précédent, les commandes virtuelles ici ont une signification physique qui exprime la relation entre le déplacement en 3D et les trois paramètres communs φ, θ et u_1 . Ces derniers sont les commandes implicites de ces mouvements. Autrement dit, la force de portance et les trajectoires en roulis et en tangage doivent prendre des trajectoires bien spécifier afin d'achever l'objectif de la commande.

Les trajectoires désirées en roulis (φ_d) et en tangage (θ_d) et u_{1d} sont obtenues par l'inversion du système d'équation (4.79) sont données par:

$$\begin{cases} \varphi_d = \arcsin\left(\frac{u_x S_{\psi_d} - u_y C_{\psi_d}}{\sqrt{u_x^2 + u_y^2 + u_z^2}}\right), \\ \theta_d = \arctan\left(\frac{C_{\psi_d} u_x + S_{\psi_d} u_y}{u_z}\right), \\ u_{1d} = \sqrt{u_x^2 + u_y^2 + u_z^2}. \end{cases} \quad (4.80)$$

La figure suivante présente le système global (quadrotor avec les commandes virtuelles) avec six entrées et six sorties.

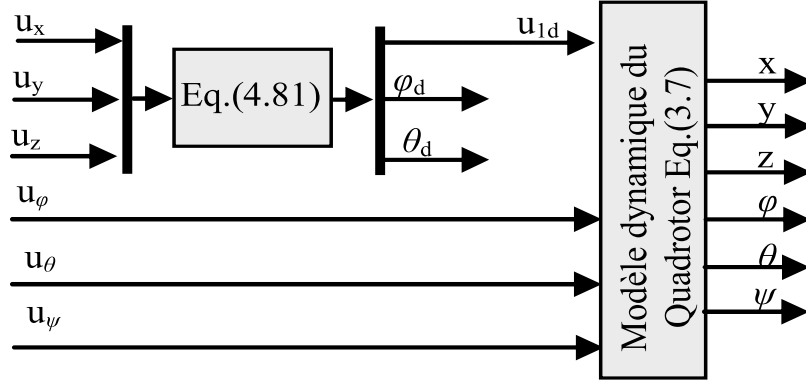


Figure 4.4. Quadrotor avec les commandes virtuelles.

La partie contrôleur est composé de six RNAs qu'ont la forme représentée sur la Figure 4.1 Les RNAs ont été utilisés dans le schéma de commande afin d'estimer la commande équivalente dans l'équation (4.12) pour chaque sous-système i . Chaque RNA est composé d'un seul neurone dans la couche de sortie, cinq neurones dans la couche d'entrée et trois neurones dans la couche cachée. Le vecteur d'entrée pour chaque réseau est donné par $\hat{\lambda}_i = [\ddot{Y}_{di} \quad \dot{Y}_{di} \quad Y_{di} \quad \dot{\hat{e}}_i \quad \hat{e}_i]^T, i = \overline{1:p}$ avec $Y_d = [Y_{d1} \quad Y_{d2} \quad Y_{d3} \quad Y_{d4} \quad Y_{d5} \quad Y_{d6}]^T = [\varphi_d \quad \theta_d \quad \psi_d \quad x_d \quad y_d \quad z_d]^T$ et $e = [\varphi_d - \varphi \quad \theta_d - \theta \quad \psi_d - \psi \quad x_d - x \quad y_d - y \quad z_d - z]^T$. Tous les poids de ces RNAs sont initialisés à zéro.

La partie observateur est composé de six RNAs qu'ont la forme représentée sur la Figure 4.2. Les RNAs sont utilisés dans le terme de correction de l'observateur proposé. Chaque RNA est composé d'un seul neurone dans les couches de sortie et cachée et deux neurones dans la couche d'entrée. Le vecteur d'entrée pour chaque réseau est donné par $\bar{E}_i = [\hat{e}_i \quad \dot{\hat{e}}_i]^T, i = \overline{1:p}$ avec $e = [\varphi_d - \varphi \quad \theta_d - \theta \quad \psi_d - \psi \quad x_d - x \quad y_d - y \quad z_d - z]^T$.

Les bruits de mesure sont supposés des bruits blanc gaussiens [Abe11, Ale12, Bat11, Din13, Mok05, Rub13, Zha11] à moyenne nulle et de variance Var [Din13, Lan13, Mok05]. Dans notre application la valeur de variance (Var) est donnée par : $\text{Var} = [V_\varphi \quad V_\theta \quad V_\psi \quad V_x \quad V_y \quad V_z]^T = [0.003 \quad 0.003 \quad 0.01 \quad 0.05 \quad 0.05 \quad 0.05]^T$.

La configuration générale du schéma de commande pour le quadrotor en utilisant la technique proposée est illustrée sur la Figure 4.5.

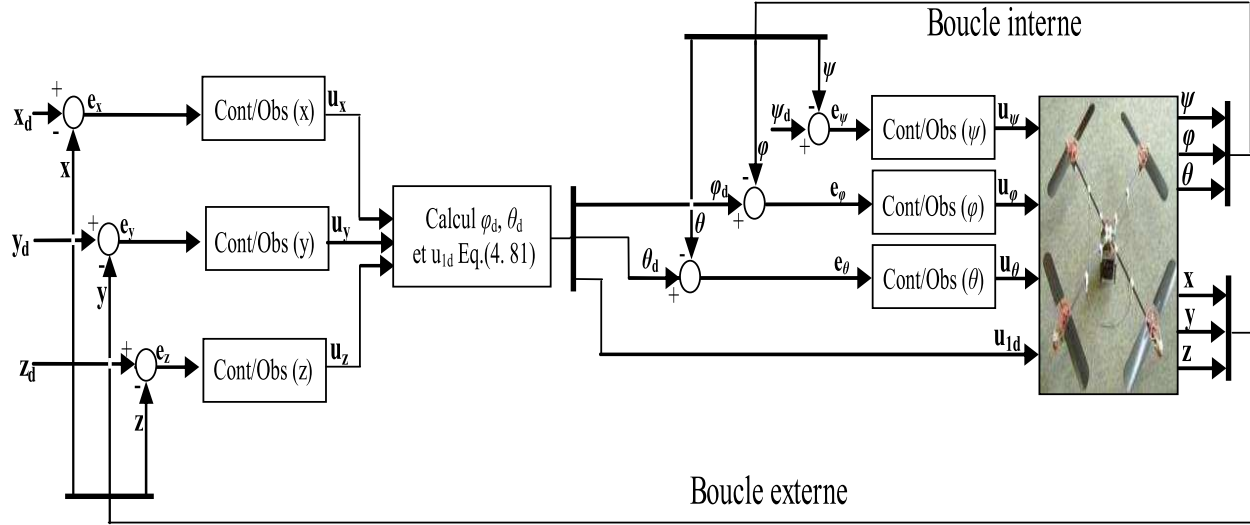


Figure 4.5. Commande adaptative par les réseaux de neurones artificiels basée sur un observateur adaptatif neuronal d'un quadrotor.

Les paramètres de simulation pour le contrôleur adaptatif et l'observateur neuronal adaptatif sont donnés par le tableau 4.1 et les paramètres du quadrotor sont les mêmes paramètres utilisés dans le troisième chapitre (Tableau 3.1).

Tableau 4.1. Paramètres Contrôleur/ Observateur.

| Contrôleur | | Observateur adaptatif Neuronal | |
|-------------------------------|------------------|----------------------------------|---|
| Symbol | Valeur | Symbol | Valeur |
| λ_η | diag(10,10,10) | $\delta_{\zeta,\theta,\varphi}$ | 100 |
| λ_ζ | diag(1,1,10) | $\bar{\mu}_z, \delta_\psi$ | 50 |
| K_η | diag(.2,.2,.2) | Δ_ζ | $\delta_\zeta, \text{diag}(1,20)$ |
| K_ζ | diag(1.5,1.5,20) | $\Delta_{\varphi,\theta}$ | $\delta_{\theta,\varphi} \text{diag}(1,50)$ |
| $F_{W_{x,y,z}}$ | $3 I_{3,3}$ | Δ_ψ | $\delta_\psi \text{diag}(1,40)$ |
| $F_{V_{x,y,z}}$ | $5 I_{5,5}$ | $\bar{A}_{x,y,\psi}$ | [0.3; 1] |
| $F_{W_{\varphi,\theta,\psi}}$ | $I_{3,3}$ | \bar{A}_z | [0.5, 1] |
| $F_{V_{\varphi,\theta,\psi}}$ | $I_{5,5}$ | $\bar{A}_{\theta,\varphi}$ | [0.2, 1] |
| $\kappa_{\zeta,\eta}$ | 0.1 | $\bar{\mu}_{\varphi,\theta,x,y}$ | 20 |
| $\gamma_{\zeta,\eta}$ | 0.01 | $\bar{\mu}_\psi$ | 25 |

• **Remarque 4.3.** Afin d'éviter le phénomène de réticence (chattering) provoqué par le terme discontinu (fonction signe) dans le terme robuste u_1^r (équation (4.18)), une fonction lisse est utilisée, à savoir : tanh, sat,... [Kha96, Slo91]

• **Remarque 4.4.** Une étude comparative est effectuée entre nos résultats et ceux obtenus par l'utilisation d'un OMG conventionnel [Lee07] dont, la fonction signe dans le terme de correction (équation (4.32)) est remplacée par la fonction $T(\cdot)$ définie dans (4.37).

Les Figures (4.6-4.9) montrent les performances en boucle fermée avec une variation paramétrique de 50% à partir de instant $t = 10$ sec.

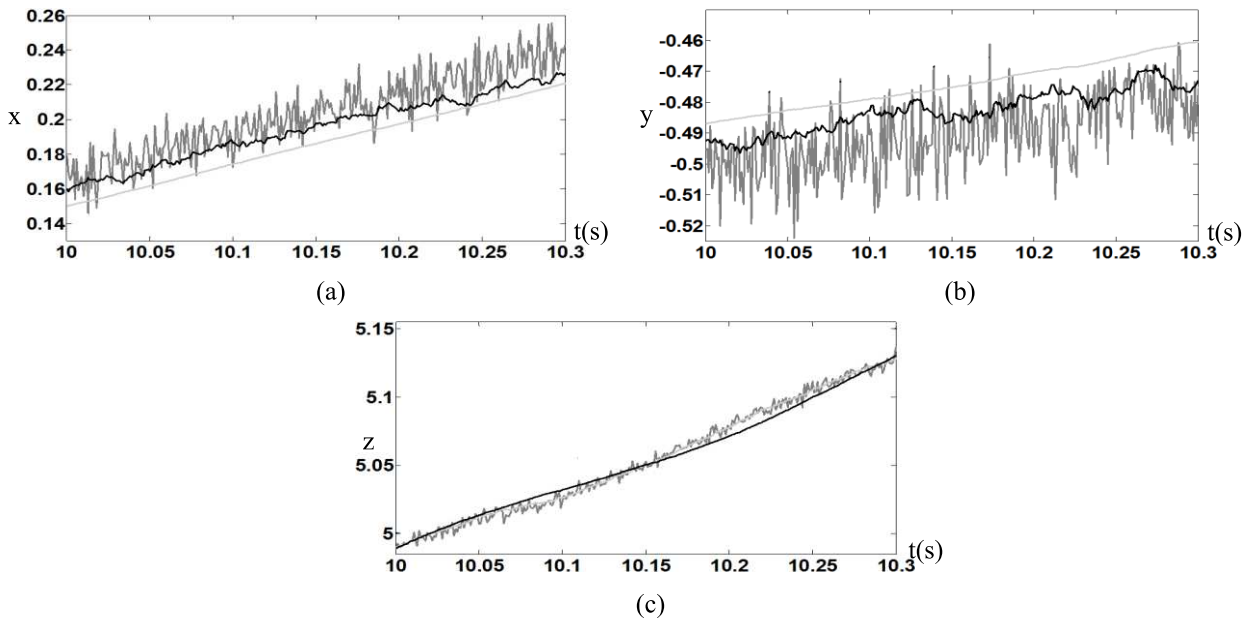
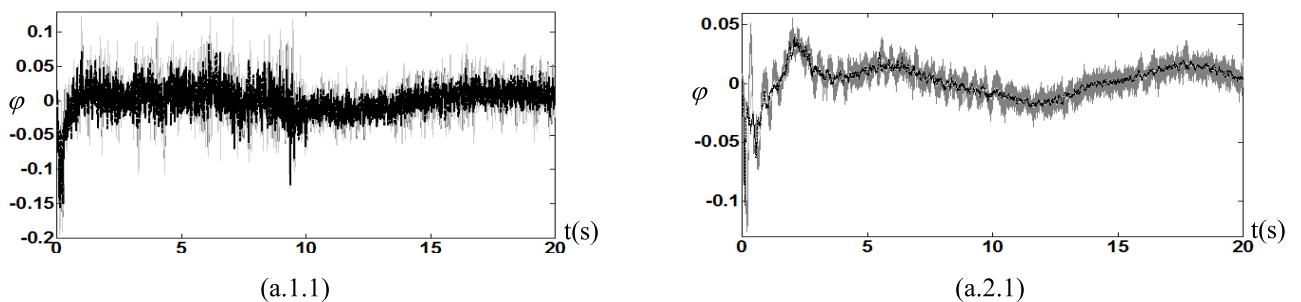


Figure 4.6. Position, (— : trajectoire réelle, - - - : trajectoire estimée par OMG, : trajectoire estimée par l'observateur neuronale proposé.

La Figure 4.6. montre un zoom sur les trajectoires estimées par les deux observateurs (OMG et observateur adaptatif neuronal) ce qui présente les performances de la technique proposée.



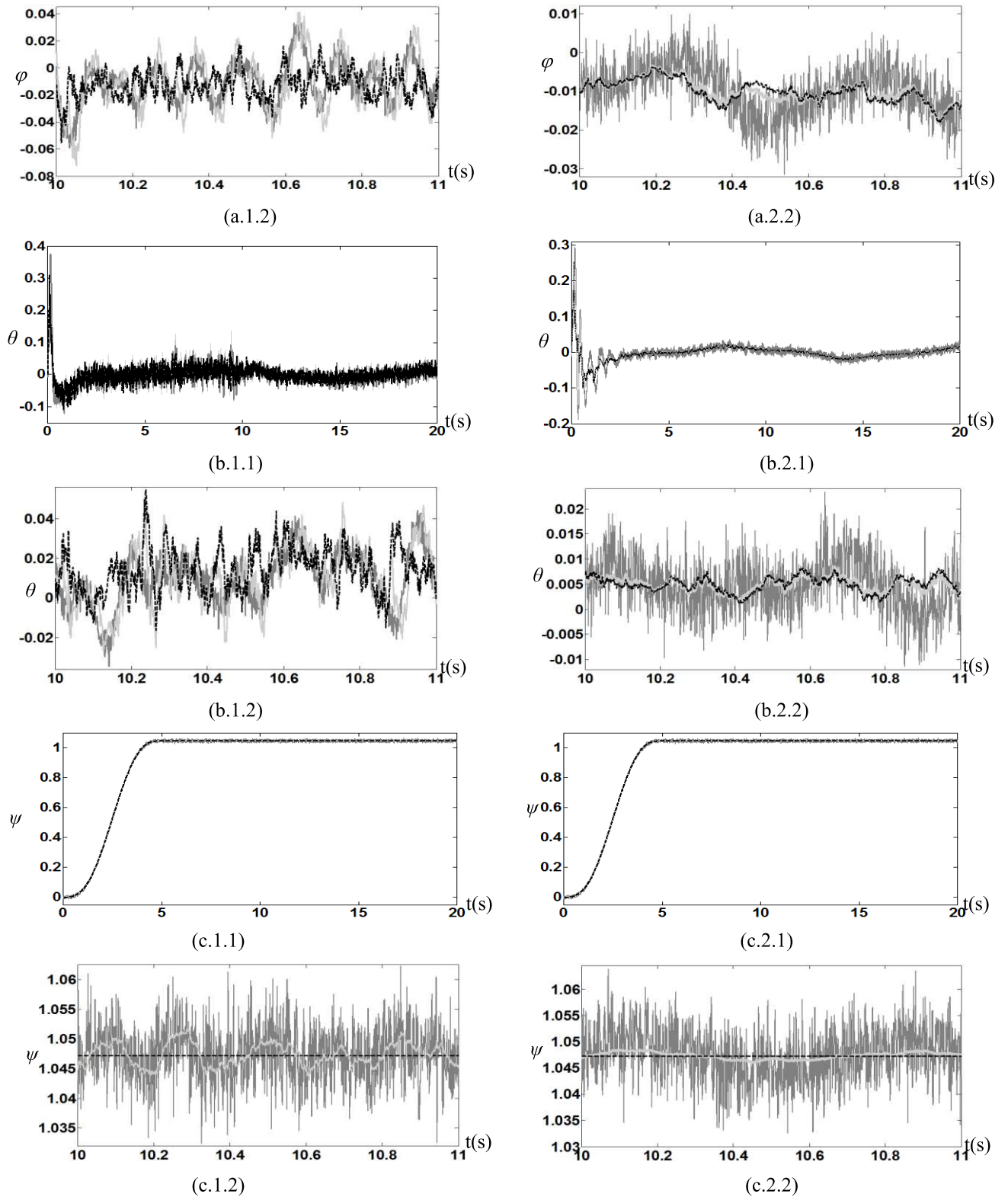


Figure 4.7. Orientation (1^{ière} colonne: OMG, 2^{ième} colonne : observateur neuronal), (- - : trajectoire désirée, — : trajectoire réelle, — : trajectoire estimée).

