République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Batna 2 – Mostefa Ben Boulaïd

Faculté de Technologie

Département d'Electronique

Thèse

Préparée au sein du Laboratoire d'Automatique
Avancée et Analyse des Systèmes (LAAAS)

Présentée pour l'obtention du titre de :

Docteur en Électronique

Option : Robotique et intelligence artificielle

Sous le Thème :

Application des techniques de l'intelligence artificielle à la commande des systèmes robotiques

Présentée par :

MEDJGHOU Ali

Devant le jury composé de :

M. BOUGUECHAL Nour-Eddine	Prof.	Université Batna -2-	Président
M. CHAFAA Kheireddine	Prof.	Université Batna -2-	Rapporteur
M. SLIMANE Noureddine	MCA	Université Batna -2-	Co-Rapporteur
M. GHODBANE Hatem	Prof.	Université de Biskra	Examinateur
M. TOUBA Mostefa	MCA	Université de Biskra	Examinateur

08 Juillet 2018

Résumé de la thèse

Résumé

Le travail présenté dans cette thèse considère l'utilisation des techniques de l’intelligence artificielle pour la commande des systèmes robotiques. Ce travail est une étude par simulation de quelques techniques de commandes non-linéaires; il constitue une contribution à la commande et l’observation robustes d’une classe de systèmes dynamiques non-linéaires. Nous avons développé deux structures de commande robuste, l’une d’elles se base sur la commande backstepping-mode-glissant, combinée avec un système flou type-1 qui est appliqué au modèle considéré. L’autre se base sur la commande linéarisante renforcée par une commande discontinue combinée à un système flou. La mise en œuvre d’une loi de commande nécessite souvent l'accès à la valeur d'un ou de plusieurs états du système à commander, cependant, ces états sont rarement disponibles à la mesure directe. Dans ce contexte, le filtre de Kalman étendu (EKF) est utilisé comme observateur pour estimer les états non mesurables. La performance de cette estimation est le problème majeur associé à ce filtre, puisqu’il est fortement influencé par les paramètres du système et par les matrices de covariance des bruits d'état et de mesure Q et R, respectivement. Ces matrices sont ajustées de manière empirique par la méthode essai-erreur, par essaim de particules (PSO) et enfin par optimisation basée sur la biogéographie (BBO). Les résultats de simulation confirment l'efficacité des lois de commande proposées, et montrent les avantages des contrôleurs avec des observateurs optimisés.

Mots clés

Intelligence artificielle, Système à logique floue, Commande linéarisante, Technique backstepping, Commande par mode glissant, Stabilité de Lyapunov, Observateur EKF, Algorithme PSO, Algorithme BBO.

Abstract

The work presented in this thesis is dedicated to the use of artificial intelligence techniques to the control of robotic systems. This work is a simulation study of some nonlinear control techniques and constitutes a contribution to the robust control and observation for a class of nonlinear dynamic systems. We have developed two robust control schemes, the first is based on the backstepping-sliding-mode control combined with a type-1 fuzzy system, which is applied to the considered model, and the second is based on the feedback linearization control reinforced by a discontinuous control, and combined with a fuzzy system. The implementation of a control law often requires access to the value of one or more states of the controlled system. However, these values are rarely available to direct measurement. In this context, the extended Kalman filter (EKF) is used as an observer to estimate non-measurable states. The performance of this estimate is the major problem associated with this filter, since it is strongly influenced by the system parameters and the covariance matrices of state and measurement noises Q and R, respectively. These matrices are empirically adjusted by trial-and-error methods, particle swarm optimization (PSO) and biogeography-based optimization (BBO). The simulation results confirm the effectiveness of the proposed control laws, and show the advantages of controllers combined with optimized observers.

Keywords

Artificial intelligence, Fuzzy logic system, Feedback linearization, Backstepping technique, Sliding mode control, Lyapunov stability, EKF observer, PSO algorithm, BBO algorithm.

ملخص

يهدف العمل المقدم في هذه الأطروحة إلى استخدام تقنيات الذكاء الاصطناعي في التحكم على الأنظمة الروبوتية، هذا العمل هو دراسة محاكاة لبعض تقنيات التحكم غير الخطية ويشكل مساهمة في التحكم والملاحظة المتينين لفئة من الأنظمة الديناميكية غير الخطية، قمنا بتطوير قانونين للتحكم المتين. أحدهما يرتكز على التحكم بنمط انزلاقي، بواسطة توليف الخطوات إلى الوراء، مدمج مع نظام منطقي غامض نوع 1، والذي تم تطبيقه على النموذج الذي تم دراسته، والآخر يستند على التحكم الخطي بواسطة التغذية المرتدة، المدعم بالتحكم غير المستمر والمدمج مع نظام غامض، غالبا ما يتطلب تنفيذ قانون التحكم الوصول إلى قيمة واحدة أو أكثر من متغيرات الحالة للنظام الخاضع للتحكم، ومع ذلك نادرا ما تكون تلك المتغيرات متاحة للقياس المباشر، في هذا السياق تم استخدام مرشح كالمان الموسع (EKF) كملاحظ لتقدير متغيرات الحالة غير المتاحة للقياس، إلا أن المشكلة الرئيسية المرتبطة بهذا المرشح هي أداء التقدير، لأنه يتأثر بشدة بـ : معاملات النموذج ومصفوفتي التغاير Q و R لتشويشيهما (الحالة والقياس) على التوالي. تم تعديل هاتين المصفوفتين تجريبيا باستعمال طريقة التجربة والخطأ ، خوارزم التحسين بسرب الجسيمات (PSO)، خوارزم التحسين القائم على الجغرافيا الحيوية (BBO).

الكلمات الدالـــــة

الذكاء الاصطناعي، نظام منطقي غامض، التحكم الخطي بواسطة التغذية المرتدة، التحكم بنمط انزلاقي، تقنية توليف الخطوات إلى الوراء، استقرار ليابونوف، مرشح كالمان الموسع، خوارزم التحسين بسرب الجسميات، خوارزم التحسين القائم على الجغرافيا الحيوية.

Dédicace

À la mémoire de mon grand-père maternel Ahmed Rahmouni.

À la mémoire de mon frère Abdelhamid.

À mes très chers parents

À ma femme, mes chères petites filles Loudjeine et Djihene, ainsi que mes frères et sœurs

À toute mes amis sans exception.

Et à toi...

Remerciements

Le travail présenté dans cette thèse a été effectué au sein du Laboratoire d'Automatique Avancée et Analyse des Systèmes (*LAAAS*), *Département d'Électronique, Faculté de Technologie de l'Université Mostefa Ben Boulaid, Batna 2.*

Avant toute chose, je tiens à remercier Dieu pour l'accomplissement de ce projet. Je tiens à exprimer mes profonds remerciements à mon directeur de thèse, Mr. Kheireddine CHAFAA, professeur au Département d'Électronique à l'Université Batna 2, qui à assuré la direction de mon travail. Je le remercie pour son soutien intellectuel, ses conseils et pour la confiance qu'il m'a accordée tout au long de cette thèse.

Je tiens également à exprimer ma profonde gratitude à mon Co-directeur de thèse, Mr. Noureddine SLIMANE, Maître de Conférences et responsable de la filière Robotique et Intelligence Artificielle à l'Université Batna 2, pour le suivi de cette thèse, ses conseils et son soutient tout au long de ce travail.

J'adresse également mes sincères remerciements à Monsieur Nour-Éddine BOUGUECHAL, Professeur à l'Université Batna 2, pour accepté de présider mon jury de thèse, ainsi qu'à Monsieur Hatem GHODBANE, professeur à l'Université de Biskra, et Monsieur Mostefa TOUBA, Maître de Conférences à l'Université de Biskra, pour l'honneur qu'ils me font en acceptant d'examiner ce travail.

Enfin, je n'oublie pas d'adresser mes vifs remerciements à toute ma famille, qui m'a accompagné tout au long de mes études par son amour inconditionnel et son soutien constant. Merci à ma petite famille, ma femme pour son indéfectible soutien et mes enfants sources de joie inaltérable, ainsi que toutes les personnes ayant contribué, de près ou de loin, au bon déroulement de ce travail.

Table des matières

Chapitre III
Modélisation et commande des systèmes non-linéaires

Chapitre IV
Commande par backstepping-mode-glissant floue type-1 : Application au quadrirotor

Chapitre V
Commande linéarisante robuste par mode glissant floue type-1 avec observateur optimisé : Application au bras manipulateur

Chapitre VI
Commande floue type-2 intervalle par mode glissant avec observateur optimisé : Application au bras manipulateur

Index des figures

Liste des tableaux

Liste des abréviations

SISO	:	*Single Input Single Output*, Mono entrée Mono sortie
EKF	:	*Extended Kalman Filter*, Filtre de Kalman étendu
SMO	:	*Sliding Mode Observer*, Observateur à mode glissant
LMI	:	*Linear Matrix Inequality*, Inégalité matricielle linéaire
MRAS	:	*Model Reference Adaptive Systems*, Système adaptatif à modèle de référence
LTV	:	*Linear Time Varying*, Linéaire à temps variant
MSE	:	*Mean Square Error*, Erreur quadratique moyenne
FLS	:	*Fuzzy Logic System*, Système à logique floue
PSO	:	*Particle Swarm Optimization*, Optimisation par essaims particulaires
GA	:	*Genetic Algorithms*, Algorithmes génétiques
BBO	:	*Biogeography-Based Optimization*, Optimisation basée sur la biogéographie
UAV	:	*Unmanned Aerial Vehicles*, Véhicule aérien sans pilote
PD	:	*Proportional - Derivative*, Proportionnel - Dérivé
PI	:	*Proportional - Integral*, Proportionnel - Intégral
PID	:	*Proportional - Integral - Derivative*, Proportionnel - Intégral - Dérivé
LQ	:	*Quadratic Linear*, Linéaire quadratique
LQR	:	*Linear Quadratic Regulator*, Régulateur linéaire quadratique
DC		*Direct Current*, Courant continu
SMC	:	*Sliding Mode Control*, Commande par mode glissant
BSMC	:	*Backstepping Sliding Mode Control*, Commande backstepping-mode-glissant
FBSMC	:	*Fuzzy Backstepping-Sliding-Mode Control*, commande backstepping-mode-glissant floue.
T2FLS	:	*Type-2 Fuzzy Logic System*, Système à logique floue type-2
IT2FLS	:	*Interval Type-2 Fuzzy Logic System*, Système à logique floue type-2 intervalle

Introduction générale

La science dissipe bien des doutes, elle nous montre ce qui est caché.

Proverbe sanskrit ; Hitopadésa - IXe siècle.

La robotique est un domaine de la technologie moderne, qui traverse relativement les limites d'ingénierie traditionnelle. Depuis la fin du dernier siècle, l'évolution des technologies a touché tous les secteurs de l'industrie en général et la robotique en particulier. De nouvelles disciplines de l'ingénierie, tels que le génie de fabrication, l'ingénierie des applications et l'ingénierie des connaissances ont émergé pour faire face à la complexité du domaine de la robotique et l'automation d'usine. En effet, le développement croissant de l'automatisation a permis l'amélioration des performances des différents dispositifs, mais aussi a entraîné une prise en compte de leur stabilité. De plus, les travaux de recherche dans l'industrie et en robotique ont généralement pour objectif la réalisation d’une tâche de manière plus précise possible à l’aide d’une stratégie de commande. Cependant, plus les contraintes sur la réalisation de la tâche est forte et plus la dynamique non-linéaire des systèmes est complexe, ce qui augmente la complexité de la commande. Les systèmes robotiques constituent une classe particulière de systèmes non-linéaires et sont caractérisés par des comportements dynamiques très sensibles aux variations des paramètres de conception et aux perturbations externes.

Les progrès enregistrés ces deux dernières décennies dans la théorie de la commande des systèmes non-linéaires ont donné naissance à certaines méthodes systématiques de synthèse de lois de commande non-linéaires. L’une des méthodes de commande non-linéaires les plus connues est la commande linéarisante (*feedback linearization*, en anglais). Elle consiste à linéariser le système par compensation et appliquer à ce nouveau système une commande linéaire classique telle que la commande par retour d’état [1][2]. Néanmoins, elle est sensible aux variations paramétriques et aux erreurs de modélisation. Avec le développement considérable des calculateurs numériques, les automaticiens ont adopté de plus en plus de nouvelles approches telles que les commandes non-linéaires robustes qui donnent des résultats acceptables dans de larges domaines de fonctionnement.

Parmi ces approches, on trouve la commande par mode glissant [1][3][4], cette dernière, apparaît comme une bonne solution permettant d'assurer la robustesse, la précision, la stabilité et la simplicité pour les systèmes perturbés. Elle fait l'objet de plusieurs travaux de recherche; seule ou en hybridation avec d'autres techniques de commande. Néanmoins, cette solution présente l'inconvénient en raison du phénomène de broutement ("*chattering*", en anglais) [3][4]. Pour réduire ce phénomène, différentes techniques peuvent être trouvées dans la littérature, à savoir : la solution de l'introduction d'une bande de transition autour de la surface de glissement pour transformer la fonction signe en saturation, ou la substitution de la fonction signe par d'autres à variation plus douce, l'usage de la couche limite [5][3], solution basée sur un observateur [6], les modes glissants d'ordre supérieur [7][8], la commande par mode glissant terminal [9]. Cependant, ces techniques, bien qu'elles peuvent réduire l'effet du broutement, engendrent un compromis entre le niveau de performances de poursuite et les sollicitations de commande au démarrage [10], d'un autre côté, une erreur statique subsiste. L'utilisation de techniques basées sur l'expertise humaine peut être une alternative de résolution de ce problème.

Dans ce contexte, les techniques de l'intelligence artificielle [11a][11b][12][13], notamment la logique floue, est utilisée pour résoudre ce problème pour son efficacité et sa robustesse vis-a-vis de l'incertitude des modèles mathématiques des systèmes à commander, et sa capacité à résoudre certains problèmes liés aux erreurs de modélisation.

La logique floue classique appelée aujourd'hui logique floue type-1 a été généralisée vers une nouvelle logique floue appelée logique floue type-2. Ces dernières années, Mendel et ses collègues ont beaucoup travaillé sur cette nouvelle logique [14][15], ils ont bâti son fondement théorique, et ils ont démontré son efficacité et sa supériorité par rapport à la logique floue type-1 [16]. Dans le contexte du phénomène de broutement, Ho *et al.*[17] ont propose d'augmenté la commande mode glissant par un terme de PI flou type-1, Noroozi *et al.*[18] ont proposé d'approximer la partie discontinue de la commande à mode glissant par un système flou type-1 adaptatif, et par un système flou type-2 intervalle dans [19].

Le comportement dynamique d'un système peut être entièrement décrit par l'évolution de ses variables d'état. La connaissance de ces variables est nécessaire dans de nombreuses stratégies, notamment de détection de défauts, de diagnostic et de commande.

L'élaboration d'une loi de commande d'un système, nécessite souvent l'accès à la valeur d'un ou de plusieurs de ses états. Ceci n'est pas toujours possible à cause de l'inaccessibilité d'état et/ou le manque de capteurs. En plus, les mesures des capteurs sont souvent entachées de bruit, ce qui limite les performances d'une boucle de commande. Pour des raisons physiques ou économiques, on a recours à un système dynamique auxiliaire, appelé observateur, qui est chargé d'estimer l'état du système à partir des entrées appliquées et des mesures fournies par des capteurs physiques. Plusieurs algorithmes sur ce sujet peuvent être trouvés dans la littérature, à savoir : l'observateur de Luenberger étendu, le filtre de Kalman étendu, l'observateur par mode glissant, l’observateur basé sur la technique MRAS, l’observateur basé sur les réseaux de neurones et l’observateur basé sur la logique floue. Parmi ces observateurs, le filtre de Kalman étendu fournit l'estimateur d'état optimal en raison de sa capacité à considérer les incertitudes stochastiques. La performance d'estimation est le problème majeur associé à EKF; il a forte influence sur les paramètres du système et les matrices de covariance du bruit d'état et du bruit de mesure Q et R, respectivement. L’amélioration de sa performance d'estimation peut être assimilée à un problème d'optimisation.

L’objectif de cette thèse est la mise en évidence des avantages des techniques de l’intelligence artificielle, pour la synthèse des lois de commande non-linéaires robustes, pour commander certain types de systèmes robotiques. Ces lois assurent la stabilité et la robustesse vis-à-vis des variations paramétriques et/ou perturbations qui peuvent être dues, soit, aux perturbations environnementales de fonctionnement soit aux problèmes d’exploitation internes.

Dans un souci de compréhension, deux systèmes robotiques sont proposés tout au long de ce travail, permettant d’évaluer les performances des approches proposées. Le premier, est un quadrirotor correspondant à un système non-linéaire multi-variable complexe du sixième ordre fortement couplé. Le second, est un bras manipulateur à deux degrés de liberté correspondant à un système non-linéaire multi-variable du deuxième ordre.

Cette thèse est composée de six chapitres. Les trois premiers chapitres sont théoriques, les trois derniers correspondent à des applications pratiques. Elle est organisée de la façon suivante:

Chapitre I : Techniques de l'intelligence artificielle : Les logiques floues type-1 et type-2

Dans ce chapitre, nous présentons un aperçu sur deux approximateurs utilisés en automatique, à savoir : Les systèmes flous type-1 et type-2 pour aborder l’utilisation du système flou en commande. Nous abordons quelques concepts de base ainsi que leurs structures générales.

Chapitre II : Méthodes d’optimisations métaheuristiques

Dans ce chapitre, on expose les fondements théoriques de trois méthodes des algorithmes d’optimisation métaheuristiques. La première, est basée sur les algorithmes génétiques. La deuxième, est l'optimisation par essaim de particules. La troisième, est l'optimisation basée sur la biogéographie.

Chapitre III : Modélisation et commande des systèmes robotiques

Ce chapitre a essentiellement pour objectif de présenter quelques rappels indispensables et nécessaires à la compréhension du contenue de cette thèse. On entame par une introduction aux systèmes robotiques et nous présentons quelques notions de stabilité des systèmes non-linéaires. Ensuite, nous exposons un rappel sur la modélisation, représentation entrée-sortie des systèmes non-linéaires. Puis, nous présentons les fondements et les aspects théoriques de quelques approches de commande non-linéaires, telles que : la commande linéarisante, la commande par mode glissant et la commande backstepping. Enfin, nous présentons quelques approches qui ont été très étudiées dans la littérature au cours des dernières années, concernant l'estimation d'état des systèmes non-linéaires, telles que l'observateur à mode glissant et le filtre de Kalman étendu.

Chapitre IV : Commande par backstepping-mode glissant floue type-1 : Application au quadrirotor

Dans ce chapitre, après une brève présentation du contexte historique et opérationnel dans lequel les mini drones sont utilisés, nous exposons l'étude bibliographique qui nous a permis de présenter un état de l'art sur les différentes stratégies de commande mises en œuvre pour remplir les différentes missions du quadrirotor. Ensuite, nous donnons une formulation du problème. Puis, nous synthétisons une loi de commande robuste pour la commande de l'altitude et de l'attitude d'un quadrirotor en présence de perturbations externes élevées.

Chapitre V : Commande linéarisante robuste par mode glissant floue type-1 avec observateur optimisé : Application au bras manipulateur

L'objectif de ce chapitre est construction d'un schéma de commande et d'observation assurant à la fois, la robustesse et la performance adaptées au système de robot manipulateur. Premièrement, nous proposons une loi de commande pour une classe de systèmes non-linéaires. La loi de commande proposée consiste en une loi de commande linéarisante classique augmentée d'une composante de robustification par mode glissant, où le gain de commande est un système flou type-1. La combinaison de ces techniques avec celles de la commande linéarisante permet de concevoir une commande linéarisante robuste vis-à-vis des variations paramétriques du système et des perturbations externes. Deuxièmement, afin d'améliorer la performance de l'observateur EKF, pour garantir la stabilité et la poursuite en boucle fermée, les matrices de covariance Q et R doivent être optimisées dans lesquelles l'optimisation est assurée par une méthode métaheuristique inspirée par les interactions sociales; qui est l'algorithme d'optimisation PSO.

Chapitre VI : Commande floue type-2 intervalle par mode glissant avec observateur optimisé : Application au bras manipulateur

Ce chapitre présente, une commande par mode glissant floue type-2 (IT2FSMC), combiné avec un observateur EKF optimisé, en présence d'incertitudes et de perturbations pour les manipulateurs robotiques. La contribution principale est la proposition d'une nouvelle méthode basée sur une combinaison du filtre de Kalman étendu avec optimisation basée sur la biogéographie afin d'obtenir une haute performance d'estimation d'états du système à commander, et il est ensuite comparé à l'algorithme PSO.

Enfin, ce travail sera clôturé par une conclusion générale à travers laquelle on expose les principaux résultats obtenus et on donne les perspectives à envisager comme suite à ce travail.

Chapitre I

Techniques de l'intelligence artificielle : Les logiques floues type-1 et type-2

Sommaire

Techniques de l'intelligence artificielle : Les logiques floues type-1 et type-2

L'intelligence artificielle se définit comme le contraire de la bêtise naturelle.

Woody Allen.

I.1. Introduction

La commande basée sur l'intelligence artificielle représente un domaine de recherche très active et vaste à cause des avantages qu'offrent ces outils surtout pour les systèmes non-linéaires. Elle est fondée sur l'exploitation des capacités d'apprentissage et d'optimisation qui caractérisent ces outils. Parmi ces outils on cite les réseaux de neurones artificiels, la logique floue et les algorithmes évolutionnaires [11a][11b][12][13][20][21]. Bien que la logique floue (*fuzzy logic* en anglais) est utilisée dans de vastes domaines d'applications (commande, classification, aide à la décision, etc.), on ne s'intéresse qu'à son utilisation dans le cadre de la commande. Ce chapitre est destiné à donner les principes de logiques floues type-1 et type-2, nécessaires pour l'élaboration d'un contrôleur basé sur un système flou.

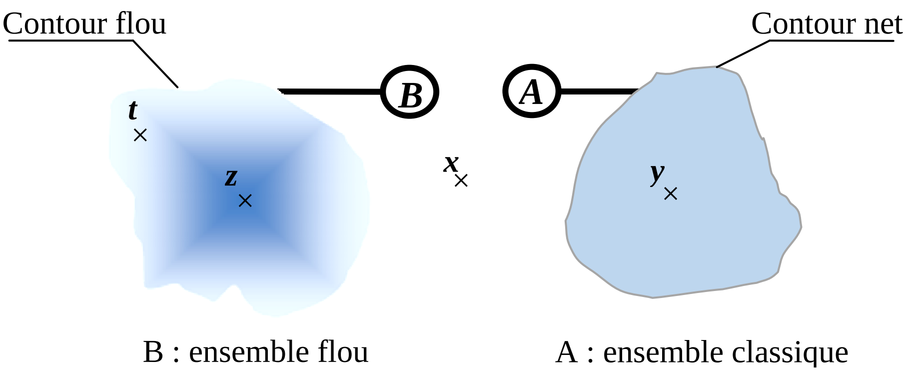
I.2. Généralités sur la logique floue

La logique floue type-1 à été introduite par le professeur Lotfi Zadeh [22][23] comme une généralisation de la logique binaire [24]. Il a introduit la notion de sous-ensemble flou pour fournir un moyen de représentation et de manipulation des connaissances imparfaitement décrites, vagues ou imprécises. L'intérêt de la logique floue réside dans sa capacité à traiter l'imprécis, l'incertain et le vague. Elle à été utilisée dans le domaine de la commande pour une large gamme de systèmes [25][26][27][28][29] et plus généralement en génie électrique [30][31]. Le principe de la commande par la logique floue s'approche de la démarche humaine, elle permet d'exploiter l'expérience humaine dans le domaine de la commande avec la capacité de traiter les informations incertaines. Ainsi, elle présente l'avantage d'utiliser des règles linguistiques simples permettant de traduire facilement le savoir faire d'un expert pour répondre à une problématique spécifique. Sur la base de ce principe, différentes réalisations ont vu le jour [32][33][34][35].

Le concept des ensembles flous type-2 a été également introduit par Lofti Zadeh en 1975 [36]. Néanmoins, le premier système flou type-2 a été développé et présenté seulement 23 ans plus tard par Karnik et Mendel [37]. Mendel et ses collègues ont beaucoup travaillé sur cette nouvelle logique [14][15]. Ils ont bâti son fondement théorique et ont démontré son efficacité et sa supériorité par rapport à la logique floue type-1 [16].