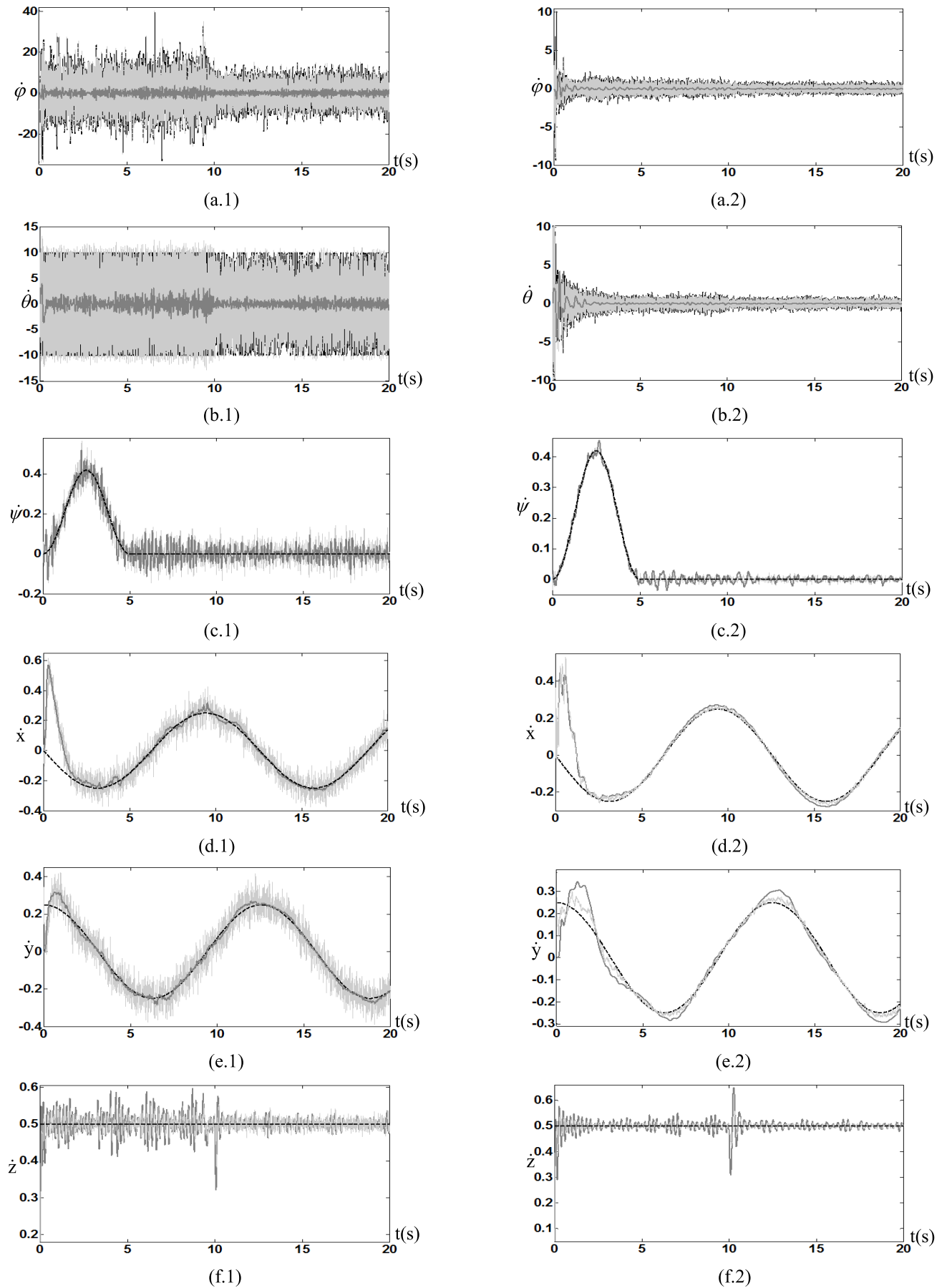


Figure 4.8. Vitesses (1^{ière} colonne: OMG, 2^{ième} colonne: observateur neuronal), (- - : trajectoire désirée, — : trajectoire réelle, — : trajectoire estimée).

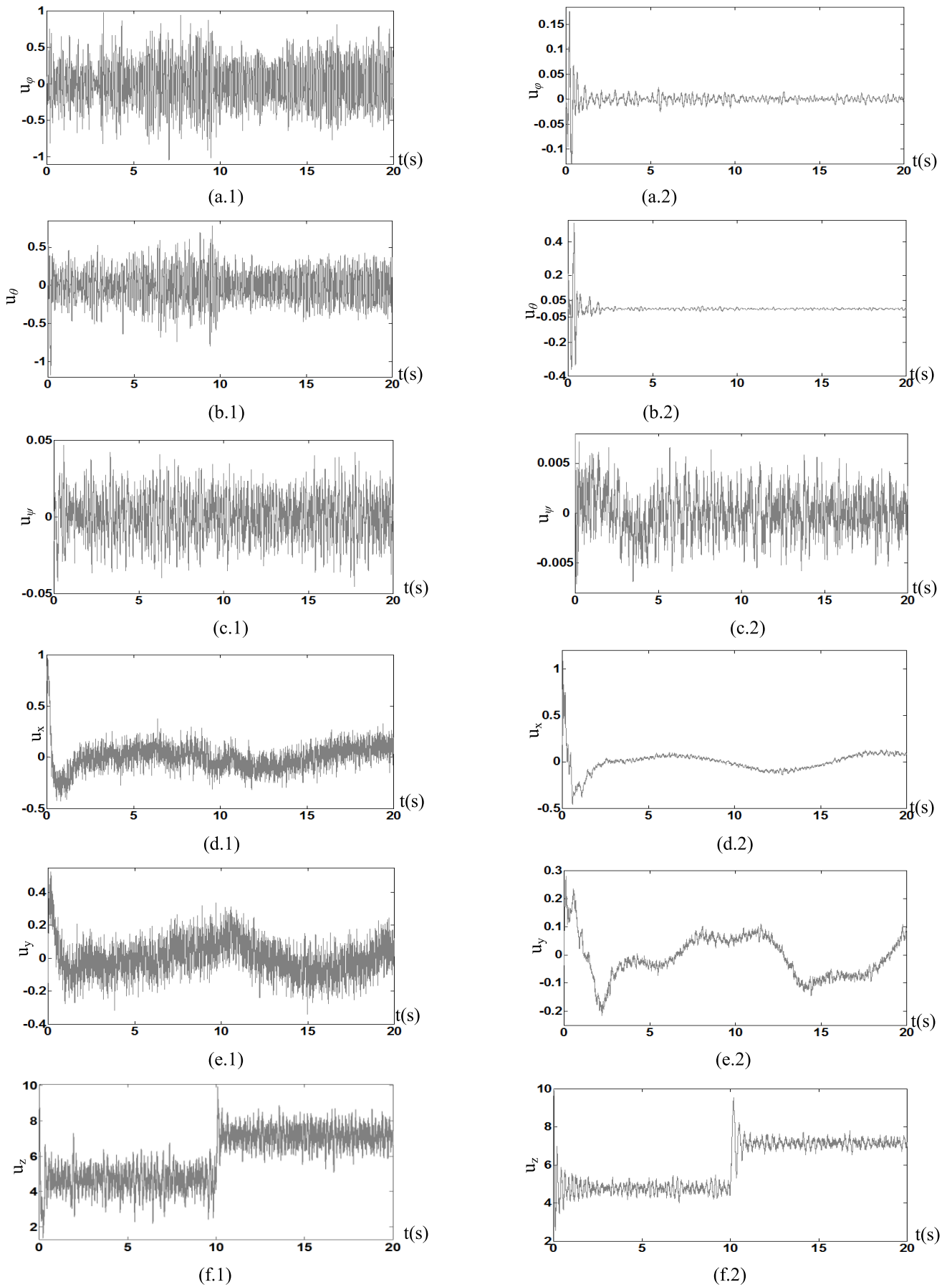


Figure 4.9. Signal de commande (1^{ière} colonne : OMG, 2^{ème} colonne ; observateur neuronal).

Les angles d'Euler obtenues par l'implémentation du l'observateur à mode glissant conventionnel sont trop bruités comparés à ceux obtenues par l'utilisation de l'observateur adaptatif neuronal proposé. Cette amplification des bruits est obtenue principalement sur les angles de roulis φ (Figure 4.7.a.1.2) et de tangage θ (Figure 4.7.b.1.2) et même leurs trajectoires désirées (φ_d, θ_d) sont influencées par cette amplification. Par contre, l'observateur proposé a donné des signaux réels et désirés avec un minimum de bruit comme il est montré sur la Figure 4.7. Au niveau de la poursuite, les résultats sont meilleurs en utilisant l'observateur proposé.

Comparant aux résultats de l'OMG, une réduction très importante du phénomène traité est obtenu pour les signaux des vitesses linéaires et angulaire si l'observateur proposé (Voir Figure 4.8.2 (2^{ième} coloone)).

La Figure 4.9.1 (c'est-à-dire l'OMG) montre que les signaux de commande sont trop bruités et leurs amplitudes est plus grandes. Lorsque l'observateur proposé est utilisé (Figure 4.9.2) les signaux de commandes sont moins bruités. Ce résultats est justifié car la commande dépend de l'orientation (Figure 4.7) et les vitesses (Figure 4.8).

• **Remarque 4.5.** Il faut noter que la méthode proposée dans ce chapitre est applicable aussi pour les systèmes SISO (Single Input Single Output) affine à la commande de la forme :

$$Y^{(r')} = F'(X') + G'(X')U' \quad (4.81)$$

Dont le modèle d'état est donné par :

$$\Sigma = \begin{cases} \dot{x}'_j = x'_{j+1}, & j = \overline{1:r-1} \\ \dot{x}'_r = F'(X') + G'(X')U' \\ Y' = x'_1 + b \end{cases} \quad (4.82)$$

Avec : r' : est le degré relatif du système ci-dessus, $X' = [Y' \ \dots \ Y^{(r'-1)}]^T \in \mathfrak{R}^{r'}$: est le vecteur d'état, $U' \in \mathfrak{R}$: est l'entrée de système, $Y' \in \mathfrak{R}$: est la sortie de système, $F'(X'), G'(X')$: sont des fonctions non linéaires, lissent et inconnues et b : est le bruit de mesure.

5. Conclusion

Dans ce chapitre, une commande adaptative neuronale basée sur l'implémentation d'un observateur adaptatif neuronal a été développée pour une classe des systèmes non linéaires incertains. La méthode proposée est applicable pour tous les systèmes MIMO ou bien SISO affine à la commande. L'approximateur universel RNA a été utilisé d'une part, le premier est pour estimer la commande équivalente et d'une autre part pour générer le terme de correction de

l'observateur. L'objectif principal de ce chapitre est la conception d'un observateur amplifié pas les bruits de mesure comme dans le cas des observateurs conventionnel (l'OMG par exemple). La méthode directe de Lyapunov a été utilisée pour l'analyse de stabilité du système global (contrôleur / observateur). Des simulations avec une étude comparative ont été effectuées sur le modèle dynamique de quadrotor pour montrer les performances de la technique proposée comparant aux résultats obtenus par l'utilisation d'un observateur à mode glissant conventionnel. Le système d'application est sous-actionné, la forme de son modèle ne correspond pas à celle traitée dans ce chapitre, alors, des commandes virtuelles et des contraintes holonomes ont été proposées afin d'obtenir la forme standard. Les résultats ont montrés l'efficacité de l'observateur adaptatif neuronal proposé comparant à l'observateur à mode glissant, dont, les signaux fournis sont moins sensibles aux bruits de mesure et par conséquent les signaux de commande ont été plus lissent et avec des bons amplitudes.

Dans le chapitre suivant, l'observateur à grand gain est associé au filtre de Kalman en cascade afin de réduire les bruits de mesure et améliorer les performances de commande.

Chapitre V

Commande adaptative neuronale avec observateur à grand gain adaptatif et filtre de Kalman modifié en cascade

1. Introduction

La mise en œuvre des lois de commande nécessite couramment la disponibilité de toutes les variables d'état à chaque instant. Néanmoins, ces dernières sont souvent partiellement ou totalement inconnues en pratique. Pour remédier à ce problème, un observateur est implémenté pour reconstruire les variables d'états non disponibles en utilisant les informations entrées/sorties. Dans le cas des systèmes linéaires, la synthèse des observateurs a été largement étudiée [Che95], en particulier, l'observateur de Luenberger [Lue71] pour les systèmes invariants dans le temps et le filtre de Kalman (continu et discret) [Kar10] pour les systèmes sous l'effet des bruits. Le principe de séparation est vérifié facilement pour les systèmes linéaires. Cependant, la démonstration de ce principe présente un grand challenge pour les systèmes non linéaires.

Un Observateur à Grand Gain (OGG) [Xue12], linéaire, simple et robuste [Bul97] est utilisé généralement pour les systèmes non linéaires. Néanmoins, la grande valeur du gain d'observation, de cet observateur produit : le phénomène de pic et l'amplification des bruits de mesure [Kha08]. Les grands dépassements au régime transitoire peuvent déstabiliser le système. Comme solutions à ce problème, on trouve celles basées sur l'implémentation des saturations au niveau des variables d'états et /ou le signal de commande [Ses00, Kha08]. Dans ce cas, la bornitude de la loi de commande protège le système du phénomène de pic.

Le problème de l'amplification des bruits de mesure est traité dans plusieurs travaux : dans [Ahr09], le gain d'observateur bascule entre deux valeurs distinctes, la plus grande pour le régime transitoire et la plus petite pour le régime permanent. Dans [Bul97, Far09, San11, Oue12]

un observateur à grand gain adaptatif est développé, dont, la loi d'adaptation permet le changement du gain d'observation selon les situations. Cependant dans [Bul97], la loi choisie, ne permet pas la diminution du gain même si l'erreur d'observation est nulle. Une autre solution est donné par [Ton12], dont, un système flou est utilisé pour estimer les termes inconnus du système, et par conséquent l'utilisation d'un gain d'observation avec une valeur moins élevée. Les techniques citées permettent de produire des signaux moins sensibles aux bruits de mesure mais dans un temps important de plus l'observateur peut perdre ces propriétés de convergence.

Le Filtre de Kalman (FK) est un observateur utilisé pour les systèmes linéaires stochastique [Bur01, Sto12]. Cet observateur peut prendre deux formes : la première pour le cas continu et la deuxième forme pour le cas discret. En effet, le cas discret présente un ensemble d'équations mathématiques récursives qui permet l'estimation des états passés (lissage), présents (filtrage) et futurs (prédiction) [Gev11]. Malheureusement, l'exploitation de cet observateur pour les systèmes non linéaires est impossible. Alors, d'autre version étendues de filtre de Kalman sont étudiée dans plusieurs travaux [Boi10, Des08, Mir08]. L'idée principale est de linéariser le système autour d'un point de fonctionnement et d'utiliser un filtre qui correspond au modèle résultant. L'amplification des bruits de mesure causée par les observateurs à grand gain est résolue par l'utilisation du filtre de Kalman dans plusieurs travaux. Dans [Liu12], un observateur à grand gain avec un terme de correction proportionnel intégral (PI) est implémenté. Un filtre de Kalman étendu avec un observateur à grand gain sont développés dans [Boi10]. Le design de ces observateurs est similaire à celui du filtre de Kalman, dont, le gain d'observation est calculé en utilisant l'équation de Riccati.

L'objectif de ce chapitre est la conception d'une loi de commande adaptative neuronale basée sur l'utilisation d'un Observateur à Grand Gain Adaptative (OGGA) et un Filtre de Kalman Modifié (FKM) en cascade pour une classe des systèmes non linéaires incertains. Comme dans le chapitre précédent, la commande équivalente idéale est estimée par un réseau de neurones à une seule couche cachée. L'estimation des états est faite par un observateur à grand gain adaptatif (OGGA) en cascade avec un filtre de Kalman modifié (FKM). Dont, la sortie du premier observateur est introduite dans le deuxième afin de fournir des signaux (variables d'état estimées) insensible aux bruits de mesure. Le théorème de Lyapunov est utilisé pour l'obtention des lois d'adaptation des poids et pour montrer la stabilité et la bornitude de tous les signaux du système bouclé. Une étude comparative a été effectuée entre les résultats de la méthode proposée et les résultats obtenus par d'autres techniques pour valider les performances de la technique proposée.

Ce chapitre est organisé comme suit: la formulation de problème est décrite dans la section 2. La section 3 présente la synthèse de la loi de commande adaptative neuronale, le filtre de Kalman, l'observateur à grand gain et la conception de la technique proposée pour l'estimation du vecteur d'état complet. Des simulations sur un pendule inversé est effectuée, dont, une étude comparative sur un pendule inversé est présentée dans la section 4 afin de montrer les performances de la technique proposée. La conclusion de ce chapitre est donnée dans la section 5.

2. Position de problème

Le système étudié est non-linéaire, SISO (Single Input Single Output), affine à la commande, d'ordre n décrit par les équations différentielles suivantes :

$$y^{(n)} = f(X) + g(X)u \quad (5.1)$$

Et son modèle d'état est donné par :

$$\begin{cases} \dot{x}_i = x_{i+1}, & i = \overline{1 : n-1} \\ \dot{x}_n = f(X) + g(X)u, \\ y = x_1 + b. \end{cases} \quad (5.2)$$

Avec : $u \in \mathfrak{R}$ est l'entrée du système, $y \in \mathfrak{R}$ est la sortie de système, $X = [x_1 \ x_2 \ \dots \ x_n]^T = [y \ \dot{y} \ \dots \ y^{(n-1)}]^T \in \mathfrak{R}^n$ est le vecteur d'état, $f(X)$, $g(X)$ sont des fonctions non linéaires, lisses et inconnues et b : est un bruit blanc gaussien représentant les bruits de mesure.

Quelques hypothèses sont prises en considération dans notre schéma de commande dont les trois premières sont :

- **Hypothèses 5.1:** la sortie y est mesurable.
- **Hypothèses 5.2:** la trajectoire désirée y_d , et ces dérivées sont lisses et bornées.
- **Hypothèses 5.3:** le gain de la commande $g(X)$ est positif, borné ($g(X) \leq \bar{g}$,) et lentement variable dans le temps.