I.2.1. Ensembles classiques et ensembles flous

La théorie mathématique des sous ensembles flous est une extension de la théorie des ensembles classiques pour la prise en compte des sous ensembles définis de façon imprécise. A l'inverse de la logique booléenne, la logique floue permet à une condition d'être un autre état que vrai ou faux. Il y a des degrés dans la vérification d'une condition. La logique floue tient compte de l'imprécision de la forme des connaissances et propose un formalisme rigoureux afin d'inférer de nouvelles connaissances. Ainsi, la notion d'un sous ensemble flou permet de considérer des classes d'objets dont les frontières ne sont pas clairement définies, par l'introduction d'une fonction caractéristique prenant des valeurs entre 0 et 1. La théorie des ensembles flous repose sur la notion d'appartenance partielle de chaque élément appartenant partiellement ou graduellement aux ensembles flous qui ont été définis. Les contours de chaque ensemble flou ne sont pas « nets », mais «flous » (voir la figure I.1).



x n'appartient ni à A ni à B
y appartient totalement à A
z appartient totalement à B
t appartient partiellement à B

Fig. I.1. Représentation d'un ensemble classique (A) et d'un ensemble flou (B).

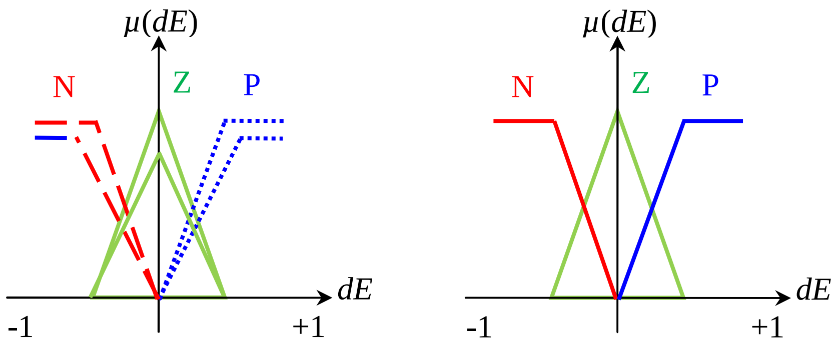
La notion d'ensemble flou provient du constat que très souvent, les classes d'objets rencontrés dans le monde physique ne possèdent pas de critères d'appartenance bien définis [38]. Mathématiquement un ensemble flou A est défini sur un univers de discours X par une fonction d'appartenance $\mu_A(x)$ qui prend ses valeurs dans l'intervalle $[0, 1]$. Cette fonction donne le degré d'appartenance de chaque élément $x \in X$ à A, à partir d'une description mathématique.

I.2.2. Variables linguistiques

Les variables linguistiques sont des variables dont les valeurs sont des mots ou des groupes de mots [24][39]. Une variable linguistique est représentée par un triplet (v , S_v , Υ_v) tel que :

- v désigne le nom de la variable.
- S_v est l'univers des valeurs prises par v .
- $\Upsilon_v = \{A_1, A_2, ...\}$ est un ensemble de sous-ensembles flous de S_v utilisés pour caractériser v .

A titre d'exemple, on peut envisager la variable linguistique représentée par le triplet (dE, $[-1, +1]$, {N, Z, P}) où N, Z et P signifient respectivement négative, zéro et positive. Cet exemple de partition floue est illustré sur la figure I.2.



(a) (b)

Fig. I.2. Exemples de partitions floues (a) type-1 (b) type-2

A ce stade, on peut remarquer que les variables linguistiques permettent de caractériser des situations avec moins d'informations que les représentations numériques, pour lesquelles un nombre important de valeurs doit parfois être spécifié. Leur usage est donc avantageux pour le traitement des informations par un système de raisonnement approximatif.

I.3. Les types de systèmes flous

Le système d’inférence flou est un système de prise de décision à partir d’une base de règles. On distingue classiquement deux types de systèmes flous à base de règles: les systèmes flous à conclusion symbolique (systèmes flous linguistiques ou de Mamdani) et les systèmes flous à conclusion fonctionnelle (systèmes flous de Takagi-Sugeno). Pour ces deux systèmes flous, les règles sont de la forme ***SI … ALORS***, mais les types de conclusions sont différents.

I.3.1. Méthode de Mamdani

Dans la méthode de Mamdani, on utilise des variables linguistiques pour représenter à la fois prémisses et conclusions. Cette méthode repose sur l’utilisation de l’opérateur min (minimum) pour l’inférence floue et de l’opérateur max (maximum) pour l’agrégation des règles. En pratique, pour la défuzzification, la méthode du centre de gravité est très utilisée. Plus de détails sont disponibles dans [25][40].

I.3.2. Méthode de Takagi-Sugeno

Dans la méthode de Takagi-Sugeno [41], les conclusions sont de type numérique sous forme de constantes (singleton), de polynômes ou de fonctions non-linéaires des entrées. L’inférence floue est réalisée avec les opérateurs min ou produit. La valeur finale est obtenue en effectuant une moyenne pondérée des conclusions.

I.4. Système d’inférence flou type-1

I.4.1. Notions et définitions

Les systèmes de Sugeno exploitent des conclusions numériques alors que les systèmes de Mamdani utilisent des conclusions symboliques de même nature que les prémisses. Cette différence dans l’écriture des conclusions de règles permet de distinguer les deux types de systèmes à leur aspect externe. D’un point de vue interne, un mécanisme de calcul est associé à chaque famille de systèmes. Dans notre étude, on limite la présentation des systèmes flous à ceux qui sont directement exploités dans les approches de commande, à savoir les systèmes flous de Mamdani à conclusions symboliques.

Dans cette section, on se limite à la présentation de quelques notions et définitions de base de la logique floue nécessaires à la construction d'un système flou type-1.

Définition I.1 : Fonction d'appartenance : Il n'y a pas de règle précise pour la définition de la fonction d'appartenance, alors chaque ensemble flou peut être représenté par sa fonction d'appartenance. En général, la forme de la fonction d'appartenance dépend de l'application.

Les fonctions d'appartenance peuvent avoir différentes formes [42][43] : triangulaire, trapézoïdale, gaussienne et sigmoïdale:

- ***Fonction triangulaire*** **:** Elle est définie par trois paramètres $\{a,b,c\}$:

$$\mu_A(x)=\max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) \tag{I.1}$$

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

$$\mu_A(x)=\max\left(\min\left(\frac{x-a}{b-a},1,\frac{d-x}{d-b}\right),0\right) \tag{I.2}$$

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

$$\mu_A(x)=\exp\left(-\frac{(x-a)^2}{2\sigma^2}\right) \tag{I.3}$$

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

$$\mu_A(x)=1\Big/\big(1+\exp\left(-a(x-b)\right)\big) \tag{I.4}$$

La représentation formelle des ensembles flous par des fonctions d'appartenance a permis de généraliser les opérateurs des ensembles classiques au cas flou. Nous pouvons citer, par exemple, égalité floue, complémentation floue, union floue, intersection floue, etc. Pour plus de détails le lecteur est invité à consulter [44][45].

Définition I.2: Règle et implication floues: Les variables linguistiques définies par des fonctions d'appartenance sont liées entre elles par des règles permettant de tirer des conclusions, on parle alors de déductions floues ou inférences. Ces règles floues sont élaborées à partir de la connaissance du système, issue très souvent de l'observation expérimentale, et permettent de décrire son évolution. Les règles floues, qui sont des objets linguistiques, doivent être mises sous forme de relations floues qui sont des objets mathématiques.

Soit A et B deux ensembles flous définis dans l'univers de discours X et Y, une règle floue est une relation entre deux propositions floues ayant chacun un rôle particulier, par exemple

$$\textbf{\textit{SI}}\ x \text{ est } A\ \textbf{\textit{ALORS}}\ y \text{ est } B \tag{I.5}$$

A partir des valeurs de la première $(x \in A)$ d'une part, et de celle de la conclusion $(y \in B)$ d'autre part, le degré de vérité de la relation floue (R) est déterminé à partir des degrés d'appartenance de x à A et de y à B comme suit :

$$\mu_R(x,y) = imp\left(\mu_A(x), \mu_B(y)\right) \tag{I.6}$$

Les opérateurs les plus utilisés en commande floue sont les implications de Mamdani et de Larsen:

– Implication de Mamdani: $\mu_R(x,y) = \min\left(\mu_A(x), \mu_B(y)\right)$

– Implication de Larsen: $\mu_R(x,y) = \mu_A(x) \times \mu_B(y)$

I.4.2. Structure d'un système flou type-1

Un système flou type-1 peut être interprété selon deux points de vue : mathématique ou logique. D'un point de vue mathématique, c'est une fonction non-linéaire reliant un vecteur de données d'entrée à un vecteur de sortie. D'un de point de vue logique, c'est un système à base de connaissance particulière composé de quatre modules principaux illustrés sur la figure I.3, à savoir: la fuzzification, la base des règles, le moteur d'inférence et la défuzzification [22][44][46].

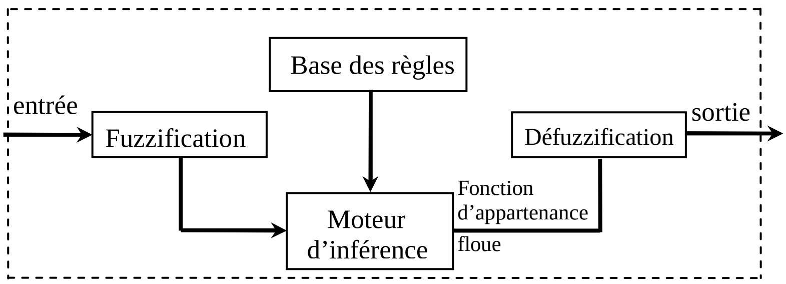


Fig. I.3. Système d'inférence flou type-1

- **Fuzzification:** Consiste à définir des fonctions d'appartenances pour les différentes variables linguistiques. Le but est la conversion d'une grandeur physique en une variable linguistique. Il s'agit d'une projection de la variable physique sur les ensembles flous caractérisant cette variable. Cette opération permet d'avoir une mesure précise sur le degré d'appartenance de la variable d'entrée à chaque ensemble flou. Dans la littérature deux approches de fuzzification sont généralement utilisées, la fuzzification singleton et la fuzzification non-singleton.

- **Base de règles floues:** La base de règles floues ou base de connaissances permet de déduire des connaissances concernant l'état du système en fonction des qualifications linguistiques fournies par l'étape de fuzzification. Généralement la base des règles floues est une collection de règles de la forme :

  SI *<Prémisse* (*antécédent*)> ***ALORS*** *<Conclusion* (*conséquence*)>

  Les règles floues sont déduites des expériences acquises par les opérateurs ou les experts. Ces connaissances sont traduites en règles simples pouvant être utilisées dans un moteur d'inférence floue.

- **Moteur d'inférence:** Ce module est le cerveau du système flou, il consiste à transformer, à l'aide des techniques de raisonnement flou, la partie floue issue de la fuzzification en une nouvelle partie floue. En utilisant les principes de la logique floue, le moteur d'inférence combine les règles floues pour effectuer une transformation à partir des ensembles d'entrée flous vers des ensembles de sortie flous.

- **Défuzzification:** Contrairement au module de fuzzification, le défuzzificateur consiste à transformer la fonction d'appartenance résultante obtenue à la sortie du moteur d'inférence en une valeur précise. Cependant il n'existe pas une procédure systématique pour choisir la stratégie de défuzzification. Un critère de choix des méthodes de défuzzification en commande floue est la simplicité des calculs. Ce critère conduit à l'utilisation de la méthode du centre de gravité.

  - ***Défuzzification par centre de gravité***

    La méthode par centre de gravité ou centroïde consiste à estimer le centre de gravité de la fonction d'appartenance en calculant la moyenne d'un certain nombre de points échantillonnés. La sortie défuzzifiée y_{cg} est donnée par l'abscisse de ce centre où la fonction d'appartenance de l'ensemble de sortie est discrétisée sur N points.

$$y_{cg} = \frac{\sum_{i=1}^{N} y_i \cdot \mu_{A_{res}}(y_i)}{\sum_{i=1}^{N} \mu_{A_{res}}(y_i)} \tag{I.7}$$

où $\mu_{A_{res}}(y_i)$ est le degré d'appartenance résultante et y_i est la variable de sortie.

- ***Défuzzification par centre maximum***

Dans cette méthode, la valeur de sortie est estimée par l'abscisse du point correspondant au centre de l'intervalle pour lequel la fonction d'appartenance est maximale. Cette valeur est fournie par l'expression:

$$y_{cm} = \frac{\inf M + \sup M}{2} \tag{I.8}$$

où M est l'ensemble des points pour lesquels la fonction d'appartenance est maximale.

$$M\left\{ y \in [-c, c] \middle| \mu_{A_{res}}(y) = H(A_{res}) \right\} \tag{I.9}$$

Dans le cas discret, on explore en fait la liste de tous les points pour lesquels la fonction d'appartenance est maximale afin de trouver le plus petit et le plus grand.

- ***Défuzzification par valeur maximum***

Cette méthode ne s'utilise que dans le cas discret. On choisit comme sortie y_m l'abscisse de la valeur maximale de la fonction d'appartenance résultante $\mu_{A_{res}}(y)$. Lorsque $\mu_{A_{res}}(y)$ est écrêtée, on prend la moyenne des abscisses du maximum :

$$y_m = \frac{\sum_{y_i \in M} y_i}{|M|} \tag{I.10}$$

où M est défini dans l'équation (I.9).

I.5. Système d'inférence flou type-2

I.5.1. Ensemble flou type-2

Un ensemble flou type-2 est caractérisé par une fonction d'appartenance floue, la valeur d'appartenance (degré d'appartenance) de chaque élément de l'ensemble est un ensemble flou dans [0, 1]. De tels ensembles peuvent être utilisés dans les situations où nous avons de l'incertitude sur les valeurs d'appartenance elles mêmes. L'incertitude peut être soit dans la forme de la fonction d'appartenance ou dans l'un de ses paramètres. Comme il est mentionné dans la littérature, les éléments du domaine de l'ensemble flou type-2 sont appelés appartenance primaire et l'appartenance de ces appartenances primaires est appelée appartenance secondaire.

I.5.2. Types d'ensemble flous type-2

Selon la forme de l'appartenance primaire, on distingue principalement trois sortes d'ensembles flous type-2.

Ensemble flou type-2 gaussien

Dans ce type d'ensembles, le degré d'appartenance de chaque point est un ensemble type-1 gaussien dont le domaine de définition est inclus dans l'intervalle [0, 1]. Notons qu'il n'est pas nécessaire que la fonction d'appartenance principale soit aussi gaussienne [47].

Ensemble flou type-2 triangulaire

Dans ce type d'ensembles, le degré d'appartenance de chaque point est un ensemble type-1 triangulaire dont le domaine de définition est inclus dans l'intervalle [0, 1] [37].

Ensemble flou type-2 intervalle

Dans ce type d'ensembles, le degré d'appartenance de chaque point est un ensemble ordinaire dont le domaine de définition est inclus dans l'intervalle [0, 1] [37][14]. Dans ce cas, toutes les appartenances secondaires sont égales à 1. Notons que malgré que chaque degré d'un ensemble type-2 intervalle est un ensemble ordinaire, l'ensemble lui-même est de type-2 parce que les degrés d'appartenance sont des ensembles et non pas des nombres ordinaires.

Remarque I.1: il n'existe pas une procédure systématique pour choisir la sortie d'un ensemble flou type-2. Un critère de choix est la simplicité des calculs pour la réduction de type. Ce critère a conduit à l'utilisation de l'ensemble flou type-2 intervalle.

I.5.3. Structure d'un système flou type-2

La structure d'un système flou type-2 est représentée sur la figure (1.4).

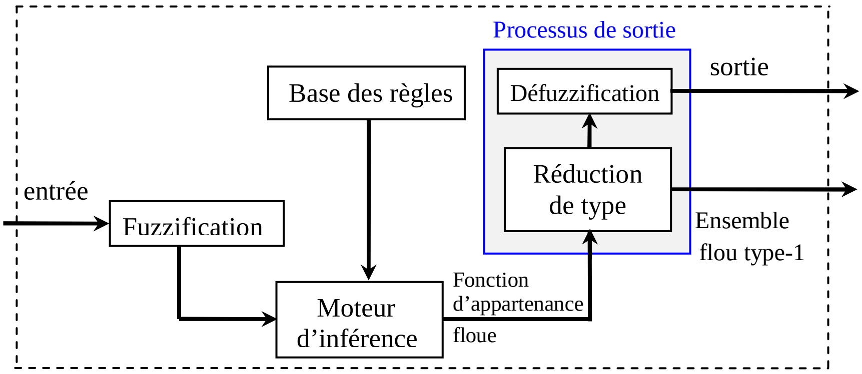


Fig. I.4. Système d'inférence flou type-2 [14]

Cette structure est similaire à celle du type-1, on remarque l'apparition d'un cinquième bloc au niveau du processus de sortie, celui de la réduction de type. Nous supposons dans cette section que les fonctions d'appartenance des prémisses et des conséquences sont de type-2.

- **Fuzzification :** Contrairement à la fonction d'appartenance type-1, la fonction d'appartenance type-2 donne plusieurs degrés d'appartenance (ou dimensions) pour chaque entrée. Par conséquent l'incertitude sera mieux représentée. Cette représentation va nous permettre de tenir compte de ce qui a été négligé dans le type-1. Dans cette thèse, seule la fuzzification de type singleton sera utilisée [38], en d'autres termes, l'entrée floue est un point singulier possédant une valeur d'appartenance unitaire.
- **Base de règles :** La structure d'une règle floue type-2 est la même que celle du type-1, la seule différence est que les ensembles flous associés aux prémisses et aux conclusions du premier sont tous des ensembles flous type-1 ou des nombres certains, alors qu'au moins un ensemble flou du deuxième système, qu'il soit associé à la prémisse ou à la conclusion, soit un ensemble flou de type-2, alors la $j^{ème}$ règle d'un système flou type-2 aura la forme [48][47]:

$$Si\ x_1\ est\ \tilde{F}_1^j\ et\ x_2\ est\ \tilde{F}_2^j\ et\ ....\ et\ x_p\ est\ \tilde{F}_p^j\ Alors\ y\ est\ \tilde{G}^j \tag{I.11}$$

où $x_1, x_2, ..., x_p$ sont les entrées, les $\tilde{F}_i^j$ sont les ensembles des prémisses tel que pour $i = 1, 2, ..., p$. y est la sortie, et les $\tilde{G}^j$ sont les ensembles des conséquences. A noter qu'il n'est pas nécessaire que toutes les fonctions d'appartenance des prémisses et des conséquences soient de type-2. Il suffit qu'une seule fonction d'appartenance dans une prémisse ou dans une conséquence le soit pour que tout le système soit de type-2.

- **Moteur d'inférence flou type-2 :** Dans la logique floue type-2, le processus d'inférence est similaire à son homologue type-1. Ce processus combine les règles et fait correspondre, à travers une fonction, des ensembles flous type-2 de sortie à des ensembles flous type-2 d'entrée. Pour ce faire, l'inférence utilise la base de règles floues (I.11) pour effectuer une relation entre un vecteur d'entrée $\underline{x} = (x_1, ..., x_p)$ et la sortie y. La première étape dans l'opération d'inférence floue est le calcul de l'intervalle d'activation associé à la $j^{ème}$ règle.

$$F(\underline{x}) = \prod_{i=1}^{p} \mu_{\bar{f}_i^j}(x_i) \tag{I.12}$$

où $\mu_{\bar{f}_i^j}(x_i)$ est l'intervalle d'activation associé à la variable x_i .

En notant l'ensemble flou de sortie correspondant à la $j^{ème}$ règle R^j par $\tilde{B}$, lorsqu'une entrée $\underline{x}'$ est appliquée, comme nous utilisons une fuzzification de type singleton qui veut dire que l'ensemble $\tilde{X}'$ auquel appartient $\underline{x}'$ possède un degré d'appartenance unitaire à $\underline{x} = \underline{x}'$ et zéro ailleurs, alors l'ensemble de sortie correspondant à la $j^{ème}$ règle est calculé en utilisant l'implication minimum ou produit (équivalent à l'opération *meet* avec T-norm minimum ou produit dans le cas du type-2) comme suit :

$$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{G}^j}(y) \cap \left[\prod_{i=1}^{p} \mu_{\tilde{F}_i^j}(x_i) \right] \tag{I.13}$$

où $\cap$ dénote l'opération *meet* basée sur la T-norme choisie.

Comme dans le cadre de notre thèse, uniquement les ensembles flous type-2 intervalles sont utilisés en choisissant l'opération T-norm produit, l'intervalle d'activation associé à la $j^{ème}$ règle sera donné par :

$$F^j(\underline{x}) = \left[\underline{f}^j(\underline{x}), \quad \bar{f}^j(\underline{x}) \right] \tag{I.14}$$

où $\underline{f}^j(\underline{x})$ et $\bar{f}^j(\underline{x})$ sont donnés comme suit :

$$\begin{cases} \underline{f}^j(\underline{x}) = \underline{\mu}_{\tilde{F}_1^j}(x_1) \times \underline{\mu}_{\tilde{F}_2^j}(x_2) \times .... \times \underline{\mu}_{\tilde{F}_p^j}(x_p) \\ \bar{f}^j(\underline{x}) = \bar{\mu}_{\tilde{F}_1^j}(x_1) \times \bar{\mu}_{\tilde{F}_2^j}(x_2) \times .... \times \bar{\mu}_{\tilde{F}_p^j}(x_p) \end{cases} \tag{I.15}$$

Les termes $\underline{\mu}_{\tilde{F}_i^j}(x_i)$ et $\bar{\mu}_{\tilde{F}_i^j}(x_i)$ sont respectivement, la valeur inférieure et supérieure de l'intervalle d'activation correspondant à $\mu_{\tilde{F}_i^j}(x_i)$.

- **Processus de sortie :** Le processus de sortie dans un système d'inférence flou type-2 comporte deux parties : le premier est le processus de réduction de type qui est un processus très complexe, notamment pour les ensembles flous généraux, et l'autre est le processus de défuzzification.

Réduction de type

Etant donné que la sortie du système d'inférence est un ensemble flou type-2, elle doit être réduite avant qu'elle soit défuzzifiée. Pour transformer la sortie floue type-2 en un ensemble flou type-1, la méthode des centres d'ensembles (COS : *center of sets*) est utilisée [15]. L'expression de l'ensemble flou de type réduit par cette méthode est donnée par [15][49].

$$Y_{cos}\left(Y^1,...,Y^k,F^1,...,F^k\right)=\int_{y^1}\cdots\int_{y^k}\cdots\int_{f^1}\cdots\int_{f^k}1\Big/\frac{\sum_{j=1}^{k}y^j f^j}{\sum_{j=1}^{k}f^j} \qquad \text{(I.16)}$$

Etant donné que chaque ensemble dans l'équation (I.16) est un ensemble type-1 intervalle, alors $Y_{cos}\left(Y^1,...,Y^k,F^1,...,F^k\right)$ est aussi un ensemble type-1 intervalle dont le domaine est situé sur l'axe des réels :

$$Y_{\cos}=\left(Y^1,\ldots,Y^k,F^1,\ldots,F^k\right)=\left[y_g\left(\underline{x}\right),y_d\left(\underline{x}\right)\right] \qquad \text{(I.17)}$$

$y_g\left(\underline{x}\right)$ et $y_d\left(\underline{x}\right)$ sont deux points de gauche et de droite caractérisant l'ensemble de type réduit Y_{cos}. $f^l\left(\underline{x}\right)$ est $l^{ème}$ élément de l'intervalle d'activation de (I.14) et y^j est un élément de l'intervalle type-2 $Y^j=\left[y_g^j,\ y_d^j\right]$

Afin de calculer les points extrêmes y_g et y_d, Karnik et Mendel [49] ont développé un algorithme itérative dont la procédure est donnée comme suit :

- **Le calcul de** $y_g\left(\underline{x}\right)$ et $y_d\left(\underline{x}\right)$

1. Discrétiser l'espace de sorti Y en un nombre suffisant de points en choisissant comme segment les centroïdes des ensembles flous impliqués dans l'activation des règles.
2. Trier Y^j dans un ordre croissant : $y^1\le y^2\le\cdots\le y^k$.
3. Initialiser f^j en prenant comme point de départ

$$f^j=\frac{\overline{f}^j+\underline{f}^j}{2},\ \ j=1,...,k$$

Ensuite calculer

$$y_g\left(\underline{x}\right)=\frac{\sum_{j=1}^{k} y_g^j f_g^j}{\sum_{j=1}^{k} f_g^j}$$

et poser $y_g'\left(\underline{x}\right)=y_g\left(\underline{x}\right)$.

où f_g^j est l'élément de l'intervalle d'activation (soit $\underline{f}^j$ ou $\overline{f}^j$)

4. Trouver un point de commutation $N\left(1\leq N\leq k-1\right)$ qui satisfait : $y^k\leq y_g\leq y^{k+1}$.

5. Poser

$$f^j=\begin{cases}\overline{f}^j, & j<N\\ \underline{f}^j, & j>N\end{cases}$$

Calculer

$$y_g\left(\underline{x}\right)=\frac{\sum_{j=1}^{k} y_g^j f_g^j}{\sum_{j=1}^{k} f_g^j}$$

et poser $y_g''\left(\underline{x}\right)=y_g\left(\underline{x}\right)$

6. Si $y_g''\left(\underline{x}\right)\neq y_g'\left(\underline{x}\right)$ aller à l'étape 3. Sinon arrêter et poser $y_g\left(\underline{x}\right)=y_g''\left(\underline{x}\right)$.