On définit l'erreur de poursuite e par :

$$e = y_d - y \quad (5.3)$$

Et l'erreur filtrée s définie comme suit :

$$s = \left(\frac{d}{dt} + \lambda \right)^{(n-1)} e = \Phi E, \quad \lambda > 0 \quad (5.4)$$

La première dérivée de s par rapport au temps est formulée comme suit :

$$\dot{s} = v - f(X) - g(X)u = e^{(n)} + \Phi_0 E \quad (5.5)$$

Avec, E est le vecteur d'état, présentant l'erreur de poursuite et ses dérivées,

$$E = \begin{bmatrix} e & \dot{e} & \dots & e^{(n-1)} \end{bmatrix}^T \in \mathfrak{R}^n \quad (5.6)$$

Les vecteurs Φ et Φ_0 sont donnés par :

$$\begin{cases} \Phi = [\nu_1 & \dots & \nu_{n-1} & 1], \in \mathfrak{R}^n \\ \Phi_0 = [0 & \nu_1 & \dots & \nu_{n-1}], \in \mathfrak{R}^n \end{cases} \quad (5.7)$$

Et :

$$\begin{cases} v = y_d^{(n)} + \sum_{i=1}^{n-1} \nu_i e^{(i)} \\ \nu_i = \frac{(n-1)!}{(n-i)!(i-1)!} \lambda^{n-i} \end{cases}, i = \overline{1 : n-1} \quad (5.8)$$

Passant par les mêmes étapes, comme dans le quatrième chapitre, on trouve que les lois de commande idéales définies de (4.9) jusqu'à (4.11) sont valides pour le système (5.1) aussi. Cependant, les mêmes problèmes d'incertitude de modèle dynamique (c'est-à-dire la fonction $f(X)$ et le gain de commande $g(X)$) et la non disponibilité du vecteur d'état X sont aussi posés, ce qui implique l'impossibilité d'implémenter cette loi de commande.

Alors, une loi de commande adaptative basée sur l'utilisation d'un réseau de neurones artificiels à une seule couche cachée est synthétisée, dont, l'estimation du vecteur d'état est faite par deux observateurs en cascade, le premier est un Observateur à Grand Gain Adaptatif (OGGA) et le deuxième est un Filtre de Kalman Modifié (FKM), afin de produire des signaux insensibles aux bruits de mesure.

3. Synthèse d'une commande adaptative neuronale avec un observateur à grand gain adaptative et un filtre de Kalman modifié en cascade

Dans cette partie, la loi de commande adaptative basée sur l'implémentation des deux observateurs cités en haut est synthétisée.

3.1. Commande adaptative neuronale

La conception de la loi de commande adaptative à base d'observateur est similaire à celle effectuée dans le quatrième chapitre.

La loi de commande appliquée u au système (5.1) est définie par :

$$u = u^{eq} + u^{pd} + u^r \quad (5.9)$$

Avec :

- u^{eq} : est la commande équivalente estimée par un réseau de neurones artificiel est donnée

par la Figure 4.1 comme suit :

$$u^{eq} = \hat{W}^T \sigma(\hat{V}^T \hat{\chi}) = \hat{W}_1^T \sigma(V_1^T \hat{\chi}_1) - \hat{W}_1^T \sigma'(\hat{V}_1^T \hat{\chi}_1) \tilde{V}_1^T \hat{\chi}_1 - \hat{W}_1^T O(\tilde{V}_1^T \hat{\chi}_1) \quad (5.10)$$

Dont: $\hat{\chi}$: est le vecteur d'entrée de RNA , $\sigma()$: est la fonction d'activation, $\hat{W} \in \mathfrak{R}^{nc \times ns}$: sont les poids estimés des couches cachée/sortie et $\hat{V} \in \mathfrak{R}^{ne \times nc}$: sont les poids estimés des couches entrée/cachée, n_e : est le nombre de neurones dans la couche d'entrée, n_c : est le nombre de neurones dans la couche cachée, n_s : est le nombre de neurones de la couche de sortie, $\tilde{W} = W - \hat{W}$, $\tilde{V} = V - \hat{V}$ sont les erreurs d'estimation des poids (différence entre poids idéale et poids estimé) et l'erreur d'approximation de la commande équivalente est donnée par :

$$\tilde{u}^{eq} = u^{eq*} - u^{eq} = \tilde{W}^T \sigma(\hat{V}^T \hat{\chi}) + \hat{W}^T \sigma'(\hat{V}^T \hat{\chi}) \tilde{V}^T \hat{\chi} + w \quad (5.11)$$

avec :

$$w = W^T (\sigma(V^T \chi) - \sigma(\hat{V}^T \hat{\chi})) + \tilde{W}^T \sigma'(\hat{V}^T \hat{\chi}) \tilde{V}^T \hat{\chi} + W^T O(\tilde{V}^T \hat{\chi})^2 + \varepsilon \quad (5.12)$$

Les lois d'adaptations des poids du RNA sont données par:

$$\begin{cases} \dot{\hat{W}} = F_W (\sigma(\hat{V}^T \hat{\chi}) \hat{s} - \kappa \hat{W}) \\ \dot{\hat{V}} = F_V (\hat{\chi} \hat{s} \hat{W}^T \sigma'(\hat{V}^T \hat{\chi}) - \kappa \hat{V}) \end{cases} \quad (5.13)$$

- u^{pd} : est un terme stabilisant de type PD donné par :

$$u^{pd} = k \hat{s} = k \Phi \hat{E}. \quad (5.14)$$

Avec \hat{E} est l'estimé du E par un observateur d'état

- u^r : est un terme robuste de type mode glissant [Kha96, Slo91] utilisé pour compenser les erreurs d'approximation, comme suit :

$$u^r = \hat{\rho} \text{sign}(\hat{s}) \quad (5.15)$$

Avec la loi d'adaptation suivante :

$$\dot{\hat{\rho}} = \gamma |\hat{s}|, \quad \gamma > 0. \quad (5.16)$$

3.2. Filtre de Kalman

La forme conventionnelle du filtre de Kalman continu et son mode d'emploi sont présentées dans cette partie.

Soit, le système linéaire aléatoire, invariant dans le temps présenté par le système d'équation suivant :

$$\begin{cases} \dot{\zeta} = \bar{A}\zeta + \bar{B}\tau + b_{\zeta} \\ \xi = \bar{C}\zeta + b_{\xi} \end{cases} \quad (5.17)$$

Où : $\zeta \in \mathfrak{R}^n$ est le vecteur d'état, $\tau \in \mathfrak{R}^m$ est le signal d'entrée, supposé déterministe et connu, $\xi \in \mathfrak{R}^p$ est le signal de sortie supposé valable pour la mesure, l'état initial est supposé gaussien dont, sa moyenne est ζ_0 et sa variance est P_0 c'est-à-dire $\bar{\zeta}_0 \sim N(\zeta_0, P_0)$, $b_{\zeta} \in \mathfrak{R}^n$ et $b_{\xi} \in \mathfrak{R}^p$ sont des bruits blancs gaussiens décorrélés (indépendants) de moyennes nulles et variances Q_{ζ} et Q_{ξ} respectivement ($b_{\zeta} \sim N(0, Q_{\zeta}), b_{\xi} \sim N(0, Q_{\xi})$) et les matrices $\bar{A}, \bar{B}, \bar{C}, Q_{\zeta}, Q_{\xi}, P_0$ sont des constantes. Le bruit b_{ζ} est considéré comme le bruit agissant sur le système tandis que b_{ξ} signifié le bruit de mesure.

Le problème de l'estimation au sens de Kalman se résume dans l'utilisation les informations disponible sur $\bar{A}, \bar{B}, \bar{C}, Q_{\zeta}, Q_{\xi}, P_0$ et les mesures sur τ et ξ afin de pouvoir construire le vecteur d'état estimé $\hat{\zeta} \in \mathfrak{R}^n$ à chaque instant.

Le filtre de Kalman continu candidat [Kar10] pour un système linéaire invariant dans le temps de la forme (5.15) est donné par (Figure 5.1) :

$$\begin{cases} \dot{\hat{\zeta}} = \bar{A}\hat{\zeta} + \bar{B}\tau + L e_{\xi} \\ \hat{\xi} = \bar{C}\hat{\zeta} \end{cases} \quad (5.18)$$

Avec e_{ξ} est l'erreur d'estimation définie par :

$$e_{\xi} = \xi - \hat{\xi} \quad (5.19)$$

La dynamique de e_{ζ} est :

$$\dot{e}_{\zeta} = \dot{\zeta} - \dot{\hat{\zeta}} = (\bar{A} - L\bar{C})e_{\zeta} + b_{\zeta} - Lb_{\xi} \quad (5.20)$$

Puisque les bruits b_ζ, b_ξ sont centrés (c'est-à-dire à moyenne nulle) on peut écrire :

$$E\{\dot{e}_\zeta\} = (\bar{A} - L\bar{C})E\{e_\zeta\} \quad (5.21)$$

Avec : $E\{\dot{e}_\zeta\}$ et $E\{e_\zeta\}$ sont les moyennes des signaux \dot{e}_ζ et e_ζ respectivement.

Dans (5.18), L est le gain de Kalman défini comme étant le vecteur de retour qui garantit la stabilité de la matrice $(\bar{A} - L\bar{C})$ et la minimisation de la matrice de covariance $\bar{P}(t)$ avec le temps, dont :

$$L = \bar{P}(t)\bar{C}^T Q_\xi^{-1} \quad (5.22)$$

Et $\bar{P}(t)$ est la matrice de covariance de l'erreur d'estimation (e_ζ) formulée comme suit :

$$\bar{P}(t) = E\left\{ (\zeta - \hat{\zeta})(\zeta - \hat{\zeta})^T \right\} \quad (5.23)$$

Où bien par l'intégration de l'expression suivante :

$$\dot{\bar{P}} = \bar{A}\bar{P} + \bar{P}\bar{A}^T - \bar{P}\bar{C}^T Q_\xi^{-1} \bar{C} \bar{P} + Q_\zeta \quad (5.24)$$

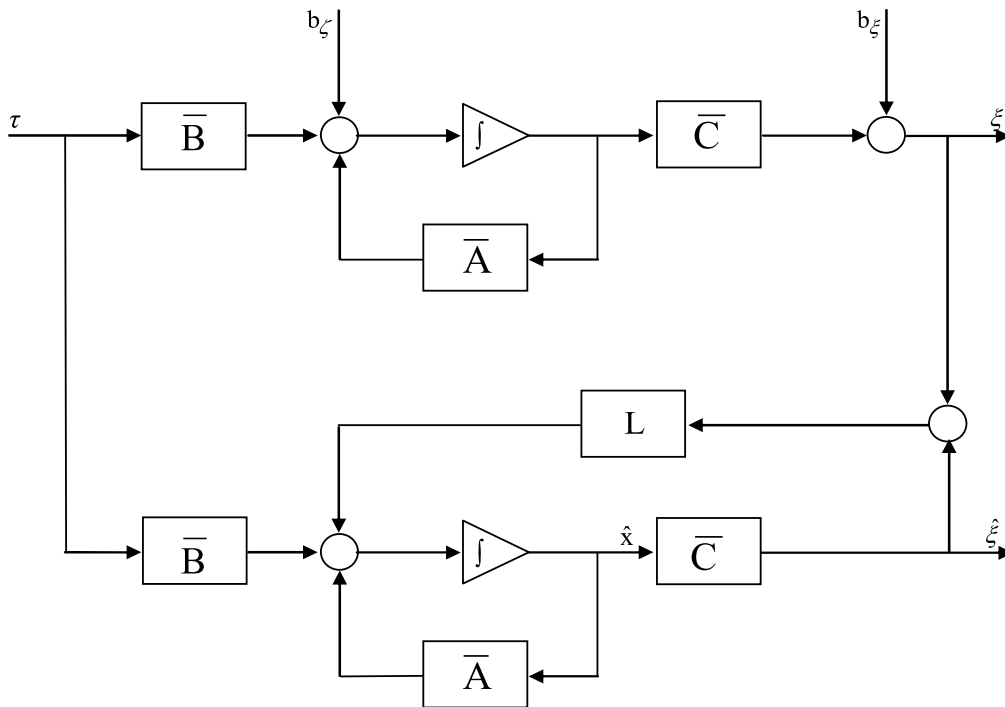


Figure 5.1. Filtre de Kalman

Pour déterminer l'estimateur des variables d'états du système discret, on a trois possibilités : la première est le prédicteur de Kalman, la deuxième est le filtre de Kalman discret

et la troisième est le lissage optimal de Kalman [Gev11]. La différence entre ces trois estimateurs, est dans les mesures utilisées pour la construction des variables d'états.

Cependant, le filtre de Kalman (équation (5.18)) est utilisable pour le cas des systèmes linéaires stochastique décrit par le modèle donné par (5.17). Par la suite, nous allons étudier l'implémentation le filtre de Kalman pour l'estimation des variables d'états pour le cas non linéaire.

3.3. Observateur à grand gain

La forme conventionnelle d'un Observateur à Grand Gain (OGG) est présentée dans cette partie. La classe de système étudié, a une représentation d'état de la forme suivante :

$$\begin{cases} \dot{E} = (A - B\Phi_0)E + B\dot{s} \\ e = C E \end{cases} \quad (5.25)$$

Avec :

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, C^T = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \quad (5.26)$$

En se basant sur le principe de recopier le modèle (5.25) et de compenser les parties non linéaires par un terme de correction, l'observateur à grand gain conventionnel [Far09, Kha08] candidat est de la forme suivante :

$$\begin{cases} \dot{\bar{E}} = (A - B\Phi_0)\bar{E} + \delta \Delta (e - \bar{e}) \\ \bar{e} = C \bar{E} \end{cases} \quad (5.27)$$

Avec $\bar{E} = [\bar{e} \quad \dot{\bar{e}} \quad \dots \quad \bar{e}^{(n-1)}]^T$ est l'estimé de E par l'OGG dans (5.27), $(e - \bar{e})$: est l'erreur d'observation, $\delta \gg 1$ est le gain d'observation et le vecteur Δ est donné par l'équation (5.28), dont, $\vartheta_i > 0 (i = \overline{1:n})$ sont des paramètres de conception.

$$\Delta = [\vartheta_1 \quad \vartheta_2 \delta \quad \dots \quad \vartheta_n \delta^{n-1}]^T \quad (5.28)$$

Pour plus de robustesse avec un OGG, des valeurs très élevées sont généralement attribuer au gain d'observation δ . Néanmoins, deux inconvénients de cet observateur peuvent apparaitre par l'augmentation du gain δ , sont : le phénomène de pic et l'amplification des bruits de mesure [Kha08]. Des solutions ont été proposées dans quelques travaux [Kha08, Ses00] pour éviter le

premier problème. La technique proposée dans ce chapitre est synthétisé afin de réduire les sensibilités aux bruits de mesure.

3.4. Conception d'un observateur à grand gain adaptatif et filtre de Kalman modifié en cascade

Afin de pouvoir implémenter la loi de commande donnée précédemment, un observateur d'état est utilisé pour estimer le vecteur d'état complet E (équation (5.6)). Pour cet objectif plusieurs observateurs peuvent être utilisés. Dans ce contexte, une nouvelle technique pour l'estimation des variables d'état est proposée. Deux différents observateurs sont utilisés en cascade, à savoir : un Observateur à Grand Gain Adaptatif (OGGA) et un Filtre de Kalman Modifié (FKM). Le premier observateur est utilisé pour l'estimation directe de E, par contre le FKM est employé pour estimer la sortie de l'OGGA autrement dit, une deuxième estimation de vecteur E.

3.4.1. Observateur à grand gain adaptatif

L'inconvénient major de cet observateur est la sensibilité aux bruits de mesure due à la valeur élevée du gain d'observation δ . Alors, un OGGA est proposé, dont, un mécanisme d'adaptation est utilisé pour le gain δ , d'où, ce dernier peut changer sa valeur selon l'erreur d'observation et par conséquent rendre l'OGGA moins sensible aux bruits de mesure comparant à l'OGG conventionnel.

L'OGGA proposé pour l'estimation du vecteur d'état E est défini par :

$$\begin{cases} \dot{Z} = A'Z + \delta(t)\Delta(t)\tilde{z} \\ z = CZ \end{cases} \quad (5.29)$$

Avec : $Z = [z_1 \quad z_2 \quad \dots \quad z_n]^T = [z \quad \dot{z} \quad \dots \quad z^{(n-1)}]^T \in \mathfrak{R}^n$ est l'estimé de E par l'OGGA, \tilde{z} ($\tilde{z} = e - z$) est l'erreur d'observation entre l'erreur de poursuite mesurée (e) et l'erreur de poursuite estimée par l'OGGA (z), et la matrice A' est définie par:

$$A' = A - B\Phi_0. \quad (5.30)$$

Le vecteur $\Delta(t)$ est variable avec le temps comme suit :

$$\Delta(t) = [\varrho_1 \quad \varrho_2 \delta(t) \quad \dots \quad \varrho_{n-1} \delta^{n-1}(t)]^T \quad (5.31)$$

La proposition dépend essentiellement du gain d'observation $\delta(t)$, dont, la loi d'adaptation proposée est formulée par l'expression suivante [Bou14]:

$$\dot{\delta}(t) = \varsigma_1 \left(\delta(t) f_o(\tilde{z}) + \varsigma_2 \left(\delta_f - \delta(t) \right) \right) \quad (5.32)$$

Avec: $\varsigma_1, \varsigma_2 > 0$ sont les gains d'adaptation et δ_f est le paramètre de conception choisi qui doit vérifier la condition : $\delta(0) > \delta_f > 1$ et la fonction $f_o(\tilde{z})$ est définie par :

$$\begin{cases} f_o(\tilde{z}) = |\tilde{z}| - \bar{\varepsilon}, \\ |f_o(\tilde{z})| \leq \varepsilon_m \end{cases} \quad (5.33)$$

Dont, $\bar{\varepsilon}, \varepsilon_m > 0$ sont des paramètres choisis suffisamment petit et $\tilde{z} = e - z$ est l'erreur d'observation par l'OGGA.

Notant \tilde{Z} le vecteur d'erreur d'observation par l'OGGA, d'où $\tilde{Z} = E - Z$ et dans ce cas, on aura l'expression (5.34), pour l'utiliser plus tard dans l'étude de la stabilité:

$$\begin{cases} \dot{\tilde{Z}} = (A' - \delta(t)\Delta(t)C)\tilde{Z} + B\dot{s} \\ \tilde{z} = C\tilde{Z} \end{cases} \quad (5.34)$$

Le mécanisme d'adaptation du gain d'observation $\delta(t)$ (équation (5.32)) permet la convergence du gain $\delta(t)$ vers des valeurs minimales si l'erreur d'observation \tilde{z} est à l'intérieure d'une zone bornée. Dans le cas contraire, c'est-à-dire \tilde{z} est à l'extérieur de cette zone, le gain d'observation augmente. Et cet observateur peut délivrer des signaux moins sensibles aux bruits de mesure. Cependant, et comme il sera montré plus tard, le résultat obtenu par l'OGGA est insuffisant, alors, nous proposons d'ajouter d'un deuxième observateur en cascade, qui est le Filtre de Kalman Modifié (FKM).