De la même façon, on détermine

$$y_d\left(\underline{x}\right)=\frac{\sum_{j=1}^{k} y_d^j f_d^j}{\sum_{j=1}^{k} f_d^j}$$

Défuzzification

Le type réduit par l'utilisation de (I.17) sera déterminé par ses deux points extrêmes de droite et de gauche $y_d\left(\underline{x}\right)$ et $y_g\left(\underline{x}\right)$. Une fois le processus de réduction de type aura généré l'ensemble flou type-1 à partir de l'ensemble flou type-2, le processus de défuzzification devient entièrement similaire à celui du système flou type-1 et fera correspondre une valeur certaine à l'ensemble flou type-1 issu du processus de réduction de type.

En appliquant le centre de gravité au type réduit, la sortie numérique sera donnée par [47] :

$$Y=\frac{y_d\left(\underline{x}\right)+y_g\left(\underline{x}\right)}{2} \tag{I.18}$$

I.6. Conclusion

Dans ce chapitre, nous avons abordé principalement deux approximations intelligentes, logiques floues type-1 et type-2. Dans un premier temps, nous avons présenté les notions de base pour un système flou type-1, nécessaires pour la compréhension de notre travail qui est présenté dans les chapitres IV et V. Le fonctionnement d'un système flou type-1 dépend d'un nombre important de paramètres (fonctions d'appartenance, règles floues, règles d'inférence, défuzzification) qu'il faut bien choisir lors de la conception. Dans un deuxième temps, nous avons décrit les notions de base pour un système flou type-2, nécessaires pour la compréhension de notre travail qui est élaboré dans le chapitre VI. Comme elle a été présentée, cette technique est une extension de la logique floue type-1. La nouveauté dans cette logique est que les fonctions d’appartenance ne sont pas définies d’une manière unique, mais d’une manière incertaine. Concernant les systèmes flous type-2, nous avons vu qu’ils sont doté d’un nouveau module appelé réducteur de type, permettant de réduire des ensembles flous type-2 à des ensembles flous type-1. Dans le chapitre suivant, nous présentons les fondements théoriques de quelques techniques d’optimisation métaheuristiques, à savoir les algorithmes génétiques, l'optimisation par essaims de particulaires et l'optimisation à base de biogéographie.

Chapitre II

Méthodes d'optimisation métaheuristiques

Sommaire

Méthodes d'optimisation métaheuristiques

... la zoologie des Archipels vaudra peine d'être examinée ...

MacArthur et Wilson.

II.1. Introduction

Un problème d'optimisation est défini par un ensemble de variables, une fonction objective (fonction de coût) et un ensemble de contraintes. L'espace d'état, appelé aussi domaine de recherche, est l'ensemble des domaines de définition des différentes variables du problème. Il est en général fini, puisque les méthodes opèrent dans des espaces bornés, et pour des raisons pratiques et de temps de calcul, l'espace de recherche doit être fini. Cette dernière limitation ne pose pas de problème, puisqu'en général le décideur précise exactement le domaine de définition de chaque variable. Même dans le cas des problèmes à variables continues, une certaine granularité est définie. La fonction objective définit le but à atteindre, on cherche à minimiser ou à maximiser celle-ci. [50].

Ce chapitre est destiné à exposer quelques fondements théoriques de quelque techniques d'optimisation métaheuristiques, à savoir : les algorithmes génétiques, l'optimisation par essaims des particulaires et l'optimisation à base de biogéographie.

II.2. Les algorithmes génétiques

Le concept d'algorithmes génétiques (GA : *Genetic Algorithms*) a été proposé par Holland et ses collègues à l'Université du Michigan en 1975, pour décrire les systèmes adaptatifs. Les algorithmes génétiques, également appelés algorithme évolutionnaires, sont inspirés du concept de sélection naturelle proposée par Charles Darwin. La solution optimale est recherchée à partir d'une population de solution en utilisant des processus aléatoires.

II.2.1. Principe :

Le principe d'un algorithme génétique est simple. Une population initiale est choisie aléatoirement; chaque individu de la population possède une certaine performance, qui évalue son rang d'adaptation à l'objectif visé (fitness). Pour aller de la génération k à la génération $k+1$, on effectue les opérations suivantes : au départ, la population est reproduite par sélection, ensuite, on applique un croisement aux couples d'individus père et mère (les parents) pour en engendrer des nouveaux individus (les enfants). Un opérateur de mutation est appliqué à une fraction de la population. Enfin, les nouveaux individus sont évalués et intégrés à la population de la génération suivante (de manière à conserver une population de taille constante). D'une génération à une autre, la force des individus de la population grandit et après un certain nombre d'itérations, la population devient entièrement homogène et sera constituée d'individus tous forts, d'où des solutions quasi-optimales du problème posé.

La recherche de la meilleure solution est effectuée en créant une nouvelle génération de solution par application successive, à la population courante, de trois opérateurs : la sélection, le croisement et la mutation. Ces opérations sont répétées jusqu'à ce qu'un critère d'arrêt soit atteint [51][52].

La sélection **:** consiste à sélectionner un individu au sein de la population puis le recopier dans la nouvelle population. La sélection se fait au moyen d'une fonction d'adaptation (fitness fonction) qui est calculée pour chaque individu de la population.

L'opérateur croisement **:** est appliqué sur des pères d'individus tirés aléatoirement. Il consiste en un échange partiel de leurs caractéristiques. Par ce biais, les gènes sont transférés d'un individu à l'autre est chacun des deux nouveaux individus hérite partiellement des caractéristiques de ses parents. Les positions à croiser sont tirées aléatoirement.

La mutation : la mutation met en jeu un seul individu. Ce processus provoque le changement de valeur de certains caractères au sein de la chaîne. Ceci peut provoquer tant amélioration qu'une diminution de la qualité de l'individu.

L'algorithme de l'AG est présenté ci-dessous :

Algorithme II.1 : L'algorithme AG

```
Initialisation de la population
Evaluer des fonctions objectives de chaque individu
Calculer l'efficacité de chaque individu
Tant que le critère d'arrêt n'est pas atteint faire
    | -Sélection des parents pour la reproduction
    | -Mutation de séquences tirées de manière aléatoire
    | -Croisement de paires de séquences choisies aléatoirement
    | -Evaluer des fonctions objectives de chaque enfant
    | -Calculer l'efficacité de chaque enfant
    | -Sélection pour remplacement
Fin tant que
```

Il existe différentes façons de coder une solution. Le codage le plus utilisé en pratique est le codage binaire dans lequel chaque solution est représentée par une chaîne de bits (acceptant les valeurs 0 ou 1). Le codage réel est une alternative au codage binaire, dans ce cas, les variables réelles se montrent directement dans le chromosome et sont utilisées par des opérateurs génétiques simples et spéciaux (expressions mathématiques).

II.3. Algorithme d'optimisation par essaim particulaire

L'optimisation par essaim particulaire est une métaheuristique d'optimisation née en 1995 aux États-Unis sous le nom de (PSO: *Particle Swarm Optimization*). Elle a été inventée par Russel Eberhart et James Kennedy [53]. Elle s'inspire des déplacements collectifs observés chez certains animaux sociaux tels que les poissons et les oiseaux migrateurs qui ont tendance à imiter les comportements réussis qu'ils observent dans leur entourage, tout en y apportant leurs variations personnelles. L'origine de cette méthode vient des observations faites lors des simulations informatiques des vols groupés d'oiseaux et de bancs de poissons [54][55]. Ces simulations ont mis en valeur la capacité des individus d'un groupe en mouvement à conserver une distance optimale entre eux et à suivre un mouvement global par rapport aux mouvements locaux de leur voisinage.

Le PSO est un algorithme à population, contrairement aux algorithmes basés sur des opérateurs génétiques, il ne se base pas sur la sélection, le croisement des meilleurs individus mais sur la collaboration entre eux, il possède de mémoire.

II.3.1. Principe :

L'optimisation par essaim de particules repose sur un ensemble d'individus originellement disposés de façon aléatoire et homogène, que nous appellerons dès lors de particules, qu'elles se déplacent dans l'hyper-espace de recherche et constituent, chacune, une solution potentielle Chaque particule dispose d'une mémoire concernant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules constituant son entourage. A partir de ces informations, la particule va suivre une tendance faite, d'une part, de sa volonté à retourner vers sa solution optimale, et d'autre part, de son mimétisme par rapport aux solutions trouvées dans son voisinage. A partir d'optimums locaux et empiriques, l'ensemble de particules va, normalement, converger vers la solution optimale globale du problème traité.

L'algorithme commence avec une initialisation aléatoire de l'essaim de particules dans l'espace de recherche. Chaque particule est modélisée par sa position dans l'espace de recherche et par sa vitesse. À chaque instant, toutes les particules ajustent leurs positions et vitesses, donc leurs trajectoires, par rapport à leurs meilleures positions, à la particule ayant la meilleure position dans l'essaim et à leur position actuelle. En réalité, chaque particule est influencée, non seulement par sa propre expérience, mais aussi par celle des autres particules.

La position et la vitesse d'une particule dans un espace de recherche à N dimensions sont définies par : $p_i = (p_1, ..., p_N)$ et $v_i = (v_1, ..., v_N)$, respectivement. Chaque particule est caractérisée par sa meilleure position $P_{bi} = (p_{b,1}, ..., p_{b,N})$ à l'itération k. La meilleure position qu'atteint l'essaim est sauvegardée dans le vecteur $P_g = (p_{g,1}, ..., p_{g,N})$. La vitesse de chaque particule est mise à jour selon l'expression suivante :

$$v_i(k+1) = w.v_i(k) + c_1.r_1(k).(p_{bi}(k) - p_i(k)) + c_2.r_2(k).(p_g(k) - p_i(k)) \tag{II.1}$$

L'équation (II.1) met à jour la vélocité à partir de la vitesse précédente à la nouvelle. La position à l'itération $(k+1)$ est alors déterminée par la somme de la position précédente et de la nouvelle vitesse obtenue :

$$p_i(k+1) = p_i(k) + v_i(k+1) \tag{II.2}$$

où c_1 et c_2 sont des constantes appelées coefficients d'accélération, r_1 et r_2 sont des nombres aléatoires uniformément distribués dans l'intervalle $[0,\ 1]$. w est le facteur d'inertie et peut être déterminé par :

$$w = w_{\max} - (w_{\max} - w_{\min})\, itr / itr_{\max} \tag{II.3}$$

où $itr_{\max}$ est le nombre maximal d'itération, itr est le nombre actuel d'itération.

L'algorithme de PSO est présenté ci-dessous :

Algorithme II.2 : L'algorithme PSO

Initialiser la population de particules avec des positions et vitesses aléatoires
Evaluer la fonction objectif pour chaque particule et calculer P_g
Pour chaque individu i, P_{bi} *est initialisée à* P_i
***Répéter** jusqu'au critère d'arrêt*
  -Mettre à jour les vitesses et les positions de particules selon les équations (II.1) et (II.2), respectivement
  -Evaluer la fonction objective pour chaque individu
  -Calculer les nouveaux P_{bi} *et* P_i
  -Sélection pour remplacement
Jusqu'à (*critère d'arrêt***)**
Afficher le meilleur état rencontré au cours de la recherche.

II.4. Algorithme d'optimisation basée sur la biogéographie

L'algorithme à base de biogéographie (BBO : *Biogeography-Based Optimization*), développé par Dan Simon en 2008 [56], trouve ses origines dans la théorie d'équilibre dynamique (appelée aussi théorie de la biogéographie insulaire), énoncée par MacArthur et Wilson [57].