3.4.2. Filtre de Kalman modifié

Le but de rajouter cet observateur est d'exploiter les propriétés de filtrage du filtre de Kalman. Il est employé pour estimer le vecteur Z , c'est-à-dire, une deuxième estimation du vecteur d'état E .

Le filtre de Kalman modifié proposé est formulé comme suit :

$$\begin{cases} \dot{\hat{E}} = (A' - \delta(t)\Delta(t)C)\hat{E} + K_a (z - \hat{e}) + \delta(t)\Delta(t)e_f \\ \hat{e} = C\hat{E} \end{cases} \quad (5.35)$$

Avec: $\hat{E}^T = [\hat{e} \quad \dot{\hat{e}} \quad \dots \quad \hat{e}^{(n-1)}] \in \mathfrak{R}^n$ est l'estimé de Z par le FKM, e_f est l'erreur de poursuite filtrée d'où : $e_f = H(p)e$ avec $H(p)$: est un filtre passe bas, le vecteur $\Delta(t)$ et le gain d'observation $\delta(t)$ sont définies par (5.31) et (5.32) respectivement et K_a est le gain de Kalman donné par :

$$K_a = P C^T Q_0^{-1} \quad (5.36)$$

Avec, $P = P^T = E \{ E_Z E_Z^T \} > 0$ avec, $E_Z = Z - \hat{E}$, est la solution unique de l'équation de Riccati définie par (5.37):

$$\dot{P} = (A' - \delta(t)\Delta(t)C)P + P(A' - \delta(t)\Delta(t)C)^T - P C^T Q_0^{-1} C P + Q_s \quad (5.37)$$

Avec : $Q_0 = Q_0^T > 0$, $Q_s = Q_s^T > 0$ sont les covariances de vecteur d'état Z et la sortie z respectivement, et la dynamique de E_Z est donnée par :

$$\begin{cases} \dot{E}_Z = (A' - \delta(t)\Delta(t) - K_a C)E_Z + \delta(t)\Delta(t)(e - e_f) \\ e_z = C E_Z \end{cases} \quad (5.38)$$

La Figure 5.2 résume le système d'observation proposé dans cette technique. Dont, l'estimation de vecteur E est faite en deux phases dans la première l'erreur de poursuite (e) est utilisée comme entrée de l'OGGA et la sortie de ce dernier (z) est utilisée comme entrée du FKM. Le résultat final de ce système d'observation (\hat{E}) est l'estimé du vecteur d'état E . Le vecteur estimé (\hat{E}) est utilisé pour la mise en œuvre des lois de commande.

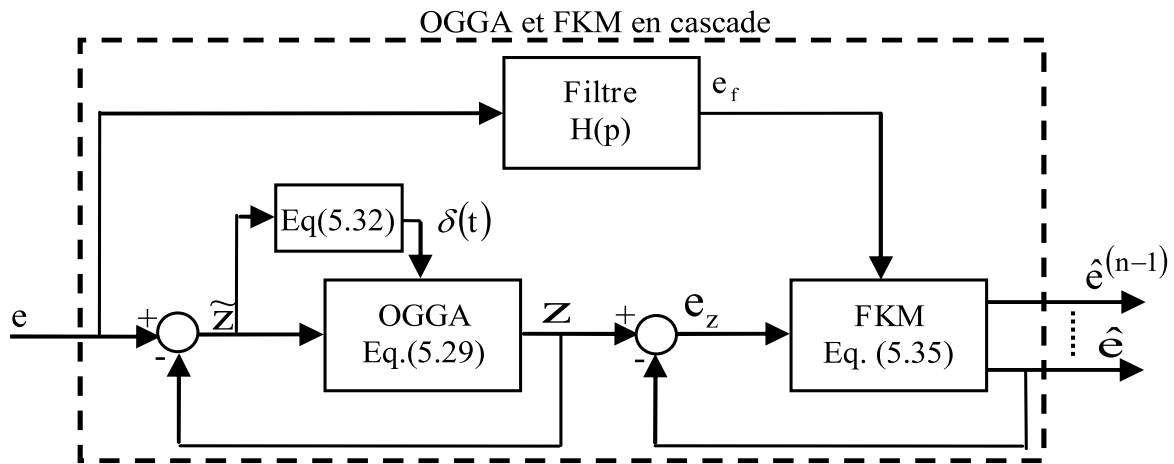


Figure 5.2. Observateur à grand gain adaptatif et filtre de Kalman modifié en cascade.

L'erreur d'observation finale entre le vecteur d'état inconnu (E) et son estimé par le système d'observation ci-dessus (\hat{E}) est notée $\tilde{E} = E - \hat{E}$, d'où sa première dérivée par rapport au temps est comme suit :

$$\begin{cases} \dot{\tilde{E}} = (A - B\Phi_0) \tilde{E} + \delta(t)\Delta(t)C \hat{E} - K_a C E_Z - \delta(t)\Delta(t)e_f + B\dot{s} \\ \tilde{e} = C \tilde{E} \end{cases} \quad (5.39)$$

3.5. Analyse de la stabilité

Une étude de la stabilité de système global, est faite dans cette partie. Pour cela on utilise :

- i). Soient les matrices $Q_1 = Q_1^T > 0$, $Q_2 = Q_2^T > 0$, $Q = Q^T > 0$, dont, Q_1 et Q sont les solutions des équations de Lyapunov suivantes:

$$\begin{cases} Q_1 + A^T Q_1 + Q_1 A = C^T C \\ (A' - \delta(t)\Delta(t)C)^T Q + Q(A' - \delta(t)\Delta(t)C) = -Q_2 \end{cases} \quad (5.40)$$

- ii). Définissant le changement de variable suivant :

$$\bar{Z} = \Delta_d^{-1}(t)\tilde{E} \quad (5.41)$$

Avec $\Delta_d(t)$ est une matrice variable dans le temps définie par :

$$\Delta_d(t) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \delta(t) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \delta^{n-1}(t) \end{bmatrix} \quad (5.42)$$

Et sa dérivée temporelle par :

$$\dot{\Delta}_d^{-1}(t) = -\frac{\dot{\delta}(t)}{\delta(t)} D \Delta_d^{-1}(t) \quad (5.43)$$

Avec :

$$\begin{cases} D = \bar{D} - I_{n \times n} \\ \bar{D} = \text{diag}(1, 2, 3, \dots, n) \end{cases} \quad (5.44)$$

Les relations suivantes sont valables :

- $Q_3 = Q_3^T > 0$, la matrice \bar{D} est la solution de l'équation [Far09] :

$$Q_1 \bar{D} + \bar{D} Q_1 = Q_3 \quad (5.45)$$

- Soit la matrice définie positive Q_4 donnée par :

$$Q_4 = Q_3 - 2Q_1 + C^T C \quad (5.46)$$

- Les constantes ς_1, ς_2 dans la loi d'adaptation de gain d'observation (5.32) sont choisies de tel sorte que :

$$\varsigma_1 \varsigma_2 = \frac{\|Q_1\|_{\min}}{\|Q_4\|} \quad (5.47)$$

Avec $\|\cdot\|_{\min}$ est la norme minimale et $\|\cdot\|$ est la norme maximale des matrices Q_1 et Q_4 respectivement.

Cependant, le changement de variables dans (5.41) implique les cinq propriétés suivantes :

$$\begin{cases} \|\bar{Z}\| \leq \|\tilde{E}\| \\ \|\tilde{E}\| \leq \delta^{n-1}(t)\|\bar{Z}\| \\ \tilde{e} = C\tilde{E} = C\Delta_d(t)\bar{Z} = C\bar{Z} = \bar{z} \\ \Delta_d^{-1}(t)A\Delta_d(t) = \delta(t)A \\ \Delta_d^{-1}(t)\Delta(t) = \bar{\mathcal{G}} \end{cases} \quad (5.48)$$

Avec $\bar{\mathcal{G}}$ est un vecteur choisi comme suit :

$$\bar{\mathcal{G}} = Q_1^{-1} C^T = [\mathcal{g}_1 \quad \mathcal{g}_2 \quad \dots \quad \mathcal{g}_n]^T \in \mathfrak{R}^n \quad (5.49)$$

En utilisant les équations (5.39), (5.41) et (5.43) la dérivé de \bar{Z} sera :

$$\dot{\bar{Z}} = \dot{\Delta}_d^{-1}(t)\tilde{E} + \Delta_d^{-1}(t)\dot{\tilde{E}} = \begin{pmatrix} -\frac{\dot{\delta}(t)}{\delta(t)}D\bar{Z} + \delta(t)A\bar{Z} - \Delta_d^{-1}(t)B\Phi_0\Delta_d(t)\bar{Z} + \delta(t)Q_1^{-1}C^TC\hat{E} \\ -\Delta_d^{-1}(t)K_a C E_z - \delta(t)Q_1^{-1}C^T e_f + \Delta_d^{-1}(t)B\dot{s} \end{pmatrix} \quad (5.50)$$

iii). L'implémentation de la loi de commande dans (5.8) conduit à l'erreur filtrée suivante :

$$\dot{s} = -g(k s - k \tilde{s} - w_1) \quad (5.51)$$

On a :

$$w_1 = \tilde{W}^T \sigma(\hat{V}^T \hat{\chi}) + \hat{W}^T \sigma'(\hat{V}^T \hat{\chi}) \tilde{V}^T \hat{\chi} + w - u^r \quad (5.52)$$

et :

$$\tilde{s} = s - \hat{s} = \Phi \tilde{E} = \Phi \Delta_d \bar{Z} \quad (5.53)$$

• **Hypothèse 5.4.**

- Les poids idéaux sont bornés par $\|W\| \leq W_m$, $\|V\| \leq V_m$. Avec: W_m et V_m sont des constantes positives inconnues.
- $\|w\| \leq \rho$, $|w_1| \leq \frac{c_1}{\delta^{n-1}} \|\tilde{W}\| + \frac{c_2}{\delta^{n-1}} \|\tilde{V}\| + \frac{c_3}{\delta^{n-1}}$, $|e - e_f| \leq c_4$ avec : ρ , c_1, c_2, c_3, c_4 sont des constantes positives inconnues.

• **Théorème 5.1.** Considérons le système non linéaire SISO (5.1), la loi de commande définie par (5.9) (5.10), (5.14), (5.15), (5.16) avec les lois d'adaptation (5.13), l'observateur à

grand gain adaptatif en cascade avec un filtre de Kalman modifié (5.29), (5.35). Supposons que les hypothèses 5.1-5.4 sont satisfaites et les paramètres de conception sont choisis tels que :

$\delta_f > \text{Max}(\beta_{11}, \beta_{12}, \beta_{13})$ et $\kappa > \max(\frac{\beta_4}{\beta_3}, \frac{\beta_6}{\beta_5})$ avec : β_i sont des constantes positives définies

plus tard, alors tous les signaux du système en boucle fermée sont Uniformément Ultiment Bornées (UUB).

• **Preuve du Théorème 5.1.** la méthode directe de Lyapunov est utilisée, pour cela, la fonction candidate est définie comme suit :

$$L = L_1 + L_2 \quad (5.54)$$

Avec :

$$L_1 = \frac{1}{2} \bar{Z}^T Q_1 \bar{Z} + \frac{1}{2} \tilde{Z}^T Q_2 \tilde{Z} \quad (5.55)$$

Et:

$$L_2 = \frac{1}{2} \left(g^{-1} s^2 + \text{tr}(\tilde{W}^T F_W^{-1} \tilde{W}) + \text{tr}(\tilde{V}^T F_V^{-1} \tilde{V}) + \frac{1}{\gamma} \tilde{\rho}^2 \right) \quad (5.56)$$

Commençant par le premier terme de la fonction de Lyapunov:

$$\dot{L}_1 = \frac{1}{2} \dot{\bar{Z}}^T Q_1 \bar{Z} + \frac{1}{2} \bar{Z}^T Q_1 \dot{\bar{Z}} + \frac{1}{2} \dot{\tilde{Z}}^T Q_2 \tilde{Z} + \frac{1}{2} \tilde{Z}^T Q_2 \dot{\tilde{Z}} \quad (5.57)$$

En utilisant les équations (5.32), (5.34), (5.43) et (5.50) et par l'utilisation de $E_z = Z - \hat{E} = \tilde{E} - \tilde{Z}$ et $\hat{E} = E - \tilde{E}$, la fonction \dot{L}_1 devient:

$$\dot{L}_1 = \left(\begin{array}{l} \delta(t) \bar{Z}^T C^T C \left(\frac{\bar{Z}}{2} - \tilde{E} \right) - \frac{1}{2} \bar{Z}^T \left(\delta(t) Q_1 + \frac{\dot{\delta}(t)}{\delta(t)} (-2Q_1 + Q_3) \right) \bar{Z} - \frac{1}{2} \tilde{Z}^T Q_2 \tilde{Z} + \tilde{Z}^T Q_2 B \dot{s} \\ - \bar{Z}^T Q_1 \Delta_d^{-1}(t) B \Phi_0 \Delta_d(t) \bar{Z} + \bar{Z}^T \delta(t) C^T (e - e_f) - \bar{Z}^T Q_1 \Delta_d^{-1}(t) K_a C \Delta_d(t) \bar{Z} \\ + \bar{Z}^T Q_1 \Delta_d^{-1}(t) K_a C \tilde{Z} + \bar{Z}^T Q_1 \Delta_d^{-1}(t) B \dot{s} \end{array} \right) \quad (5.58)$$

En utilisant le fait que :

$$\bar{Z}^T C^T C \left(\frac{\bar{Z}}{2} - \tilde{E} \right) \leq 0 \quad (5.59)$$

Et les équations (5.51)-(5.53), la dernière inégalité et l'hypothèse 5.4, (5.58) devient :

$$\dot{L}_1 \leq \left(\begin{aligned} & -\frac{\delta(t)}{2} \|Q_1\|_{\min} \|\bar{Z}\|^2 - \frac{1}{2} \|Q_2\|_{\min} \|\tilde{Z}\|^2 - \frac{1}{2} \bar{Z}^T \left(\frac{\dot{\delta}(t)}{\delta(t)} (-2Q_1 + Q_3) \right) \bar{Z} + k \|Q_2 B_g\| \|\tilde{Z}\| |s| \\ & + k \|Q_1 \Delta_d^{-1}(t) B_g\| \|\bar{Z}\| |s| + \frac{c_1}{\delta^{n-1}} \|Q_2 B_g\| \|\tilde{W}\| \|\tilde{Z}\| + \frac{c_2}{\delta^{n-1}} \|Q_2 B_g\| \|\tilde{V}\| \|\tilde{Z}\| + \frac{c_3}{\delta^{n-1}} \|Q_2 B_g\| \|\tilde{Z}\| \\ & + \frac{c_1}{\delta^{n-1}} \|Q_1 \Delta_d^{-1}(t) B_g\| \|\tilde{W}\| \|\bar{Z}\| + \frac{c_2}{\delta^{n-1}} \|Q_1 \Delta_d^{-1}(t) B_g\| \|\tilde{V}\| \|\bar{Z}\| \\ & + \left(k \|Q_2 B_g \Phi \Delta_d(t)\| + \|Q_1 \Delta_d^{-1}(t) K_a C\| \right) \|\bar{Z}\| \|\tilde{Z}\| + \left(c_4 \delta(t) + \frac{c_3}{\delta^{n-1}} \|Q_1 \Delta_d^{-1}(t) B_g\| \right) \|\bar{Z}\| \\ & + \left(\|Q_1 \Delta_d^{-1}(t) B \Phi_0 \Delta_d(t)\| + \|Q_1 \Delta_d^{-1}(t) K_a C \Delta_d(t)\| + k \|Q_1 \Delta_d^{-1}(t) B_g \Phi \Delta_d(t)\| \right) \|\bar{Z}\|^2 \end{aligned} \right) \quad (5.60)$$

Implémentant l'ensemble des inégalités suivantes :

$$\left\{ \begin{aligned} & \|Q_2 B_g\| \leq \bar{g} \|Q_2\|, \|Q_1 \Delta_d^{-1}(t) B_g\| \leq \frac{\bar{g}}{\delta^{n-1}} \|Q_2\| \\ & \|Q_2 B_g \Phi \Delta_d(t)\| \leq \bar{g} \delta^{n-1} \|Q_2\| \|\Phi\|, \|Q_1 \Delta_d^{-1}(t) K_a C\| \leq \|Q_1\| \|K_a\| \\ & \|Q_1 \Delta_d^{-1}(t) B \Phi_0 \Delta_d(t)\| \leq \|Q_1\| \|\Phi_0\|, \|Q_1 \Delta_d^{-1}(t) K_a C \Delta_d(t)\| \leq \|Q_1\| \|K_a\| \\ & \|Q_1 \Delta_d^{-1}(t) B_g \Phi \Delta_d(t)\| \leq \bar{g} \|Q_1\| \|\Phi\| \end{aligned} \right. \quad (5.61)$$

dans (5.60), donne :

$$\dot{L}_1 \leq \left(\begin{aligned} & -\left(\frac{\delta(t)}{2} \|Q_1\|_{\min} - \|Q_1\| (\|\Phi_0\| + \|K_a\| + k \bar{g} \|\Phi\|) \right) \|\bar{Z}\|^2 - \frac{1}{2} \|Q_2\|_{\min} \|\tilde{Z}\|^2 \\ & - \frac{1}{2} \bar{Z}^T \left(\frac{\dot{\delta}(t)}{\delta(t)} (-2Q_1 + Q_3) \right) \bar{Z} + k \bar{g} \|Q_2\| \|\tilde{Z}\| |s| + k \frac{\bar{g}}{\delta^{n-1}} \|Q_2\| \|\bar{Z}\| |s| + \frac{c_1}{\delta^{n-1}} \bar{g} \|Q_2\| \|\tilde{W}\| \|\tilde{Z}\| \\ & + \frac{c_2}{\delta^{n-1}} \bar{g} \|Q_2\| \|\tilde{V}\| \|\tilde{Z}\| + \frac{c_3}{\delta^{n-1}} \bar{g} \|Q_2\| \|\tilde{Z}\| + c_1 \frac{\bar{g}}{\delta^{2n-2}} \|Q_2\| \|\tilde{W}\| \|\bar{Z}\| + c_2 \frac{\bar{g}}{\delta^{2n-2}} \|Q_2\| \|\tilde{V}\| \|\bar{Z}\| \\ & + \left(k \bar{g} \delta^{n-1} \|Q_2\| \|\Phi\| + \|Q_1\| \|K_a\| \right) \|\bar{Z}\| \|\tilde{Z}\| + \left(c_4 \delta(t) + c_3 \frac{\bar{g}}{\delta^{2n-2}} \|Q_2\| \right) \|\bar{Z}\| \end{aligned} \right) \quad (5.62)$$

La dérivée de deuxième terme de la fonction de Lyapunov donne :

$$\dot{L}_2 = g^{-1} s \dot{s} + \text{tr} \left(\tilde{W}^T F_W^{-1} \dot{\tilde{W}} \right) + \text{tr} \left(\tilde{V}^T F_V^{-1} \dot{\tilde{V}} \right) + \frac{1}{\gamma} \tilde{\rho} \dot{\tilde{\rho}} \quad (5.63)$$

En remplaçant (5.9) (5.10), (5.14), (5.15), (5.16), les lois d'adaptation (5.13), l'erreur filtrée donnée par (5.51) l'hypothèse 5.4, et la propriété que $\text{tr}(\tilde{W}^T \dot{\tilde{W}}) \leq \|\tilde{W}\| \|\dot{\tilde{W}}\| - \|\tilde{W}\|^2$, l'équation (5.63)

devient:

$$\dot{L}_2 \leq \left(\begin{array}{l} -k s^2 + k \delta^{n-1} \|\Phi\| |s| \|\bar{Z}\| + c_1 \|\Phi\| \|\bar{Z}\| \|\tilde{W}\| + c_2 \|\Phi\| \|\bar{Z}\| \|\tilde{V}\| + c_3 \|\Phi\| \|\bar{Z}\| \\ + \kappa \left(\|\tilde{W}\| W_m + \|\tilde{V}\| V_m - \|\tilde{W}\|^2 - \|\tilde{V}\|^2 \right) \end{array} \right) \quad (5.64)$$

Dans ce cas, \dot{L} devient :

$$\dot{L} \leq \left(\begin{array}{l} -\left(\frac{\delta(t)}{2} \|Q_1\|_{\min} - c_9 \right) \|\bar{Z}\|^2 - \frac{1}{2} \|Q_2\|_{\min} \|\tilde{Z}\|^2 - \frac{1}{2} \bar{Z}^T \left(\frac{\dot{\delta}(t)}{\delta(t)} (-2Q_1 + Q_3) \right) \bar{Z} - k s^2 \\ + 2k \delta^{n-1} c_5 \|\bar{Z}\| |s| + 2c_1 c_5 \|\tilde{W}\| \|\bar{Z}\| + 2c_2 c_5 \|\tilde{V}\| \|\bar{Z}\| + 2k \delta^{n-1} c_6 \|\tilde{Z}\| |s| + 2c_1 c_6 \|\tilde{W}\| \|\tilde{Z}\| \\ + 2c_2 c_6 \|\tilde{V}\| \|\tilde{Z}\| + 2c_3 c_6 \|\tilde{Z}\| + 2\delta^{n-1} c_7 \|\bar{Z}\| \|\tilde{Z}\| + 2c_8 \|\bar{Z}\| \\ + \kappa \left(\|\tilde{W}\| W_m + \|\tilde{V}\| V_m - \|\tilde{W}\|^2 - \|\tilde{V}\|^2 \right) \end{array} \right) \quad (5.65)$$

Avec les paramètres c_5, c_6, c_7, c_8 et c_9 sont donnés par les expressions suivantes :

$$\begin{cases} c_5 = \frac{\bar{g}}{2\delta^{2n-2}} \|Q_2\| + \frac{\|\Phi\|}{2}, & c_6 = \bar{g} \frac{\|Q_2\|}{2\delta^{n-1}}, & c_7 = k \bar{g} \|Q_2\| \frac{\|\Phi\|}{2} + \|Q_1\| \frac{\|K_a\|}{2\delta^{n-1}}, \\ c_8 = c_4 \frac{\delta(t)}{2} + c_3 c_5, & c_9 = \|Q_1\| (\|\Phi_0\| + \|K_a\| + k \bar{g} \|\Phi\|) \end{cases} \quad (5.66)$$

Par l'utilisation des inégalités suivantes :

$$\left\{ \begin{array}{l} 2\|\tilde{Z}\| |s| \leq \frac{\|\tilde{Z}\|^2}{\alpha_1} + \alpha_1 |s|^2, \quad 2\|\bar{Z}\| |s| \leq \frac{\|\bar{Z}\|^2}{\alpha_2} \delta^{n-1} + \frac{\alpha_2}{\delta^{n-1}} |s|^2, \quad 2\|\tilde{W}\| \|\tilde{Z}\| \leq \frac{\|\tilde{Z}\|^2}{\alpha_3} + \alpha_3 \|\tilde{W}\|^2 \\ 2\|\tilde{V}\| \|\tilde{Z}\| \leq \frac{\|\tilde{Z}\|^2}{\alpha_4} + \alpha_4 \|\tilde{V}\|^2, \quad 2c_6 \|\tilde{Z}\| \leq \frac{\|\tilde{Z}\|^2}{\alpha_5} + \alpha_5 c_6^2, \quad 2\|\tilde{W}\| \|\bar{Z}\| \leq \frac{\|\bar{Z}\|^2}{\alpha_6} + \alpha_6 \|\tilde{W}\|^2 \\ 2\|\tilde{V}\| \|\bar{Z}\| \leq \frac{\|\bar{Z}\|^2}{\alpha_7} + \alpha_7 \|\tilde{V}\|^2, \quad 2\|\bar{Z}\| \|\tilde{Z}\| \leq \frac{\|\bar{Z}\|^2}{\alpha_8} \delta^{n-1} + \frac{\alpha_8}{\delta^{n-1}} \|\tilde{Z}\|^2, \quad 2c_8 \|\bar{Z}\| \leq \frac{\|\bar{Z}\|^2}{\alpha_9} + \alpha_9 c_8^2 \\ 2\|\tilde{W}\| W_m \leq \frac{W_m^2}{\alpha_{10}} + \alpha_{10} \|\tilde{W}\|^2, \quad 2\|\tilde{V}\| V_m \leq \frac{V_m^2}{\alpha_{11}} + \alpha_{11} \|\tilde{V}\|^2 \end{array} \right. \quad (5.67)$$

avec α_i sont des constantes positives de conception, on obtient :

$$\dot{L} \leq \left(\begin{array}{l} -k(1-\beta_1) s^2 - \left(\frac{1}{2} \|Q_2\|_{\min} - \beta_2 \right) \|\tilde{Z}\|^2 - \left(\frac{\delta(t)}{2} \|Q_1\|_{\min} - \beta_3 \right) \|\bar{Z}\|^2 \\ - (\beta_3 \kappa - \beta_4) \|\tilde{W}\|^2 - (\beta_5 \kappa - \beta_6) \|\tilde{V}\|^2 - \frac{1}{2} \bar{Z}^T \left(\frac{\dot{\delta}(t)}{\delta(t)} (-2Q_1 + Q_3) \right) \bar{Z} + \beta_8 \end{array} \right) \quad (5.68)$$

Avec :

$$\left\{ \begin{array}{l} \beta_1 = \alpha_1 c_6 \delta^{n-1} + \alpha_2 c_5, \beta_2 = c_6 \left(k \frac{\delta^{n-1}}{\alpha_1} + \frac{c_1}{\alpha_3} + \frac{c_2}{\alpha_4} + \frac{c_3}{\alpha_5} \right) + \alpha_8 c_7, \\ \beta_3 = \frac{kc_5}{\alpha_2} + \frac{c_7}{\alpha_8} + \frac{1}{\delta^{2n-2}} \left(c_9 + \frac{c_1 c_5}{\alpha_6} + \frac{c_2 c_5}{\alpha_7} + \frac{1}{\alpha_9} \right), \beta_4 = 1 - \frac{\alpha_{10}}{2}, \\ \beta_5 = \alpha_3 c_1 c_6 + c_1 c_5 \alpha_6, \beta_6 = 1 - \frac{\alpha_{11}}{2}, \beta_7 = \alpha_4 c_2 c_6 + \alpha_7 c_2 c_5, \\ \beta_8 = \alpha_5 c_3 c_6^2 + \alpha_9 c_8^2 + \frac{\kappa}{2} \left(\frac{W_m^2}{\alpha_{10}} + \frac{V_m^2}{\alpha_{11}} \right). \end{array} \right. \quad (5.69)$$

En utilisant la définition de la matrice Q_4 , dans l'équation (5.46), (5.68) devient :

$$\dot{L} \leq \left(\begin{array}{l} -k(1-\beta_1)s^2 - \left(\frac{1}{2} \|Q_2\|_{\min} - \beta_2 \right) \|\tilde{Z}\|^2 - \left(\frac{\delta(t)}{2} \|Q_1\|_{\min} - \delta^{2n-2} \beta_3 \right) \|\bar{Z}\|^2 \\ - (\beta_3 \kappa - \beta_4) \|\tilde{W}\|^2 - (\beta_5 \kappa - \beta_6) \|\tilde{V}\|^2 - \frac{1}{2} \frac{\dot{\delta}(t)}{\delta(t)} \bar{Z}^T Q_4 \bar{Z} + \frac{1}{2} \frac{\dot{\delta}(t)}{\delta(t)} \bar{Z}^2 + \beta_8 \end{array} \right) \quad (5.70)$$

L'implémentation de la loi d'adaptation pour le gain d'observation donné par (5.32) dans (5.70), donne :

$$\dot{L} \leq \left(\begin{array}{l} -k(1-\beta_1)s^2 - \left(\frac{1}{2} \|Q_2\|_{\min} - \beta_2 \right) \|\tilde{Z}\|^2 - \left(\frac{\delta(t)}{2} \|Q_1\|_{\min} - \delta^{2n-2} \beta_3 \right) \|\bar{Z}\|^2 \\ (\beta_3 \kappa - \beta_4) \|\tilde{W}\|^2 - (\beta_5 \kappa - \beta_6) \|\tilde{V}\|^2 - \frac{\zeta_1}{2} f_0(\tilde{z}) \bar{Z}^T Q_4 \bar{Z} - \frac{\zeta_1 \zeta_2}{2} \left(\frac{\delta_f}{\delta(t)} - 1 \right) \bar{Z}^T Q_4 \bar{Z} \\ + \frac{\zeta_1}{2} (|\tilde{z}| - \bar{\varepsilon}) \bar{Z}^2 - \frac{\zeta_1 \zeta_2}{2} \left(1 - \frac{\delta_f}{\delta(t)} \right) \bar{Z}^2 + \beta_8 \end{array} \right) \quad (5.71)$$

En utilisant (5.47), on obtient :

$$\dot{L} \leq \left(\begin{array}{l} -k(1-\beta_1)s^2 - \left(\frac{1}{2} \|Q_2\|_{\min} - \beta_2 \right) \|\tilde{Z}\|^2 - \left(\frac{1}{2} \left(\delta(t) + \frac{\delta_f}{\delta(t)} - 1 \right) \|Q_1\|_{\min} - \delta^{2n-2} \beta_3 \right) \|\bar{Z}\|^2 \\ (\beta_3 \kappa - \beta_4) \|\tilde{W}\|^2 - (\beta_5 \kappa - \beta_6) \|\tilde{V}\|^2 - \frac{\zeta_1}{2} f_0(\tilde{z}) \bar{Z}^T Q_4 \bar{Z} + \frac{\zeta_1}{2} (|\tilde{z}| - \bar{\varepsilon}) \bar{Z}^2 \\ - \frac{\zeta_1 \zeta_2}{2} \left(1 - \frac{\delta_f}{\delta(t)} \right) \bar{Z}^2 + \beta_8 \end{array} \right) \quad (5.72)$$

Nous avons :

$$\forall t, \delta(t) > \delta_f > 1 \Rightarrow \left\{ \begin{array}{l} \delta(t) > \delta(t) + \frac{\delta_f}{\delta(t)} - 1 > \delta(t) + \frac{1}{\delta(t)} - 1 > \delta_f + \frac{1}{\delta(t)} - 1 > 0 \\ 0 < 1 - \frac{\delta_f}{\delta(t)} < 1 - \frac{1}{\delta(t)} \end{array} \right. \quad (5.73)$$

Alors:

$$\dot{L} \leq \left(-k(1-\beta_1)s^2 - \left(\frac{1}{2}\|Q_2\|_{\min} - \beta_2 \right) \|\tilde{Z}\|^2 - \frac{\delta^{n-1}}{2} \left(\frac{\beta_9\|Q_1\|_{\min}}{\delta^{2n-2}} - \beta_{10} \right) \|\tilde{E}\|^2 - (\beta_3\kappa - \beta_4) \|\tilde{W}\|^2 - (\beta_5\kappa - \beta_6) \|\tilde{V}\|^2 + \beta_8 \right) \quad (5.74)$$

Avec :

$$\beta_9 = \left(\delta_f - \left(\frac{1}{\|Q_4\|} - 1 \right) \left(\frac{1}{\delta(t)} - 1 \right) \right), \beta_{10} = 2\beta_3 + \frac{\varepsilon_m \varsigma_1 (\|Q_4\| + 1)}{\delta^{2n-2}}. \quad (5.75)$$

En choisissant, $\delta_f > \text{Max}(\beta_{11}, \beta_{12}, \beta_{13})$ et $\kappa > \max\left(\frac{\beta_4}{\beta_3}, \frac{\beta_6}{\beta_5}\right)$ avec :

$$\left\{ \begin{array}{l} \beta_{11} = \left(\frac{2 - \alpha_1 \bar{g} \|Q_2\| - \alpha_2 \|\Phi\|}{\alpha_2 \|Q_2\|} \right)^{\frac{1}{2n-2}}, \\ \beta_{12} = \left(\frac{\frac{c_1}{\alpha_3} + \frac{c_2}{\alpha_4} + \frac{c_3}{\alpha_5} + \bar{g} \|Q_2\| + \alpha_8 \|Q_1\| \|K_a\|}{\|Q_2\|_{\min} - \alpha_1 \bar{g} \frac{\|Q_2\|}{k} - \alpha_8 \bar{g} k \|\Phi\| \|Q_2\|} \right)^{\frac{1}{n-1}}, \\ \beta_{13} = \frac{2 \left(\frac{kc_5}{\alpha_2} + \frac{c_7}{\alpha_8} + c_9 + \frac{c_1 c_5}{\alpha_6} + \frac{c_2 c_5}{\alpha_7} + \frac{1}{\alpha_9} \right) + \frac{1}{\|Q_4\|} - 1}{\|Q_1\|_{\min}}. \end{array} \right. \quad (5.76)$$

nous pouvons garantir que \dot{L} est négatif si $s, \tilde{E}, \tilde{Z}, \tilde{W}, \tilde{V}$ sont en dehors des ensembles compacts $\Omega_s, \Omega_{\tilde{E}}, \Omega_{\tilde{Z}}, \Omega_{\tilde{W}}, \Omega_{\tilde{V}}$ définis comme suit :

$$\left\{ \begin{array}{l} \Omega_s = \left\{ s / \|s\| \leq \sqrt{\frac{\beta_8}{k(1-\beta_1)}} \right\}, \quad \Omega_{\tilde{E}} = \left\{ \tilde{E} / \|\tilde{E}\| \leq \sqrt{\frac{2\beta_8 \delta^{n-1}}{(\beta_9\|Q_1\|_{\min} - \delta^{2n-2} \beta_{10})}} \right\} \\ \Omega_{\tilde{Z}} = \left\{ \tilde{Z} / \|\tilde{Z}\| \leq \sqrt{\frac{2\beta_8}{\|Q_2\|_{\min} - 2\beta_2}} \right\} \\ \Omega_{\tilde{W}} = \left\{ \tilde{W} / \|\tilde{W}\| \leq \sqrt{\frac{\beta_8}{\beta_3\kappa - \beta_4}} \right\}, \quad \Omega_{\tilde{V}} = \left\{ \tilde{V} / \|\tilde{V}\| \leq \sqrt{\frac{\beta_8}{\beta_5\kappa - \beta_6}} \right\} \end{array} \right. \quad (5.77)$$

De plus, le rayon de ces ensembles peut être rendu arbitrairement petit si k, κ sont choisis suffisamment grands.

4. Résultats de Simulation

La validation de schéma de commande proposé est faite dans cette section par simulation sur le modèle dynamique d'un pendule inversé (Figure 5.3). Ce système est de la même classe des systèmes étudiés. Il est de deuxième ordre dont les fonctions $f(X)$ et $g(X)$ sont définies comme suit :

$$\begin{cases} f(X) = \frac{(g \sin(x_1) - m_p l x_2^2 \cos(x_1) \sin(x_1)) / (m_p + m_c)}{l(4/3 - m_p \cos^2(x_1)) / (m_p + m_c)} \\ g(X) = \frac{\cos(x_1) / (m_p + m_c)}{l(4/3 - m_p \cos^2(x_1)) / (m_p + m_c)} \end{cases} \quad (78)$$

Les variables d'états sont l'orientation $x_1 = \theta$ et la vitesse angulaire $x_2 = \dot{\theta}$ respectivement et $g=9.8m/s^2$, $m_c=1kg$, $m_p=0.1kg$, $l=0.5m$.

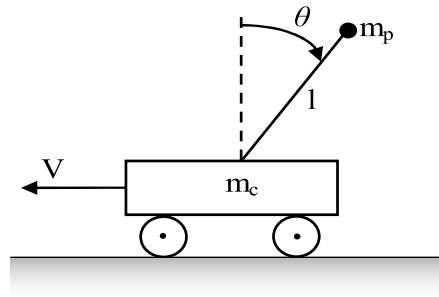


Figure 5.3. Pendule inversé.

Un RNA à une seule couche cachée (voir la Figure 4.3) est utilisé dans le schéma de commande pour estimer la commande équivalente dans (4.16). Ce RNA est composé d'un seul neurone de sortie, trois neurones cachés, cinq neurones d'entrée dont, le vecteur d'entrée est choisi comme suit : $\hat{\chi}_1 = [\ddot{y}_{di} \quad \dot{y}_{di} \quad y_{di} \quad \dot{e} \quad \hat{e}]^T$ et les poids sont initialisés à zéro.