La théorie de la biogéographie consiste en l'étude de la répartition spatiale des espèces vivantes (végétales et animales) et des causes de cette répartition. Elle traite de la façon dont la richesse des espèces (nombre d'espèces) est maintenue dans un système d'île qui est sujette à l'immigration et sur lesquelles des espèces s'éteignent [58]. Elle stipule que les milieux insulaires sont à l'origine vides d'espèces et que celles-ci y arrivent peu à peu en provenance de régions vastes (désignées sous le terme de « continents », bien qu'il ne s'agisse pas forcément de continents à proprement parler) ou d'îles voisines. Certaines espèces sont d'ailleurs mieux outillées que d'autres pour conquérir de nouveaux territoires, elles ont donc des capacités de colonisation des milieux insulaires plus grandes que d'autres. Les interactions compétitives sur l'île tendent par contre à accélérer les extinctions. Le croisement de ces deux processus dynamique permet d'expliquer la richesse actuelle du peuplement. À l'équilibre, il y a un remplacement constant des espèces.

L'algorithme BBO manipule une population d'individus appelés îles (ou *habitats*). Chaque île représente une solution possible au problème à résoudre. La « fitness » de chaque île est déterminée par son HSI (*Habitat Suitability Index*), une mesure de la qualité d'une solution candidate, et chaque île est représentée par des SIVs (*Suitability Index Variables*).

Une bonne solution au problème d'optimisation est une île avec un grand nombre d'espèces, ce qui correspond à une île avec un faible HSI.

Selon la théorie de MacArthur et Wilson, le nombre d'espèces présentes sur une île dépend essentiellement d'un équilibre entre le taux d'immigration de nouvelles espèces et le taux d'émigration des espèces déjà établies sur l'île. Dans BBO, chaque habitat a son propre taux d'immigration (λ) – arrivées venant de l'extérieur – et son taux d'émigration (μ) – départs vers l'extérieur. Ces paramètres sont influencés par le nombre d'espèces (s) présentes sur l'île.

Le taux d'immigration (λ) décroît avec l'augmentation du nombre d'espèces (s) déjà présentes sur l'île. Plus le nombre d'espèces déjà installées sur l'île augmente, de moins en moins d'immigrants appartenant à une nouvelle espèce rejoignent l'île. Mais, au fur et à mesure que le nombre d'espèces déjà présentes sur l'île diminue, plus le taux d'immigration augmente. Le taux d'immigration maximale (I) est atteint lorsque l'île est vide. Une fois que toutes les espèces sont présentes sur l'île, alors $s = s_{\max}$ (capacité maximale de l'île) et le taux d'immigration tombe à zéro, ne favorisant plus l'installation de nouveaux arrivants (plus l'île est peuplée, moins les espèces étrangères ont de chances de s'y implanter). Le taux d'immigration, quand il y a s espèces sur l'île, est donné par :

$$\lambda = I\left(1 - \frac{s}{s_{\max}}\right) \tag{II.4}$$

Le taux d'émigration (μ) augmente avec le nombre d'espèces (s) présentes sur l'île. Le taux d'émigration maximum (E) se produit lorsque toutes les espèces sont présentes sur l'île ($s = s_{\max}$), et devient nul si les espèces présentes sur l'île s'éteignent (ou quittent l'île). Le taux d'émigration quand il y a s espèces sur l'île est donné par :

$$\mu = E\left(\frac{s}{s_{\max}}\right) \tag{II.5}$$

La figure II.1 représente graphiquement le modèle d'équilibre du nombre d'espèces sur les îles. Le nombre d'espèces déjà établies sur une île a un effet négatif sur l'immigration (compétiteurs, prédateurs et parasites déjà présents, moins d'espèces qui restent à immigrer), et un effet positif sur l'émigration (moins de ressources par espèce, forte compétition interspécifique).

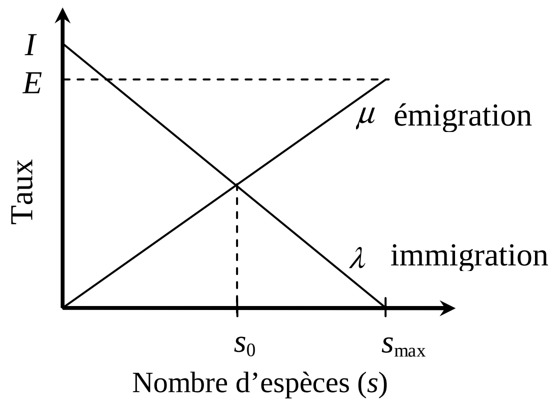


Fig. II.1. Modèle linéaire de taux d'émigration (μ) et le taux d'immigration (λ) [56]

Le taux d'immigration chute rapidement au début lorsque les meilleurs colonisateurs

s'établissent sur l'île. Le taux d'émigration s'accroît plus rapidement avec un nombre élevé d'espèces déjà présentes sur l'île. Le nombre d'espèces à l'équilibre sur l'île (s_0) est déterminé par l'intersection des courbes d'émigration (E) et d'immigration (I). Le modèle de la figure II.1 représente l'évolution du taux d'immigration (resp. d'émigration) par une fonction linéaire décroissante (resp. croissante) du nombre d'espèces présentes sur l'île. Il existe toutefois différents modèles mathématiques de la biogéographie, qui comprend des variables plus complexes [57]. Il y a, en effet, d'autres facteurs importants qui influencent les taux de migration entre les habitats, tels que la distance entre les habitats, la taille de l'habitat, variations climatiques (pluviométrie, température), la diversité végétale et animale, en plus de l'activité humaine. Ces facteurs rendent les courbes d'immigration et d'émigration plus complexes, contrairement à celles décrites dans le document original sur BBO [56].
Pour examiner l'influence de différents modèles de migration sur les performances de BBO, Haiping Ma [59] a étudié le comportement de six modèles de migration. Les résultats expérimentaux montrent clairement que les modèles de migration les plus proches de la nature (c'est-à-dire, non-linéaires) sont nettement mieux que les modèles linéaires.

Considérons à présent la probabilité P_i que l'île abrite exactement i espèces. Le nombre des espèces change pendant l'intervalle de temps $[t, t+\Delta t[$ selon l'équation suivante :

$$P_i(t+\Delta t) = P_i(t)(1-\lambda_i\,\Delta t - \mu_i\,\Delta t) + P_{i-1}\,\lambda_{i-1}\,\Delta t + P_{i+1}\,\mu_{i+1}\,\Delta t \tag{II.6}$$

L'équation (II.6) stipule que le nombre des espèces sur l'île dépend du nombre total des espèces déjà établies sur l'île, de la fréquence à laquelle les nouvelles espèces arrivent et de la fréquence à laquelle les anciennes disparaissent. Nous supposons ici que Δt est assez petit pour que la probabilité que deux changements ou plus se produisent pendant un tel intervalle est nulle. Afin de disposer de (i) espèces à l'instant $(t+\Delta t)$, l'une des conditions suivantes doit être remplie :

– Il y a i espèces à l'instant t, et aucune immigration ni émigration n'a eu lieu entre l'instant t et l'instant $t+\Delta t$;
– Il y a $i-1$ espèces sur l'île à l'instant t, et une nouvelle espèce s'y installe.
– Il y a $i+1$ espèces sur l'île à l'instant t, et une espèce quitte l'île.

La limite de (II.6) quand $\Delta t \to 0$ est donnée par l'équation (II.7).

$$\dot{P}_i = \begin{cases} -\left(\lambda_i + \mu_i\right) P_i + \mu_{i+1} P_{i+1} & \text{si} \quad i = 0 \\ -\left(\lambda_i + \mu_i\right) P_i + \lambda_{i-1} P_{i-1} + \mu_{i+1} P_{i+1} & \text{si} \quad 1 \leq i \leq s_{\max} - 1 \\ -\left(\lambda_i + \mu_i\right) P_i + \lambda_{i-1} P_{i-1} & \text{si} \quad i = s_{\max} \end{cases} \tag{II.7}$$

où $\dot{P}_i$ est le changement de probabilité d'existence de l'habitat i.

L'algorithme BBO peut être décrit globalement par l'algorithme II.3. Les deux opérateurs de base qui régissent le fonctionnement de BBO sont la migration et la mutation. En plus, une stratégie d'élitisme est adoptée dans l'algorithme BBO, afin de garder dans la nouvelle population la meilleure solution.

Algorithme II.3 : L'algorithme BBO

Générer aléatoirement un ensemble de solutions initiales (îles)
***Tant que** le critère d'arrêt n'est pas atteint faire*
  Évaluer la fitness (HSI) de chaque solution
  Calculer le nombre d'espèce s, le taux d'immigration λ et d'émigration μ pour chaque solution
  Migration :
  pour** $i = 1$ à NP **faire
    Utiliser λ_i pour décider, de manière probabiliste, d'immigrer à X_i
    si** rand(0, 1) < λ_i **alors
      pour** $j = 1$ à NP **faire
        Sélectionner l'île d'émigration X_j avec une probabilité α μ_j
        si** rand(0, 1) < μ_j **alors
          Remplacer une variable de décision (SIV) choisie aléatoire-ment dans X_i par la variable correspondante dans X_j
        fin
      fin
    fin
  fin
  ***Mutation :** Muter les individus au taux de mutation donné par l'équation* (II.8).
  Remplacement de la population par les descendants
  Implémenter l'élitisme
fin
retourner la meilleure solution trouvée

L'idée générale de la migration est l'échange de caractéristiques entre les îles. Les taux d'immigration (λ) et d'émigration (μ) de chaque île sont utilisés pour transmettre, de manière probabiliste, les caractéristiques entre les îles, $rand\left(0, 1\right)$ est un nombre aléatoire uniformément distribué dans l'intervalle [0, 1] et $X_{i,j}$ est le $j^{ème}$ SIV de la solution X_i. La stratégie de migration de BBO est similaire à la recombinaison des stratégies d'évolution [60], dans laquelle plusieurs parents sont recombinés entre eux pour former un unique enfant. La principale différence réside dans le fait que la recombinaison est utilisée pour créer de nouvelles solutions, tandis que la migration est utilisée pour modifier des solutions existantes.

Le HSI d'une île peut changer brusquement, en raison d'événements aléatoires : des catastrophes naturelles (tempêtes, ouragans, incendies, etc.) ou des épidémies, etc. BBO modélise ce phénomène comme une mutation des SIVs, et utilise les probabilités de nombre d'espèces (species count probabilités P_i) pour déterminer les taux de mutation. La mutation est utilisée pour améliorer la diversité de la population, empêchant ainsi la recherche de stagner. La probabilité, qu'une solution donnée i, existe a priori comme une solution pour le problème considéré, est spécifiée par la probabilité du nombre d'espèces (P_i). Si une île i est sélectionnée pour la mutation, alors une variable SIV est modifiée de façon aléatoire en fonction de sa probabilité P_i. Dans ce contexte, il convient de remarquer que des solutions avec des valeurs de HSI très élevées ou très faibles ont une faible probabilité d'exister. Tandis que les solutions avec un HSI moyen sont relativement probables. Si une solution donnée a une probabilité faible, elle est susceptible d'être mutée à une autre solution. A l'inverse, une solution avec une forte probabilité est en revanche moins susceptible d'être mutée.
Le taux de mutation m_i pour l'habitat i est calculé en fonction de la probabilité de la solution exprimée dans l'équation (II.8).

$$m_i = m_{\max}\left(\frac{1-P_i}{P_{\max}}\right) \tag{II.8}$$

où $m_{\max}$ est le taux maximum de mutation, et $P_{\max} = \underset{i}{argmax}\ P_i \ , i=1,\dots,s_{\max}$. Si une île est sélectionnée pour la mutation, alors un SIV choisi au hasard dans l'île est simplement remplacé par une variable aléatoire générée dans son intervalle de définition.

II.5. Conclusion

Dans ce chapitre, nous avons décrit brièvement le principe des algorithmes génétiques, d'algorithme d'optimisation par essaime de particules et d'algorithme d'optimisation basé sur la biogéographie, qui permettent d'explorer de façon très efficace l'espace des solutions possibles d'un problème. Nous verrons dans les chapitres V et VI de cette thèse, comment on peut utiliser les algorithmes, PSO et BBO pour ajuster les paramètres du filtre de Kalman étendu afin d'améliorer la performance d'estimation des états du système à commander.