Le schéma de commande global contenant le système d'observation (Observateur à grand gain adaptatif en cascade avec le filtre de Kalman modifié) est donné par la Figure 5.4.

Les bruits de mesure (bruits des capteurs) sont supposés des bruits blanc gaussiens [Bou11, Nag12, Oue12, San11, Sef12, Sti11, Ton12] à moyenne nulle et de variance Var [Bou11, Oue12, Ton12]. Dans notre application la valeur de Var est donnée par : $Var = 0.01$.

Les paramètres de simulation pour le contrôleur adaptatif et le système d'observation proposé sont donnés sur les tableaux 5.1 et 5.2 respectivement.

Une étude comparative entre la technique proposée et deux autres techniques est effectuée pour monter les performances de la technique proposée. Dans la première méthode l'OGGA proposé est utilisé pour estimer E, dont les signaux produises sont filtrés par un filtre passe bas et

les signaux résultants sont utilisés pour la mise en œuvre de la commande. Dans la deuxième méthode, l'observateur proposé dans [San11] est implémenté pour atteindre le même objectif.

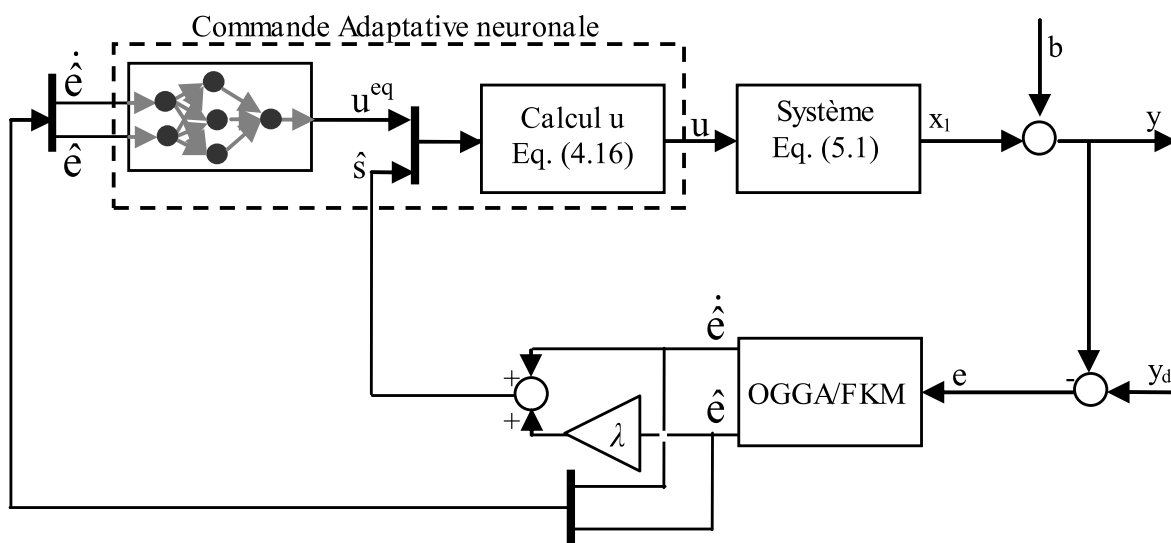


Figure 5.4. Commande adaptative neuronale avec observateur à grand gain adaptative et filtre de Kalman modifié en cascade incertains (système de 2^{ième} ordre).

Tableau 5.1. Paramètres de la commande adaptative.

| Symbol | Valeur | Symbol | Valeur |
|-----------|--------|---------------|--------------|
| λ | 8 | n_s | 1 |
| K | 10 | F_w | $50 I_{3,3}$ |
| κ | 0.01 | F_v | $50 I_{2,2}$ |
| n_e | 2 | ε | 0.1 |
| n_c | 3 | γ | 0.001 |

Tableau 5.2. Paramètres d'OGGA et FKM en cascade

| Symbol | Valeur | Symbol | Valeur |
|-------------|---------------------------|---------------|--------------------|
| $\delta(0)$ | 40 | $P(0)$ | $\text{diag}(2,2)$ |
| $\Delta(0)$ | [40,8000] | ς_1 | 10 |
| Q_o | 0.1 | ς_2 | 0.2 |
| Q_s | $0.01 * \text{diag}(2,2)$ | δ_f | 20 |

Les résultats de simulation sont donnés sur les figures suivantes:

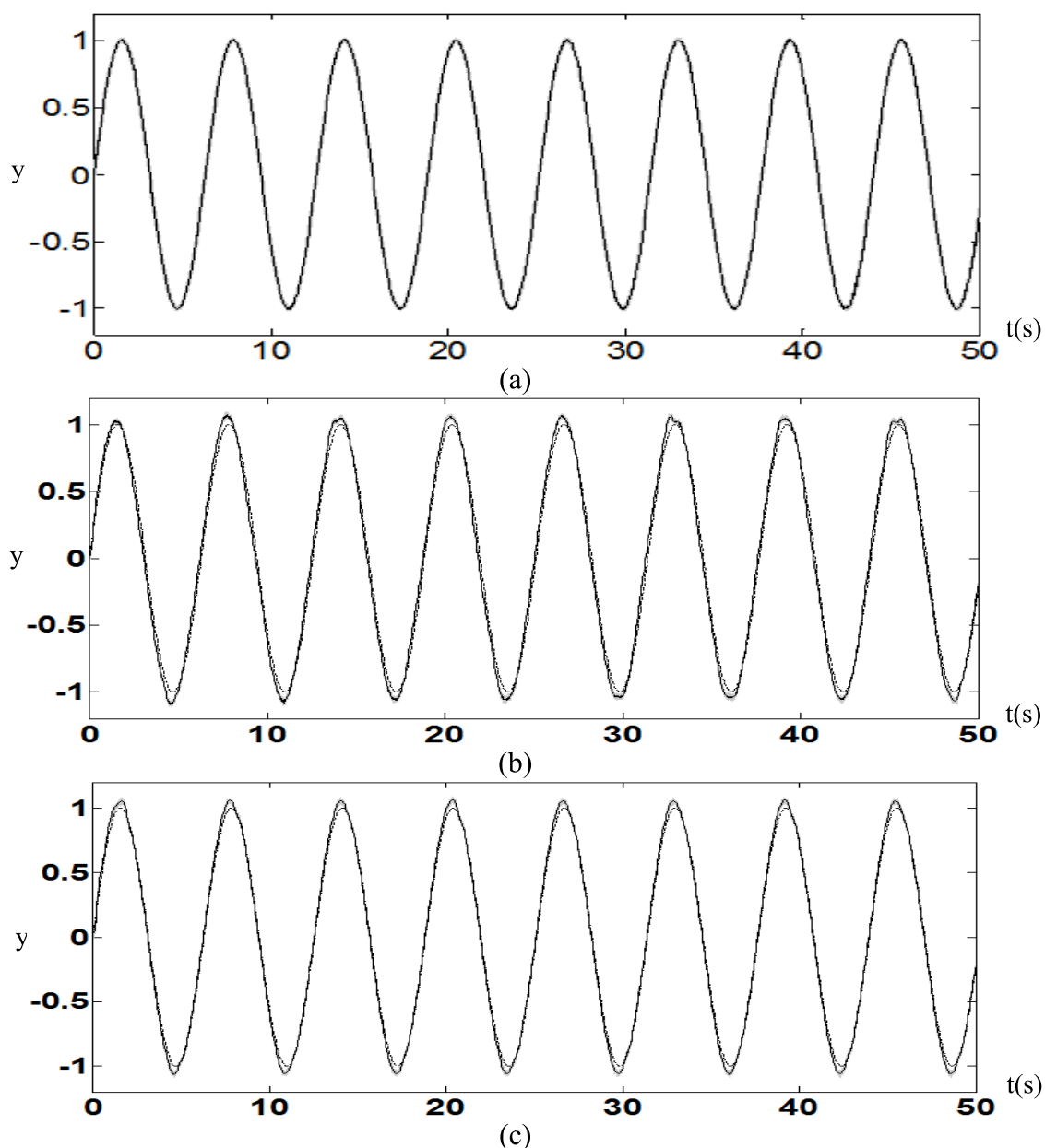
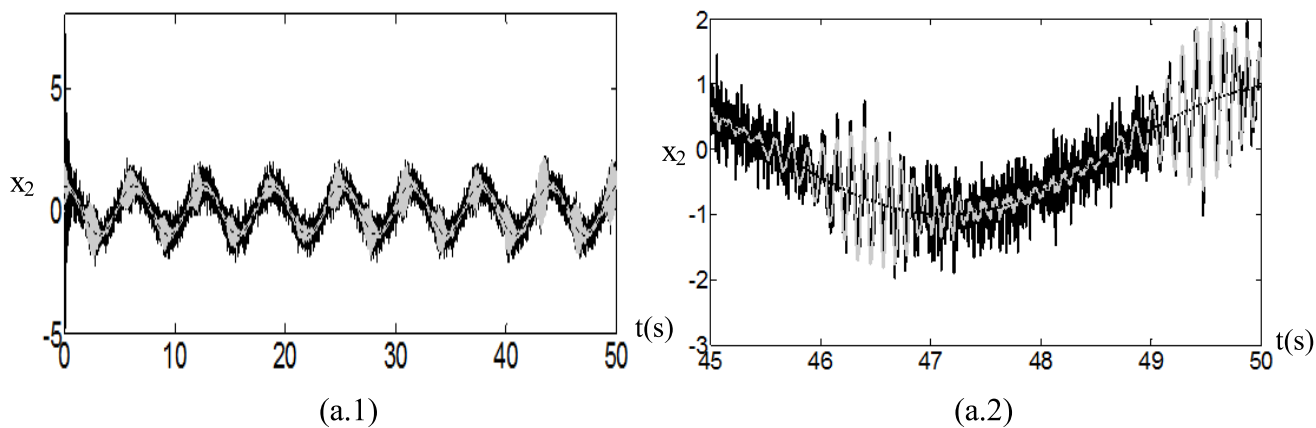


Figure 5.5. Sortie de système (- - : trajectoire désirée, — : trajectoire réelle, — : trajectoire estimée). ((a): OGGA en cascade avec un filtre passe bas simple, (b): [San11], (c): AGGO en cascade avec FKM).



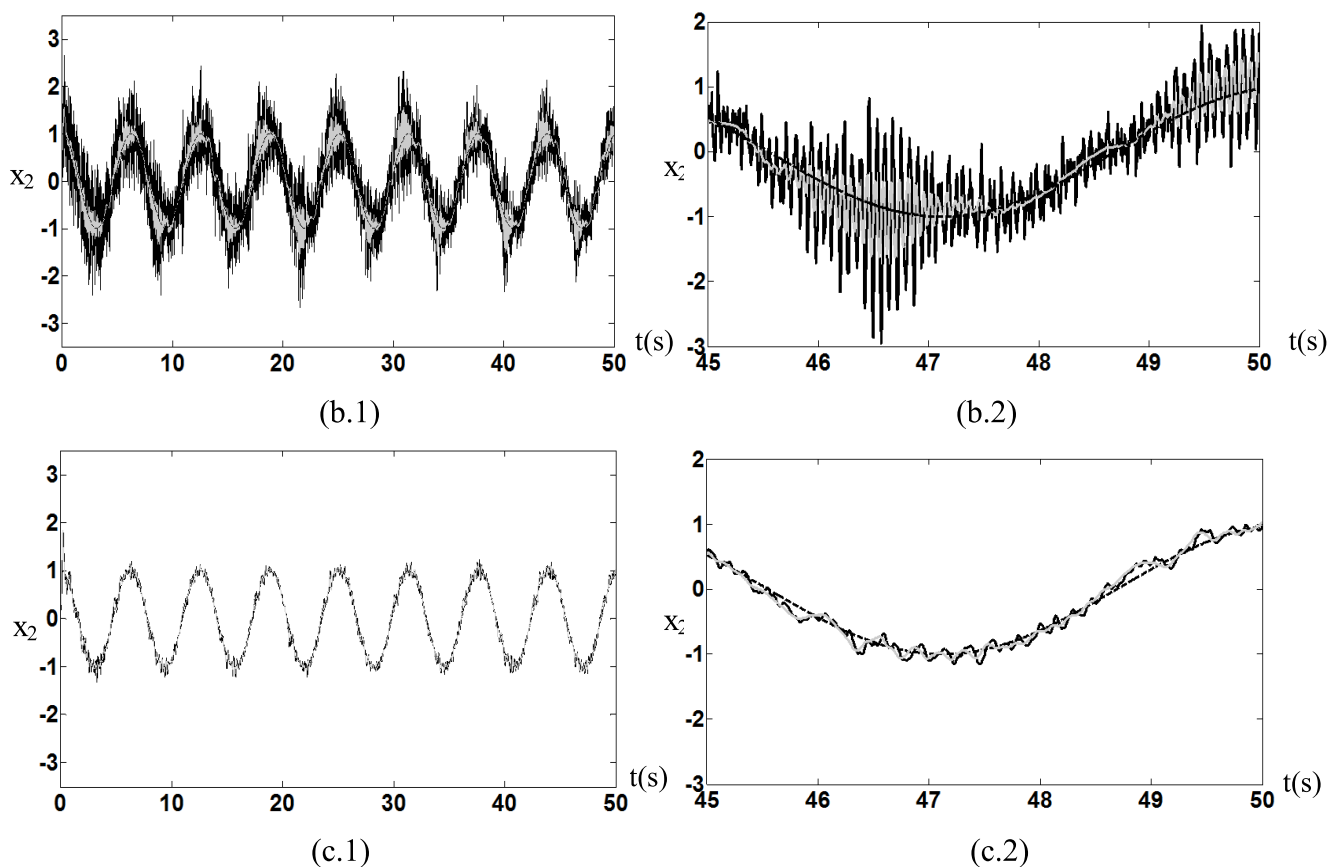
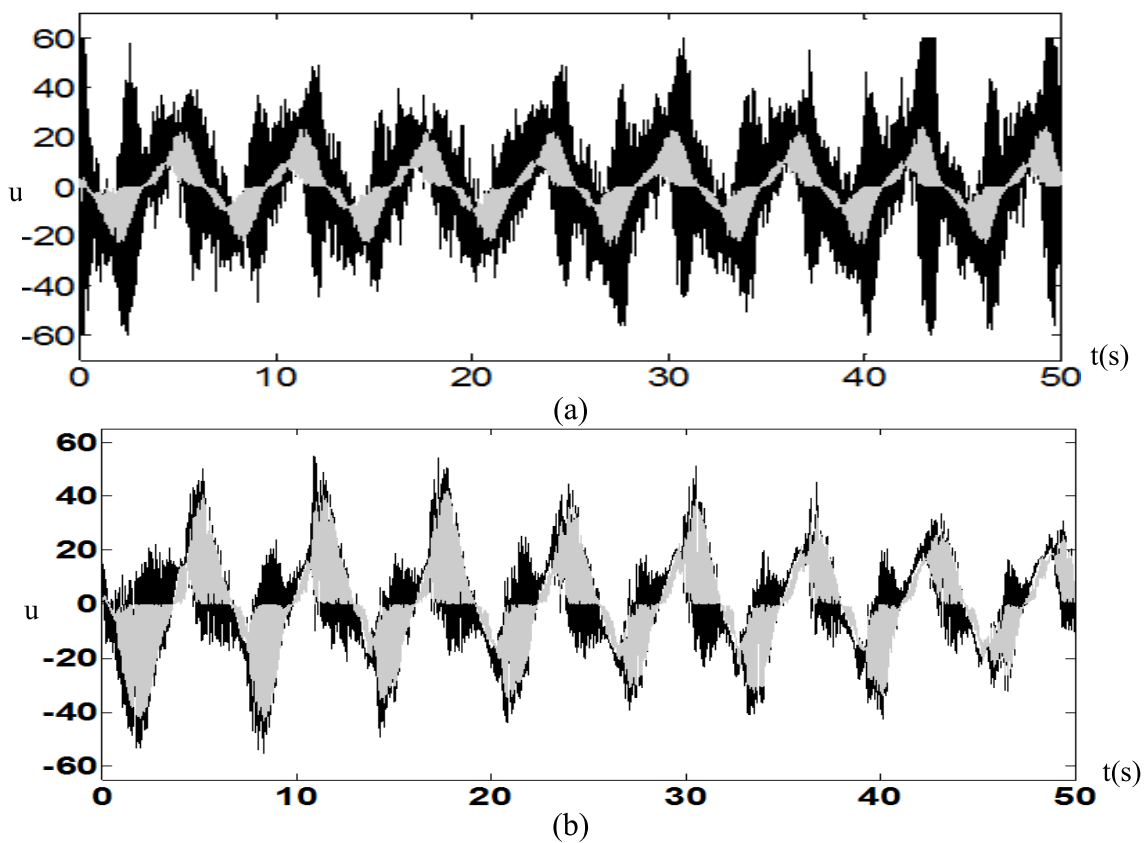


Figure 5.6. vitesse angulaire (--- : trajectoire désirée, — : trajectoire réelle, — : trajectoire estimée). ((a): OGGA en cascade avec un filtre passe bas simple, (b): [San11], (c): AGGO en cascade avec FKM).



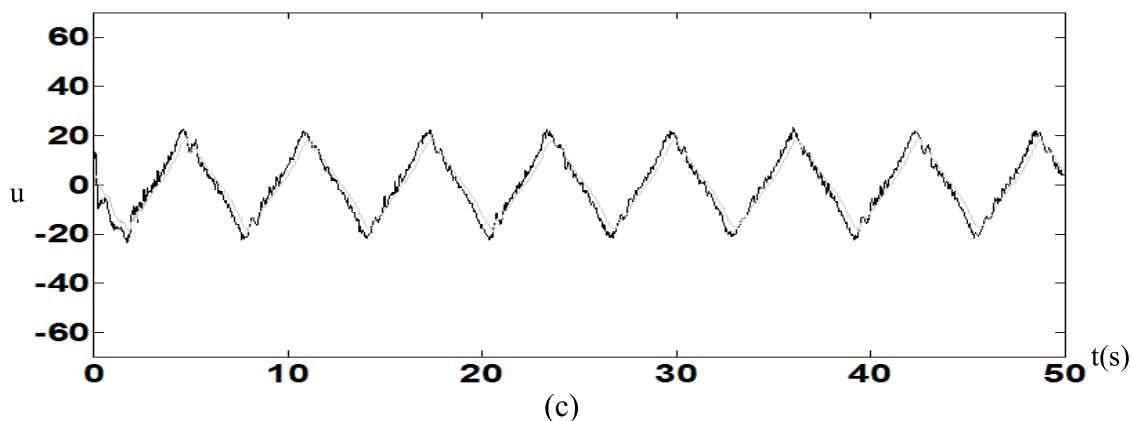


Figure 5.7. Signal de commande (—: commande adaptative u , —: commande u). ((a): OGGA en cascade avec un filtre passe bas simple, (b): [San11], (c): AGGO en cascade avec FKM).

La Figure 5.5 montre les trajectoires réelles et estimées par les trois techniques. Il est clair que la poursuite est bonne dans tous les cas.

Comparant les résultats des vitesses angulaires, on constate une réduction importante du phénomène indésirable (variance et l'amplitude des signaux) est obtenue par la technique utilisée (Figure 5.6.c) par rapport aux deux autres méthodes (Figure 5.6.a et Figure 5.6.b). En plus les signaux obtenus par la technique proposée sont les moyennes des signaux obtenus par les deux autres méthodes.

Les Figures 5.7.a et 5.7.b montrent l'inconvénient de l'utilisation de ces deux techniques, dont, le signal de commande est trop bruité et son amplitude est plus élevée par contre l'utilisation de système d'observation proposé donne un signal de commande plus lisse avec une amplitude moins grand Figure 5.7.c).

5. Conclusion

Dans ce chapitre, on a proposé un schéma de commande adaptatif neuronale avec un observateur à grand gain adaptatif (OGGA) et un filtre de Kalman modifié (FKM) en cascade pour une classe des systèmes non linéaires incertains. D'une manière similaire au quatrième chapitre, la commande équivalente inconnue était approximée par un réseau de neurones artificiel composé de trois couches. La différence principale se réside dans la technique utilisée pour l'observation des variables d'états. Un observateur à grand gain adaptatif en cascade avec un filtre de Kalman modifié pour reconstruire des signaux insensible aux bruits de mesure. La démonstration de la stabilité, et les lois d'adaptation des poids sont déterminées utilisant la

méthode directe de Lyapunov. Des simulations et une étude comparative avec d'autres observateurs ont été effectuées sur le modèle dynamique d'un pendule inversé pour montrer les performances de la technique proposée. Selon les résultats, on constate la minimisation remarquable de la sensibilité aux bruits de mesure utilisant OGGA et FKM en cascade.

Conclusion générale

Conclusion Générale

Cette thèse est consacrée au développement des nouvelles techniques de commande adaptative par l'approximateur universels qui sont les Réseaux de Neurones Artificiels (RNA) avec et sans observateur d'état.

L'objectif principal de chaque stratégie de commande développée par l'automaticien est d'assurer la stabilité de système bouclé et sa robustesse vis-à-vis les perturbations et les incertitudes. Cependant, ceci est impossible si la technique de commande utilisée est basée sur le modèle dynamique de système à commander. En profitant des propriétés d'approximation universelle des RNA, la commande adaptative neuronale peut atteindre les performances désirées même à la présence des incertitudes. Dans ce contexte, des techniques de commande adaptative, stables, robustes et qui ne nécessitent aucune information sur le modèle dynamique ont été proposées dans cette thèse. Tous les schémas développés sont de type auto-ajustable, où les termes inconnus dans les lois de commandes ont été estimés en ligne par un mécanisme d'adaptation. De plus, l'utilisation des observateurs d'états conventionnel tels que l'observateur à grand gain et à mode glissant introduit l'amplification des bruits de mesures réduisant ainsi les performances de commande et la dégradation des actionneurs. Pour cela, et dans le but de minimiser la sensibilité aux bruits, nous avons proposé et validé de nouvelles structures d'observateur permettant la réduction des bruits de mesure et d'améliorer les performances de commande. Pour chaque schéma, la méthode directe de Lyapunov a été utilisée pour l'étude de la stabilité de système bouclé.

Dans le premier chapitre, une description générale de l'approximateur universel RNA est présentée. Les étapes les plus importantes de son développement et les règles d'apprentissage sont présentées. Les RNAs sont non linéaires, et selon le théorème d'approximation universel, ils sont capables d'approximer n'importe quelle fonction inconnue. En automatique cette propriété est utilisée dans plusieurs schémas de commande adaptative et d'identification (directe et inverse) où, l'incertitude de modèle dynamique est compensée et l'algorithme de rétro-propagation est utilisé pour l'apprentissage des poids.

Les notions et les formules nécessaires pour l'étude de stabilité des systèmes non linéaires sont introduites au début du deuxième chapitre. Par la suite, trois types de commandes sont présentés. La première commande est synthétisée en se basant sur la technique backstepping.

Cette technique permet la construction récursive de la fonction de Lyapunov. Pour cela, le système est décomposé en plusieurs sous-systèmes commandés en série par des commandes virtuelles, afin d'obtenir la commande réelle. La loi de commande résultante, n'est pas robuste car elle est basée sur la connaissance parfaite du modèle dynamique du système. La deuxième technique est la commande par mode glissant. Contrairement au backstepping, la commande par mode glissant présente une robustesse qui dépend de l'amplitude de terme discontinu. La troisième technique est la commande adaptative. On a constaté l'avantage et l'intérêt de cette commande par rapport aux deux autres et les raisons de son utilisation dans les schémas de commandes adaptative standards.

Le troisième chapitre comporte un schéma de contrôle adaptatif proposé pour un système sous-actionné : quadrotor. L'approche proposée passe par trois étapes : dans la première étape, des commandes virtuelles ont été ajoutées pour rendre le gain de commande inversible. Un mécanisme d'adaptation est dérivé à partir de l'étude de la stabilité au sens de Lyapunov pour estimer en ligne les commandes virtuelles et la fonction non linéaire inconnue du modèle dynamique dans la deuxième étape. La dernière étape consiste la proposition des contraintes holonomes basées sur la technique de linéarisation à partir des relations physiques du système afin de pouvoir commander toutes les sorties en utilisant un nombre inférieur des entrées. Les résultats des simulations numériques montrent la validité et les bonnes performances de la technique proposée.

Dans le quatrième chapitre, on a traité le problème de l'amplification des bruits de mesure causée par les observateurs d'état conventionnels. La classe des systèmes étudiée est multivariable (MIMO). Un RNA composé de trois couches a été implémenté dans le schéma de commande adaptatif pour approximer en ligne la commande équivalente inconnue. La notion d'adaptation par l'approximateur intelligent est introduite à l'observateur pour générer le terme de correction. La méthode directe de Lyapunov a été utilisée pour analyser la stabilité et étudier la bornitude de tous les signaux de système bouclé. Le quadrotor a été le modèle d'application de cette technique, où des nouvelles commandes virtuelles et contraintes holonomes ont été proposées afin de rendre le quadrotor de la même classe étudiée. Les résultats de simulations ont montrés que l'observateur proposé a réduit considérablement les bruits de mesure.

Une autre commande adaptative neuronale avec observateur à grand gain adaptatif et filtre de Kalman modifié en cascade, pour une classe des systèmes non linéaires incertains est développée dans le cinquième chapitre. Le RNA est utilisé pour estimer la commande équivalente inconnue. L'estimation des variables d'états est effectuée en deux phases : dans la

première la sortie de système a été utilisée comme entrée à l'Observateur à Grand Gain Adaptatif (OGGA) et dans la deuxième, la sortie de ce dernier a été utilisée comme entrée du Filtre de Kalman Modifié (FKM). L'objectif de cette proposition est de minimiser la sensibilité aux bruits de mesure. Comme dans les autres chapitres, la stabilité de système global et la bornitude de tous les signaux ont été montrées utilisant la méthode directe de Lyapunov. Les performances de la technique proposées ont été testées par simulation sur un pendule inversé, de plus une étude comparative avec d'autres types d'observateurs a été effectuée. Nous avons constaté une minimisation remarquable de phénomène indésirable si la technique proposée est utilisée comparant aux autres observateurs.

Les travaux réalisés dans cette thèse, nous conduit à l'ensemble des perspectives suivantes :

- Modifier les approches proposées pour qu'elles soient applicables à d'autres classes des systèmes telles que : les systèmes non affine à la commande,
- Elargir l'étude réalisée pour le cas où le gain de commande est variable dans le temps et son signe est inconnu,
- Développer d'autre techniques pour commander les systèmes sous actionnés et utiliser les approximateurs universels dans le schéma de commande,
- Travailler sur la conception des observateurs adaptatifs intelligents pour résoudre d'autres types de problèmes,
- Développement de nouvelles structures des approximateurs universels,
- Validation des techniques développées dans cette thèse expérimentalement,

Références bibliographiques

Bibliographie

- [Abe11] D. Abeywardena et al: A Virtual Odometer for a Quadrotor Micro Aerial Vehicle. Proc Australasian Conference on Robotics and Automation, Australia pp1-8, 2011.
- [Adi07] P. Adigbli. Nonlinear Attitude and Position Control of a Micro Quadrotor using Sliding Mode and Backstepping Techniques. 3RD US-European Competition and Workshop on MAVS & EMAV Conference and Flight Competition, pp 17-21, France Sep 2007.
- [Ahr09] J. H. Ahrensa, H.K. Khalil. High-gain observers in the presence of measurement noise: A switched-gain approach, *Automatica*, Vol.45, No.4, pp 936–943, 2009.
- [Ais09] A. G. Aissaoui, H. Abid et M. Abid. Robust Fuzzy sliding mode controller design for motor driver, *Acta Electrotechnica et Informatica* Vol. 9, No. 2, pp 64–71, 2009.
- [Ala09] M. S. Alaoui : Commande et Observateur par modes de glissants d'un système de pompage et un bras manipulateur. Thèse de Doctorat, université de Sidi Mohammed Ben Abdellah, 2009.
- [Ale12] K. Alexis *et al.* Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory and Applications*, Vol.6, No.12, pp. 1812–1827, 2012.
- [Bal12] J.E.Gomez-Balderas *et al* : Vision Based Tracking for a Quadrotor Using Vanishing Points. *Journal of Intelligent & Robotic Systems*, Vol.65, No.4, pp 361-371, 2012.
- [Bat11] P. Batista et al: Partial attitude and rate gyro bias estimation: observability analysis, filter design, and performance evaluation. *International Journal of Control*, Vol.84, No.5, pp 895–903, 2011.
- [Bec05] Mustafa Resa Becan : Sliding Mode Control with Fuzzy Boundary Layer to Air-Air Interception Problem. *World Academy of Science, Engineering and Technology*, pp. 108-111, 2005.
- [Ben00] A.R.Benaskeur : Aspects de l'application du Backstepping adaptatif à la commande décentralisée des systèmes non linéaires. Thèse de Doctorat, université de Laval, 2000.
- [Ben08] A. Benallegue, A. Mokhtari and L. Fridman: High-order sliding-mode observer for a quadrotor UAV. *Int. J. Robust Nonlinear Control*. Vol.18, No4-5, pp 427-440, 2008.
- [Bod11] L. Bodizsa, B. Srinivasanb, D. Bonvina. On the design of integral observers for

- unbiased output estimation in the presence of uncertainty. *Journal of Process Control* Vol.21, pp 379–390,2011.
- [Boi10] N.Boizot, E.Busvelle and J.P.Gauthier. An adaptive high-gain observer for nonlinear systems. *Automatica*, Vol.46, pp 1483-1488, 2010.
- [Bou00] R. Boukazzoula. *Commande floue d’une classe de systèmes non linéaires : application au problème de suivi de trajectoire*, Thèse de doctorat, Université de Savoie, Mars 2000.
- [Bou04] S.Bouaddallah, A.Noth and R.Siegwart : PID vs. LQ control Techniques Applied to an Indoor Micro Quadrotor. In Proc IEEE international conference on intelligent Robots and systems, Japan, 2004.
- [Bou07] S.Bouabdallah : *Design and control of a Miniature Quadrotors*. Thèse de doctorat, Ecole Polytechnique Fédérale de Lausanne, Feb 2007.
- [Bou10] **H. Boudjedir**. *Commande adaptative par les réseaux de neurones artificiels des quadrotors*. Mémoire de magister, université de jijel, algérie, 2010.
- [Bou11a] H.Bouadi *et al.* Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking. *IEEE 12th IntSym CINTI*, pp 449 – 455, 2011.
- [Bou11b] O.Bouhali et **H. Boudjedir** (2011). Neural Network control with neuro sliding mode observer applied to Quadrotor Helicopter,” In Proc INISTA’11, pp 24-28, Istanbul, Turkey.
- [Bou12a] **H. Boudjedir**, F. Yacef, O. Bouhali and N. Rizoug, “A new direct adaptive Neural Network Control Applied to Quadrotor Helicopter”. *International Conference on Industrial Engineering and Manufacturing ICIEM’12*, May 06-07, 2012, Batna, Algeria.
- [Bou12b] **H. Boudjedir**, F. Yacef, O. Bouhali and N. Rizoug, “ Dual neural network for adaptive sliding mode control of quadrotor helicopter stabilization”. *International Journal of Informations Sciences and Techniques IJIST*. Vol.2, No.4, pp1-14.
- [Bou14] **H. Boudjedir**, O. Bouhali et N.Rizoug : Adaptive neural network control based on neural observer for quadrotor unmanned aerial vehicle. *Advanced Robotics*, 28:17, 1151-1164, DOI 10.1080/01691864.2014.913498 (**impact Factor 0.562**).
- [Bro92] Brown, R.H., Ruchti, T.L., and Feng, X. Artificial neural network identification of partially known dynamic nonlinear systems. *Procs. of the 32nd IEEE Conference on Decision and Control*, Vol. 4, pp. 3694-3699, 1992.
- [Buh86] H. Buhler : *Réglage par mode de glissement*. Presses polytechniques Rommandes,

- 1986.
- [Bul97] E. Bullinger and F.Allgower (1997). An Adaptive High-Gain Observer for Nonlinear Systems. Proc IEEE, the 36th Conference on decision and control USA.
- [Bur01] R.S.Burns (2001). Advanced control engineering,” Butterworth Heinemann.
- [Cas04] P.Castillo, A.Dzul, and R.Lozano: Real-Time Stabilization and Tracking of a Four-Rotor Mini Rotorcraft. IEEE Trans on control systems technology, VOL. 12, No. 4, July 2004.
- [Cha96] W. Chatlatanagulaci. Backstepping intelligent control applied to a flexible joint robots manipulator. Thèse de Doctorat, université de Purdue, 1996.
- [Cha01] P.T. Chan, A.B. Rad, J. Wang : Indirect adaptive fuzzy sliding mode control, Part II: parameter projection and supervisory control. Fuzzy Sets and Systems, pp 21-30, 2001.
- [Cha10] H. Chao et al : A Comparative Evaluation of Low-Cost IMUs for Unmanned Autonomous Systems. IEEE IntConf on Multisensor Fusion and Integration for Intell Systems, USA, pp 211- 216, 2010.
- [Che95] Chi-Tsong Chen. Linear System Theory and Design. Oxford University Press, Inc. New York, NY, 1995.
- [Che06] W. Chen and M. Saif : Novel Sliding Mode Observers for a Class of Uncertain Systems. Proc American Control Conference, USA, 2006.
- [Che09] F. Chen and M.W. Dunnigan :Kalman filters for full state estimation in an induction machine. Electric Power Applications, Vol.149, No.1 pp 53-64, 2009.
- [Che11] A.L.Chen et al : Sensor data fusion for attitude stabilization in a low cost quadrotor system. IEEE 15thInt Symposium on Consumer Electronics, pp 34- 39, 2011.
- [Coz06] C. Coza and C.J.B. Macnab: A new robust adaptive-fuzzy control method applied to Quadrotor Helicopter stabilisation, in proc IEEE, FIPS Conference, pp. 454-458, Montreal, Canada 2006.
- [Das09] A. Das, F. Lewis et K. Subbarao. Backstepping Approach for Controlling a Quadrotor Using Lagrange Form Dynamics. J Intell Robot syst, pp 127-151, 2009.
- [Des95] C. W. de Silva. intelligent control: fuzzy Logic Applications, CRC Press, Boca Raton, FL, 1995.
- [Des08] D. L. Desages. Commande d’un robot mobile rapide à roues non directionnelles sur sol naturel. Thèse de doctorat, de l’université paris 6 – pierre et marie curie, 2008.

- [Die08] T. Dieks and S. Jagannathan. Neural Networks Out Feedback of a Quadrotor UAV. In Proc IEEE, 47th Conference on decision and control, pp 3633-3639, Cancun Mexico, Dec 2008.
- [Die09] T.Dieks and S.Jagannathan : Neural Network control of Quadrotor UAV Formations. Proc IEEE, American control conference, USA, 2009.
- [Din13] X. Ding and Y. Yu: Motion Planning and Stabilization Control of a Multi propeller Multifunction Aerial Robot. IEEE/ asme transactions on mechatronics, vol.18, no.2, pp 645-656, 2013.
- [Erg12] B. Erginer, E. Altuğ. Design and implementation of a hybrid fuzzy logic controller for a quadrotor VTOL vehicle. Int J of Control, Automation and Systems, Vol.10, No.1, pp 61-70, 2012.
- [Far06] J. A. Farrell, et M. M. Polycarpou. Adaptive approximation based control Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches. John Wiley & Sons, Inc., 2006.
- [Far09] M.Farza *et al.* Dynamic High Gain Observer Design. ProcIEEE, 6thInt Multi-Conf on systems, signals and Device, 2009.
- [Ioa84] P.A.Ioannou : Robust adaptive control, ACC, pp 1574-1578, 1984.
- [Ioa96] P. A. Ioannou and J. Sun, Robust adaptive control, Prentice-Hall, Englewood Cliffs, NJ, 1996.
- [Ioa06] P. A. Ioannou and B. Fidan : Adaptive control Tutorial. the Society for Industrial and Applied Mathematics, 2006.
- [Gau99] E. Gauthier. Utilisation des réseaux de Neurones Artificiels pour la Commande d'un Véhicule Autonome. Thèse de doctorat, l'institut national polytechnique de Grenoble, 1999.
- [Gev11] Gevers, M. and Vandendorpe, L. Processus stochastiques, estimation et prediction. DOI=<http://www.tele.ucl.ac.be/EDU/INMA2731/>,
- [Gui09] M.Guisser et H.Medromi : A High Gain Observer and Sliding Mode Controller for an Autonomous quadrotor. Int J. intelligent control and system, vol 14, No 3, pp 204- 212, 2009.
- [Ha99] Q.P. Ha *et al.* Fuzzy moving sliding mode control with application to robotic manipulators, Automatica Vol. 3, pp 607-616, 1999.