Chapitre III

Modélisation et commande des systèmes non-linéaires

Sommaire

Modélisation et commande des systèmes non-linéaires

La théorie, c'est quand on sait tout et que rien ne fonctionne. La pratique, c'est quand tout fonctionne et que personne ne sait pourquoi.
Albert Einstein.

III.1. Introduction

La première partie de ce chapitre a pour objectif de présenter quelques rappels, nécessaires et indispensables à la compréhension du contenu de cette thèse. Nous fournissons au début une introduction aux systèmes robotiques et nous présentons quelques notions de stabilité des systèmes non-linéaires. Puis, nous exposons un rappel sur la modélisation, représentation entrée-sortie des systèmes non-linéaires. Enfin, nous présentons les fondements et les aspects théoriques de quelques approches de commande non-linéaires telles que la commande linéarisante, la commande par mode glissant et la commande backstepping.

Dans une deuxième partie, nous présentons quelques approches qui ont été très étudiées dans la littérature au cours de ces dernières années, concernant l'estimation d'état des systèmes non-linéaires, telles que l'observateur à mode glissant et le filtre de Kalman étendu.

III.2. Systèmes robotiques commandés

La robotique est un domaine de la technologie moderne qui traverse relativement les limites de l'ingénierie traditionnelle. De nouvelles disciplines de l'ingénierie, telles que l'ingénierie de fabrication, l'ingénierie des applications et l'ingénierie des connaissances ont émergés pour faire face à la complexité du domaine de la robotique.

Les systèmes robotiques intéressent de très nombreux domaines civils et militaires. Les champs d'application de la robotique sont :

1) La production manufacturière (usinage, assemblage, soudure, etc.)
2) Les interventions en milieux hostiles (sous-marin, exploration planétaire, drone, manipulateur mobile, etc.)

3) es systèmes de transport des biens et des personnes (véhicules intelligents, robots mobiles, etc.)
4) Les domaines de la santé (robots de chirurgie, rééducation, etc.)
5) L'aide et l'assistance aux personnes (robots personnels et de service, etc.)

Comprendre la complexité des robots et leurs applications nécessite des connaissances en génie électrique, génie mécanique, génie des systèmes industriels, automatique, informatique, économie et mathématiques.

Les systèmes robotiques sont par nature des systèmes non-linéaires, ils se trouvent sous plusieurs formes, robots aéronefs, robots manipulateurs, robots mobiles et sont régis par des phénomènes physiques non-linéaires. Chargés mécaniquement, ils subissent les non linéarités des systèmes qu'ils entraînent, ce qui va se répercuter sur les grandeurs de réglage.

Un système robotique commandé est un ensemble d'équations différentielles non-linéaires, décrivant l'évolution temporelle des variables constitutives du système sous l'action d'un nombre fini de variables indépendantes appelées entrées ou variables de commande, ou simplement commandes, que l'on peut choisir librement pour réaliser certains objectifs.

III.3. Stabilité des systèmes non-linéaires

III.3.1. Définitions

Définition III.1 ***(Système autonome)*** *Un système dynamique non-linéaire est dit autonome lorsqu'il ne dépend pas explicitement du temps. Un système autonome est donné ci-dessous :*

$$\dot{x} = f(x) \tag{III.1}$$

où $x \in \mathbb{R}^n$*. Un système autonome est indépendant du temps initial, alors qu'un système non autonome ne l'est pas. Dans un système autonome, tout instant peut être considéré comme instant initial, et tout état* $x(t)$ *du système peut être considéré comme un état initial.*

Définition III.2 ***(Point d'équilibre)*** *On appelle point d'équilibre du système (III.1), le point* x_e *tel que :*

$$f(x_e) = 0$$

III.3.2. Stabilité d'un point d'équilibre

Soit le système autonome (III.1), où $f : D \to \mathbb{R}^n$ est une projection localement Lipschitz de $D \subset \mathbb{R}^n$. On suppose que le point x_e est le point d'équilibre, c'est à dire : $f\left(x_e\right) = 0$

Définition III.3 ***(Stabilité au sens de Lyapunov)*** *Le point d'équilibre x_e est dit stable si $\forall \varepsilon > 0, \exists \delta > 0$ tel que :*

$$\|x(0) - x_e\| < \delta \;\Rightarrow\; \|x(t) - x_e\| < \varepsilon, \forall t \geq 0$$

Définition III.4 ***(Instabilité)*** *Le point d'équilibre x_e est dit instable s'il n'est pas stable au sens de Lyapunov.*

Définition III.5 ***(Stabilité asymptotique)*** *Le point d'équilibre x_e est asymptotiquement stable s'il est stable et si l'on peut choisir $\varepsilon > 0$ tel que :*

$$\|x(0) - x_e\| < \varepsilon \;\Rightarrow\; \lim_{t \to \infty} x(t) = x_e$$

La stabilité asymptotique signifie qu'on peut déterminer un voisinage du point d'équilibre tel que n'importe quelle trajectoire, issue d'un point $x(0)$ appartenant à un voisinage de x_e tends vers x_e lorsque $t \to \infty$.

Définition III.6 ***(Stabilité exponentielle)*** *Le point d'équilibre x_e est exponentiellement stable si $\alpha > 0$ et $\lambda > 0$ tel que :*

$$\forall t > 0, \exists B_r\left(x_e, r\right), \forall x\left(0\right) \in B_r, \quad \|x(t) - x_e\| \prec \alpha \|x(0) - x_e\| < e^{-\lambda t}$$

dans lequel $B_r = \left\{x \in \mathbb{R}^n / \|x(t)\| \leq r\right\}$, et $\|.\|$ *est une norme sur $\mathbb{R}^n$. Dans ce cas λ est appelé "taux de convergence".*

Définition III.7 ***(Stabilité globale)*** *Si le système est asymptotiquement (exponentiellement) stable quel que soit $x(0)$, le point d'équilibre est dit globalement (exponentiellement) stable.*

Définition III.8 ***(fonction de Lyapunov)***

La fonction $L : U \subseteq \mathbb{R}^n \to \mathbb{R}^+$ est une fonction candidate de Lyapunov si elle satisfait les deux conditions suivantes :

– $L(x)$ *est continue et ses dérivées partielles* $\dfrac{\partial L(x)}{\partial x_i}, \forall i = 1,\ldots,n$ *, existent et sont continues.*

– $L(x)$ *est définie positive, soit* $L(x) > 0,\ \forall x \neq 0$ *et* $L(0) = 0$.

D'après la seconde méthode de Lyapunov, la stabilité des systèmes non-linéaires peut être caractérisée par l'existence d'une fonction de Lyapunov contractive le long des trajectoires stables.

III.3.3. Méthode directe de Lyapunov

La méthode directe de Lyapunov (appelée aussi la seconde méthode) permet d'analyser la stabilité d'un système autour de son point d'équilibre sans avoir besoin de chercher les solutions de celui-ci [61]. Cette méthode repose sur l'existence d'une fonction particulière $L(x,t)$ définie positive, appelée fonction de Lyapunov.

Soit le système autonome (III.1), on suppose que le point x_e est le point d'équilibre.

Théorème III.1

S'il existe une fonction $L : U \to \mathbb{R}$*, continue sur un voisinage* U *de* x_e *et différentiable telle que:*

1) $\dot{L}(x_e) = 0$ et $L(x) > 0$ *si* $x \neq x_e$,
2) $\dot{L}(x) \leq 0,\ \forall x \in U$

alors x_e *est un point d'équilibre stable pour le système (III.1)*

3) *Si de plus, la fonction* L *est telle que :*

$$\dot{L}(x) < 0,\ \forall x \in U$$

alors x_e *est un point d'équilibre asymptotiquement stable.*

III.4. Modélisation des systèmes non-linéaires

La modélisation du processus est une étape primordiale dans la mise en œuvre d'un contrôleur dans les systèmes robotiques. La nature non-linéaire et la complexité de ces systèmes rendent leur description analytique, avec des équations différentielles, très difficile.

En général, un système robotique ayant pour entrée (commande) u et comme sortie y peut être décrit par :

$$\begin{cases} x^{(n)} = F(x, \dot{x}, ..., x^{(n-1)}, u) \\ y = H(x, \dot{x}, ..., x^{(n-1)}) \end{cases} \tag{III.2}$$

où $(x, \dot{x}, ..., x^{(n-1)})$ est le vecteur d'état, F et H sont des fonctions non-linéaires continues, généralement partiellement ou totalement inconnues.

Cette description ne permet pas la mise en œuvre de contrôleurs pour assurer la régulation ou l'asservissement. Afin de contourner ce problème, la linéarisation entrée-sortie a été largement utilisée; il s'agit de trouver, à l'aide de la théorie de la géométrie différentielle, une relation explicite entre l'entrée du système et sa sortie [1][62]. Dans ce cas, un système d'ordre n "affine en la commande" peut être décrit par les équations différentielles suivantes :

$$\begin{cases} \dot{X} = f(X) + g(X)u \\ y = h(X) \end{cases} \tag{III.3}$$

où $X = (x, \dot{x}, ..., x^{(n-1)})^T$ est le vecteur d'état. (f, g et h) sont des fonctions non-linéaires continues.

Cette nouvelle description permet d'utiliser facilement les différentes approches basées sur la rétroaction pour résoudre les problèmes de poursuite de trajectoire ou de régulation. Rappelons qu'un système non-linéaire est un système qui ne satisfait pas le principe de superposition où $f(X)$ et $g(X)$ sont des fonctions non-linéaires supposées connues. Elles peuvent être de formes particulières relativement simples, déterminant des classes importantes qui peuvent être déduites de tels systèmes dynamiques, à savoir les systèmes non-linéaires affines en commande [1].

III.5. Représentation entrée-sortie d'un système non-linéaire

Nous considérons la classe des systèmes non-linéaires mono-variables affines en commande décrits par (III.3). Il est clair que ce modèle d'état ne présente pas une relation directe entre la commande et la sortie, ce qui ne facilite pas la synthèse d'une loi de commande. Pour extraire la relation directe et explicite entre la sortie et l'entrée de commande, il est nécessaire alors d'introduire une transformation de la représentation d'état en une représentation entrée-sortie [1]. L'objectif de la représentation entrée-sortie est de trouver une relation directe entre la sortie du système (III.3) et son entrée.

Pour exprimer explicitement cette relation, il est nécessaire de dériver itérativement la sortie du système jusqu'à l'apparition de la commande u. On obtient alors :

$$\dot{y} = \frac{\partial h}{\partial X} f(X) + \frac{\partial h}{\partial X} g(X) u \tag{III.4}$$

L'équation (III.4) indique qu'il y'aura présence de l'influence de l'entrée u lorsque $(\partial h / \partial X) g(X) \neq 0$.

En utilisant la notion de dérivée de Lie présentée dans l'annexe C, on peut écrire plus succinctement la dérivée temporelle de la sortie sous la forme [1]:

$$\begin{cases} y = h(X) \\ \dot{y} = L_f\, h(X) \\ \vdots \\ y^{(\gamma)} = L_f^{\gamma} h(X) + L_g L_f^{\gamma-1} h(X) u \end{cases} \tag{III.5}$$

où $L_f\, h(X) : \mathbb{R}^n \to \mathbb{R}$ et $L_g\, h(X) : \mathbb{R}^n \to \mathbb{R}$ sont définies comme étant les dérivées de Lie de $h(X)$ par rapport à $f(X)$ et $g(X)$, respectivement.

Soit U un ensemble ouvert contenant le point d'équilibre X_0, c'est-à-dire un point où $f(X)$ devient nul ($f(X_0) = 0$). Le degré relatif du système (III.5) est défini comme le nombre de fois que la sortie doit être différenciée avant que l'entrée n'apparaisse dans son expression. En général, le degré relatif d'un système non-linéaire à $X_0 \in U$ est défini comme un entier γ satisfaisant:

$$\begin{aligned} & L_g L_f^{i} h(X) \equiv 0,\ \forall X \in U,\ i = 0, \ldots, \gamma - 2 \\ & L_g L_f^{\gamma-1} h(X_0) \neq 0 \end{aligned} \tag{III.6}$$

Dans le cas où $n = \gamma$, le système (III.3) est dit dans sa forme normale et ne présente pas de dynamique zéro, il est alors à minimum de phase.