- [Haj10] M. Hajatipour, M. Farrokhi : Chattering free with noise reduction in sliding-mode observers using frequency domain analysis. *Journal of Process Control* Vol.20, pp 912–921, 2010.
- [Her09] M. Hernandez & Y. Tang. Adaptive output-feedback decentralized control of a class of second order nonlinear systems using recurrent fuzzy neural networks. *Neurocomputing*, pp 461-467, 2009.
- [Hua08] Ting Huang, et al. Neural Sliding-Mode Control of Engine Torque. *Proceedings of the 17th World Congress. The International Federation of Automatic Control*, Seoul, Korea, pp 9453- 9458, 2008.
- [Hun04] Lon-Chen Hung and Hung-Yuan Chung : Hybrid Neural Sliding Mode Controller Design for a Robot Manipulator, Submitted to *The Journal of Grey System* Date: Sept 2004.
- [Hus09] A. Hussain : Contribution à la commande adaptative robuste par modes Glissants. Thèse de doctorat, Université Reims Champagne Ardenne, 2009.
- [Ioa84] P.A.Ioannou : Robust adaptive control, *ACC*, pp 1574-1578, 1984.
- [Ioa06] P. Ioannou. B. Fidan. *Adaptive Control Tutorial*. Society for Industrial and Applied Mathematics Philadelphia. 2006.
- [Kal11] Kalyana C. Veluvolu *, Dongik Lee. Sliding mode high-gain observers for a class of uncertain nonlinear systems. *Applied Mathematics Letters*, Vol. 24, pp 329–334, 2011.
- [Kam13] M. Kamran et al : Position Controller Design for Quad-rotor under Perturbed Condition. *Wulfenia J*, Vol.20, No.7; pp 178-198, 2013.
- [Kar10] Kar, S, Moura, J.M.F; Ramanan, K.. Sample path large deviations for the randomly sampled continuous-discrete Kalman filter. Published in 49th IEEE Conference of Decision and Control (CDC), pp 7375 – 7382, 2010
- [Kha96] H. Khalil: *Nonlinear systems*, Printence-hall, London, Inc, 1996.
- [Kha08] H. Khalil: *High-Gain Observers in Nonlinear Feedback Control*. *ProcIEEE, IC on Control, Aut and Syst*, 2008
- [Kim04] D. H. Kim et al. *SICE Annual Conference in Sapporo, Hokkaido Institute of Tecnology, Japan* pp 629-634, 2004.
- [Kim11] Hongryel Kim et al : A High-Speed Sliding-Mode Observer for the Sensorless Speed

- Control of a PMSM. IEEE transactions on industrial electronics, vol.58, No.9, pp 4069-4077, 2011.
- [Kos99] E. B. Kosmatopoulos et P.A.Ioannu. A Switching Adaptive Controller for Feedback Linearizable Systems. IEEE trans on Automatic Control, Vol.44, No.4, Apr 1999.
- [Kos13] A. J. Koshkouei et K. J. Burnham. Adaptive backstepping sliding mode control for feedforward uncertain systems. International Journal of Systems Science, Vol. 42, No.12, pp 1935-1946, DOI: 10.1080/00207721.2011.617059.
- [Lan13] Sven Lange, et al : Incremental Smoothing vs. Filtering for Sensor Fusion on an Indoor UAV. 2013 IEEE Int Conference on Robotics and Automation (ICRA), Germany, May 6-10, 2013, pp 1773-1778.
- [Lee07] J. Lee et al. Robust Nonlinear Observer for Flexible Joint Robot Manipulators with Only Motor Position Measurement. International Conference on Control, Automation and Systems, pp 56-61, Seoul, Korea, 2007
- [Lee09] D. Lee, H. Jin Kim, et S. Sastry. Feedback Linearization vs. Adaptive Sliding Mode Control for a Quadrotor Helicopter. International Journal of Control, Automation, and Systems, Vol. 7, No.3, pp 419-428, 2009.
- [Len11] Z. Lendek *et al*: Application of Takagi-Sugeno observers for state estimation in a Quadrotor. 50th IEEE (CDC-ECC) pp 7530-7535, Orlando, FL, USA, Dec 12-15, 2011.
- [Leu09] Y. G. Leu et J. Y. Lin. Adaptive Backstepping Fuzzy Control for a Class of Nonlinear Systems. ISSN 5552, pp. 1123–1129, 2009.
- [Lia07] X. Liao, L. Wang et P. Yu. Stability of Dynamical Systems. Elsevier ISBN: 978-0-444-53110-0, ISSN: 1574-6917.
- [Lim12] Hyon Lim, Jaemann Park, Daewon Lee, and H.J. Kim: Build your own quadrotor. IEEE Robotics & Automation Magazine, pp 33-45, sept 2012.
- [Liu05] Y. Liu and Y. Li. Sliding Mode Adaptive Neural-Network Control for Nonholonomic Mobile Modular Manipulators. Journal of Intelligent and Robotic Systems, 2005.
- [Lin06] Chih-Min Lin, Chun-Fei Hsu and I-Fang Chung. Intelligent Backstepping Control for Chaotic Systems Using Self-Growing Fuzzy Neural Network. Springer-Verlag Berlin Heidelberg, pp.299–308, 2006.

- [Liu12] Y. Liu and D. Söffker. Variable high-gain disturbance observer design with online adaptation of observer gains embedded in numerical integration. *Mathematics and Computers in Simulation* Vol.82 pp 847–857, 2012.
- [Lot08] F. Lotte. Study of electroencephalographic signal processing and classification techniques towards the use of brain-computer interfaces in virtual reality applications. Thèse de doctorat de l'institut national des sciences appliquées de rennes mention informatique, 2008.
- [Lue71] DG. Luenberger. An introduction to observers . *IEEE Trans. On Auto. Const*; Vol. 16, No.6, 1971.
- [Mac06] C.J.B. Macnab : Robust Associative-Memory Adaptive Control in the Presence of Persistent Oscillations, *Neural information Processing Letters*, vol.10, no.12, pp 277-287, Dec 2006.
- [Mad07] T. Madani and A.Benallegue: Backstepping Control with Exact 2-Sliding Mode Estimation for a Quadrotor Unmanned Aerial Vehicle. *Proc of IEEE/RSJ International CIRS*, 2007.
- [Mar10] P. Martin and E. Salaun : The True Role of Accelerometer Feedback in Quadrotor Control. *IEEE Int Conference on Robotics and Automation Anchorage Convention District*, pp 1623- 1629, 2010.
- [Mat92] Osamu Matsumotoa, ShuujiKajitab&Kazuo Tani : Estimation and control of the attitude of a dynamic mobile robot using internal sensors. *Advanced Robotics*, Vol.7, No.2, pp 159-178, 1992.
- [Mir08] F. M. Mirzaei et S. I. Roumeliotis. A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation, *IEEE transactions on robotics*, vol. 24, no. 5, october 2008.
- [Mok05] A. Mokhtari, A. Benallegue, B. Daachi: Robust Feedback Linearization and GH^∞ Controller for a Quadrotor Unmanned Aerial Vehicle. *IEEE/RSJ IntConf on Intell Rob and Sys*. pp1009-1014, 2005.
- [Mok06] A. Mokhtari, A. Benallegue and B. Daachi: Robust Feedback Linearization and GH^∞ Controller for a Quadrotor Unmanned aerial vehicle. *Journal of Electrical Engineering*, vol.57, No.1, pp 20.27, 2006.
- [Mor06] Y. Morel, A. Leonessa. Direct Adaptive Tracking Control of Quadrotor Aerial Vehicles.

- 2006 Florida Conference on Recent Advances in Robotics, FCRAR, pp 1-6, 2006.
- [Nag12] K. Nagatani, A.Kushleyev& D. D. Lee: Sensor Information Processing in Robot Competitions and Real World Robotic Challenges. *Advanced Robotics*, Vol.26, No.14, pp 1539-1554, 2012.
- [Nag13] A. Nagaty *et al.* Control and Navigation Framework for quadrotors. *J Intelligent & Robotic Systems*, Vol.70, No.1, pp 1-12, 2013.
- [Nar00] F. Nardi. Neural Networks based adaptive algorithm for nonlinear control. Thèse de Doctorat, Institut de Technologie, Georgia, 2000.
- [Nec07] D. Neculescu, Y. W. Jiang et B. Kim. Neural Network Based Feedback Linearization Control of an Unmanned Aerial Vehicle. *International Journal of Automation and Computing*. Vol.4, No.1, pp 71-79, DOI: 10.1007/s11633-007-0071-y, 2007.
- [Oos12] A. Oosedo *et al* : Design and Attitude Control of a Quad-Rotor Tail-Sitter Vertical Takeoff and Landing Unmanned Aerial Vehicle. *Advanced Robotics*, Vol.26, pp 307–326, 2012.
- [Ond07] M. Onder Efe. Robust Low Altitude Behavior Control of a Quadrotor Rotorcraft Through Sliding Modes. Proc 15th med conf on control and aut, July 27-29, 2007, Greece.
- [Ond01] M.Onder Efe et Okyay Kaynak : Radial Basis Function Neural Networks in Variable Structure Control of a Class of Biochemical Processes. The 27th Annual Conference of the IEEE Industrial Electronics Society, 2001.
- [Oue12] M. Oueder et al. A high gain observer with updated gain for a class of MIMO non-triangular systems. *Systems & Control Letters*, Vol.61, pp 298–308, 2012.
- [Pir02] K. Pirabakaran and V.M. Becerra : PID Autotuning using Neural Networks and Model Reference Adaptive Control. 15th Triennial World Congress, Barcelona, Spain, 2002.
- [Poq97] L. Poquet et al. Identification and adaptive control of systems using recurrent Neural Networks. 3rd IFAC symposium on Intelligent Components and Instruments for control application, France, 1997.
- [Pra08] J. Prakash, R. Senthil. Design of observer based nonlinear model predictive controller for a continuous stirred tank reactor. *Journal of Process Control* Vol.18, pp 504–514, 2008.

- [Rao10] R. Raoufi : Nonlinear Robust Observers for Simultaneous State and Fault Estimation. Thèse de doctorat, université de Alberta, 2010.
- [Rub13] J. J. Rubio et al : An observer with controller to detect and reject Disturbances. International Journal of Control, DOI:10.1080/00207179.2013.845912, pp 1-13, 2013.
- [San92] R. M. Sanner et J. J. Slotine : Gaussian Networks for Direct Adaptive control. IEEE trans on Neural Networks, Vol.3, No.6, Nov 1992.
- [San11] R.Sanfelicce, L.Praly (2011). On the performance of high-gain observers with gain adaptation under measurement noise. Automatica, Vol.47, pp 2165–2176.
- [Sar06] J. Sarangapani. Neural Network Control of Nonlinear Discrete-Time Systems. Taylor & Francis Group, LLC, 2006.
- [Sef12] S. Sefriti *et al* . Adaptive Neural Network Sliding Mode Control For Electrically-Driven Robot Manipulators. Journal of Control Engineering and Applied Informatics. Vol.14, No.4, pp 27-32, 2012.
- [Ses00] S. Seshagiri and H.K. Khalil, “Output feedback control of nonlinear systems using RBF neural networks,”IEEE Trans. Neural Networks, vol. 11, no. 1, pp. 69 –79, 2000
- [Shi06] Shiraki R, Mori Y. The state presumption using the error presumptive observer loosen the restriction range by the sliding mode observer. SICE-ICASE International Joint Conference; Korea, 2006.
- [Shu05] H. Shu et Y. Pi. Decoupled Temperature Control System Based on PID Neural Network. ACSE 05 Conference, CICC, Cairo, Egypt, Dec 2005.
- [Slo91] J. J. Slotine : Applied nonlinear control. Prentice Hall, 1991.
- [Spo02] J.T.Spooner *et al* : Stable Adaptive Control and Estimation for Nonlinear Systems. John Wiley & Sons, 2002.
- [Sti09] E. Stingu, F. Lewis: Design and Implementation of a Structured Flight Controller for a 6DoF Quadrotor Using Quaternions. 17th Med Conf on Control & Auto, Greece, pp 1233- 1238, 2009.
- [Sto12] M. Stoica (2012). Kalman Filtering for Model-Based Networked Control Systems. Journal of Control Engineering and Applied Informatics, Vol.14, No.4, pp 10-16.
- [Sun13] G. Sun et al. Robust adaptive neural control of uncertain pure feedback nonlinear systems. Journal of Control, 86:5, 912-922, DOI: 10.1080/00207179.2013.765039

- [Rai08] J. C. Raimundez and A. F. Villaverde. Adaptive Tracking control for a Quad-rotor. ENOC-2008, saint Petersburg, Russia July 2008.
- [Ram06] H. S. Ramirez, R. M. Ortigoza. Control Design techniques power Electronics Devis. Springer-Verlag London 2006.
- [Rum86] D. E. Rumelhart et J. L. McClelland. Parallel Distributed Processing, MIT Press, Cambridge, 1986.
- [Tan13] M. H. Tanveer *et al*: Stabilized Controller Design for Attitude and altitude controlling of quad-rotor under disturbance and noisy conditions. American J of Applied Sci, Vol.10, No.8, pp 819-831, 2013.
- [Tao03] Gang Tao. Adaptive control design and analysis. John Wiley & sons, 2003.
- [Tap01] A. V. Tapalov et O. Kaynak. Online Learning in Adaptive Neurocontrol Schemes with a Sliding Mode Algorithm. IEEE trans on systems, man and Cybernetics, Vol.31, No.3, June 2001.
- [Tee92] A. R. Teel, Global stabilization and restricted tracking for multiple integrators with bounded controls. Syst. Control Lett., 1992.
- [Tin07] Hu Tingliang, Zhu Jihong and Sun Zengqi : Robust adaptive Neural Control of a class of MIMO Nonlinear systems. Tsinghua Science and Technology, Vol.12, No.1, pp14-21, Feb 2007.
- [Tli06] B. Tlili, F. Bouani et M. Ksouri. Estimation des paramètres du modèle NARMA a l'aide des réseaux de neurones et des algorithmes génétiques. 6e conférence francophone de modélisation et simulation -MOSIM'06, rabat, Maroc. 2006.
- [Ton12] Ren, S. Tong and Y. Li (2012). Fuzzy adaptive high-gain-based observer backstepping control for SISO nonlinear systems with dynamical uncertainties. Nonlinear Dyn, Vol.67, pp 941–955.
- [Top07] A. V. Topalov et al. Sliding Mode Neuro-Adaptive Control of Electric Drives. IEEE transactions on industrial electronics, vol. 54, no. 1, pp 671- 679, 2007.
- [Vel11] Kalyana C. Veluvolu, Dongik Lee : Sliding mode high-gain observers for a class of uncertain nonlinear systems. Applied Mathematics Letters, Vol.24, No.3, pp 329–334, 2011.
- [Wan01] J. Wang, A.B. Rad, and P.T. Chan : Indirect adaptive fuzzy sliding mode control, Part I: fuzzy switching. Fuzzy Sets and Systems, pp 21-30, 2001.

- [Xu01a] H. Xu, P.A.Ioannu and H.R.Boussalis : Robust adaptive sliding control of linearizable systems. Proc of American control conference, pp25-27, June 2001.
- [Xu01b] H. Xu, P.A.Ioannu. robust adaptive control for a class of MIMO nonlinear systems. USC report No 1-12-01, nov.12, 2001
- [Xue12] Xuewu Dai; Zhiwei Gao ; Breikin, T. ; Hong Wang. High-Gain Observer-Based Estimation of Parameter Variations With Delay Alignment. IEEE Transactions on Automatic control (Volume:57 , Issue: 3) pp726 – 732, 2012.
- [Yas06] M.Yasser and al. An Adaptive Sliding Mode Control Using Simple Adaptive Control for A Class of SISO Nonlinear Systems with Bounded-Input Bounded-Output and Bounded Nonlinearity. Proc of the 45th IEEE Conference on Decision & Control Manchester San Diego, USA, Dec 2006.
- [Zem06] K. Zemalache. Commande d'un système sous actionnée : Application à un drone à quatre hélices. Thèse de Doctorat. Université d'Evry Val d'Essonne, 2006.
- [Zem09] K. Zemalache et H. Maaref. Controlling a drone : comparison between a based model method and a fuzzy inference system. Applied soft Computing, pp 553-562, 2009.
- [Zha00] T. Zhang, S.S.Ge and C.C.Hang. Adaptive neural network control for strict-feedback nonlinear systems using backstepping design. Automatica Vol 36, pp 1835-1846, 2000.
- [Zha05] Y. Zhang et X. Wang. Neural network indirect adaptive sliding mode tracking control for a class of nonlinear interconnected systems. springer-verlag berlin heidelberg , lncs 3611, pp. 283 – 291, 2005.
- [Zha10] Zhao J, Shen ZY, Gu XS, Xia CM. LMI design approach to robust fuzzy sliding-mode observer in dynamical uncertain systems. Proceedings of the 8th World Congress on Intelligent Control and Automation; 2010; China.
- [Zha11] T. Zhang et al. Multi-Sensory Motion Estimation and Control of an Autonomous Quadrotor. Advanced Robotics, Vol.25, pp 1493–1514, 2011.
- [Zil01] A. Zilouchian et M. Jamshidi. Intelligent Control Systems Using Soft Computing Methodologies. CRC Press LLC, 2001.
- [Zon12] Q. Zong *et al.* Output feedback back-stepping control for a generic Hypersonic Vehicle via small-gain theorem. Aerospace Science and Technology, Vol. 23, pp 409–417, 2012.