En introduisant un changement de variable, l'équation (III.5) peut s'écrire comme suit :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \vdots \\ \dot{z}_n = \lambda_1(z_1,...,z_n) + \lambda_2(z_1,...,z_n)u \\ y = z_1 \end{cases} \tag{III.7}$$

où $\lambda_1(z) = L_f^{(n)} h\left(\phi^{-1}(z)\right)$ et $\lambda_2(z) = L_g L_f^{(n-1)} h\left(\phi^{-1}(z)\right)$

avec :

$$\phi(X) = \begin{bmatrix} \phi_1(X) = h(X) \\ \vdots \\ \phi_n(X) = L_f^{(n-1)} h(X) \end{bmatrix} \text{ et } z = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} \tag{III.8}$$

et peut être représenté schématiquement comme suit :

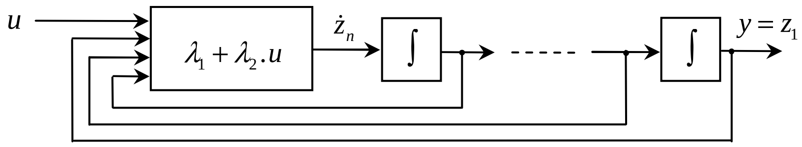


Fig. III.1. Représentation entrée-sortie du système (III.7) [1]

Dans le cas où le degré relatif est strictement inférieur à l'ordre du système, il est toujours possible de choisir $(n-\gamma)$ fonctions $\left(\phi_1(X),...,\phi_n(X)\right)$ afin d'assurer que le difféomorphisme $\phi(X) = \left[\phi_1(X),...,\phi_n(X)\right]^T$ ait un Jacobien non singulier.

Les fonctions $\phi_1(X),...,\phi_n(X)$ peuvent être choisies telles que [1]:

$$L_g \phi_i(X) = 0 \text{ pour } \gamma + 1 \leq i \leq n \tag{III.9}$$

Dans ce cas, le système (III.7) peut s'écrire comme suit :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \vdots \\ \dot{z}_\gamma = \lambda_1(z) + \lambda_2(z)u \\ \dot{z}_{\gamma+1} = q_{\gamma+1}(z) \\ \vdots \\ \dot{z}_n = q_n(z) \\ y = z_1 \end{cases} \tag{III.10}$$

où : $q_i(z) = L_f\phi_i\left(\phi^{-1}(z)\right)$ pour $\gamma+1 \le i \le n$

La représentation fonctionnelle du système (III.10) est illustrée sur la figure III.2.

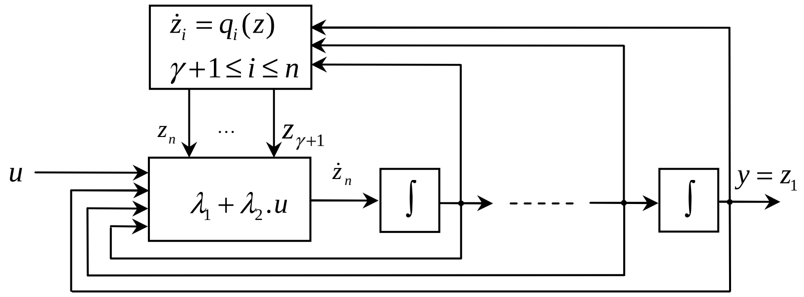


Fig. III.2. Représentation entrée-sortie du système (III.10)

Si l'on considère les systèmes ayant un degré relatif $\gamma = n$ et qui s'écrivent sous la forme canonique

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_n = f(x_1,...,x_n) + g(x_1,...,x_n)u \\ y = x_1 \end{cases} \tag{III.11}$$

On peut alors facilement vérifier que :

$$\begin{aligned} y^{(n)} &= f(X) + g(X)u \\ &= f\left(y, \dot{y}, ..., y^{(n-1)}\right) + g\left(y, \dot{y}, ..., y^{(n-1)}\right)u \end{aligned} \tag{III.12}$$

Le vecteur d'état étant composé des dérivées successives de la sortie, la représentation entrée-sortie du système (III.12) est illustrée sur la figure III.3.

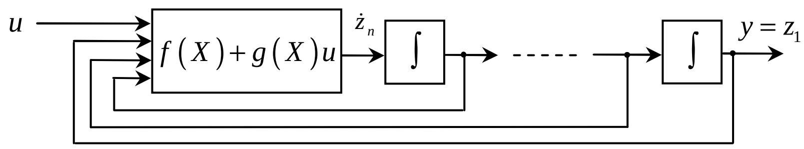


Fig. III.3. Représentation entrée-sortie du système (III.12)

III.6. Commande des procédés non-linéaires

Parmi les nombreuses techniques de commande non-linéaires, certaines ont fait l'objet de théories poussées. Dans ce paragraphe, nous rappelons les fondements et les aspects théoriques de la commande linéarisante, la commande par mode glissant et la commande backstepping.

III.6.1. Commande linéarisante

Les méthodes dites de «linéarisation par bouclage» ou la commande linéarisante sont des méthodes de synthèse classiques pour les systèmes non-linéaires [1][3]. Dans ces méthodes, pour satisfaire un cahier de charges généralement simple (stabilité, suivi de référence ou rejet de perturbations), la loi de commande est choisie pour compenser les non linéarités [3], nous allons donc parler de méthode de type «compensation» des non linéarités. Cela conduit à un système linéaire stationnaire par un premier bouclage. Dans un deuxième temps, un correcteur est synthétisé pour ce système linéaire stationnaire, pour assurer les propriétés du cahier de charges. Cette étape peut être traitée par toute méthode de synthèse pour les systèmes linéaires stationnaires. Ce principe est illustré sur la figure III.4.

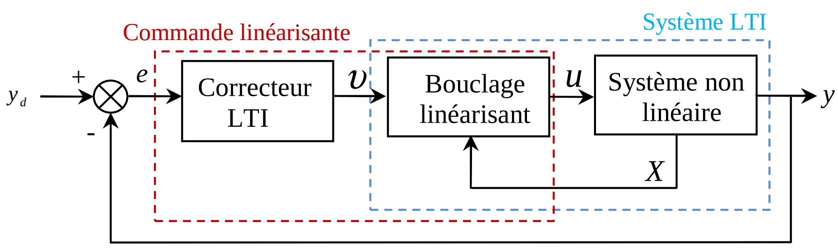


Fig. III.4. Diagramme schématique du principe de la commande linéarisante

En se basant sur la représentation entrée-sortie, une linéarisation du système par bouclage des états est possible. L'objectif est alors de trouver une loi de commande $u(t)$ telle que la boucle fermée reste stable et que la sortie du système $y(t)$ suive la trajectoire désiré $y_d(t)$. Le système (III.11) peut être réécrit sous la représentation d'état suivante :

$$\begin{aligned} \dot{X} &= A.X + B\left[f(X) + g(X)u \right] \\ y &= C^T.X \end{aligned} \qquad \text{(III.13)}$$

où

$$A = \begin{bmatrix} 0 & & I_{(n-1)} \\ 0 & \cdots & 0 \end{bmatrix},\ B = [0,...,0,1]^T,\ C = [1,0,...,0]^T \tag{III.14}$$

Définissons l'erreur de poursuite par

$$e(t) = y(t) - y_d(t) \tag{III.15}$$

et le vecteur erreur de poursuite de tous les états est définit par $e = X - Y_d$ avec

$Y_d = \left[y_d, \dot{y}_d, ..., y_d^{(n-1)} \right]^T \in R^n$.

La dynamique des erreurs de poursuite est donc donnée par:

$$\dot{e} = A.e + B\left[f(X) + g(X)u - y_d^{(n)} \right] \tag{III.16}$$

Selon le principe de linéarisation par bouclage, une nouvelle entrée de commande υ est introduite. Soit la loi de commande suivante [1] :

$$u = \frac{1}{g(X)}\left(-f(X) + \upsilon\right) \tag{III.17}$$

En remplaçant (III.17) dans (III.16), on obtient :

$$\dot{e} = A.e + B.\upsilon \tag{III.18}$$

ce qui représente, finalement, un système linéaire. La nouvelle entrée de commande υ peut alors être choisie en utilisant les méthodes de conception de contrôleurs pour les systèmes linéaires, par exemple par placement de pôles.

III.6.2. Commande par mode glissant

La commande par mode glissant est une technique de commande des systèmes à structure variable, largement utilisée dans la littérature. Son succès est dû à sa robustesse vis à vis des variations paramétriques et des perturbations externes. Elle consiste à amener la trajectoire d'état du système sur une surface dite de glissement convenablement choisie, puis à la faire glisser jusqu'au point d'équilibre [63][3].

La conception de cette commande est basée sur trois étapes dont la première consiste à choisir la surface de glissement qui représente la dynamique désirée. La deuxième établie la condition d'existence du mode de glissement (attractivité) et la dernière étape détermine la loi de commande qui aura pour rôle de garantir le maintien et le glissement le long des trajectoires du système sur cette surface.

Dans un souci de faciliter la compréhension, une classe des systèmes SISO affine en la commande est considérée dans cette section.

Soit le système non-linéaire SISO, affine en la commande, donné par (III.3) :

$$\begin{cases} \dot{X} = f(X) + g(X)u \\ y = h(X) \end{cases}$$

Supposition 1.1 : la fonction $g(X)$ est supposée non nulle ($g(X) \neq 0$).

III.6.2.1. Choix de la surface de glissement

La forme des surfaces de glissement est déterminée en fonction de l'application et des objectifs visés. Elle est choisie de sorte qu'elle assure la convergence de la grandeur de sortie vers sa valeur de référence [64].

On définie dans $\mathbb{R}$ la surface de glissement ou de commutation notée s [3] :

$$s(X,t) = \left(\frac{\partial}{\partial t} + \lambda\right)^{(n-1)} e, \qquad s : \mathbb{R}^n \to \mathbb{R} \tag{III.19}$$

avec λ une constante strictement positive et n est le degré relatif lié à la sortie y .

Le problème de poursuite revient à rapprocher les trajectoires des erreurs vers la surface de glissement $s(X) = 0$ puis, à maintenir ces trajectoires le long de cette surface.

Remarque III.1 [65]

Soit $s(e_0,...,e_{n-1}) = e_{n-1} + \lambda_{n-2}e_{n-2} + ... + \lambda_1 e_1 + \lambda_0$ où les coefficients λ_i sont choisis de telle manière que $e^{n-1} + \lambda_{n-2}e^{n-2} + ... + \lambda_1 e + \lambda_0$ soit un polynôme de Hurwitz. Alors $s(e, \dot{e}, ..., e^{n-1}) = 0$ est une équation linéaire ordinaire stable.

Remarque III.2

On dit qu'il existe un régime glissant idéal sur s s'il existe un temps fini t_s tel que la solution de (III.3) satisfasse $s(X,t) = 0$ pour $t \geq t_s$.

III.6.2.2. Condition d'existence du mode de glissement (attractivité)

L'étude de l'existence du mode de glissement basée sur la méthode de Lyapunov, consiste à définir une fonction scalaire $L(X)$ définie positive pour chaque sortie y et de construire une loi de commande afin que la dérivée de cette fonction décroisse et garantisse ainsi la stabilité du système, ce qui peut être traduit mathématiquement par [66]:

$$L(X) = s\dot{s} \leq 0 \tag{III.20}$$

Cette condition indique que les trajectoires des systèmes convergent asymptotiquement vers la surface de glissement $s(X,t)=0$ puis restent dans un ε-voisinage de cette surface comme le présente la figure III.5.

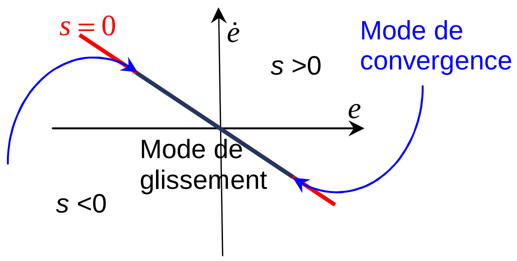


Fig. III.5. Attractivité de la surface

Remarque III.3

Pour une convergence en temps fini, la condition (III.20) qui ne garantie qu'une convergence asymptotique vers la surface de glissement est remplacée par une condition plus restrictive dite de η-attractivité et donnée dans [66]:

$$s\dot{s} \leq -\eta |s| \,,\ \eta > 0 \tag{III.21}$$

qui assure une convergence en temps fini $\left(t_s \leq \dfrac{|s(0)|}{\eta} \right)$ vers la surface $s(X,t)=0$.

III.6.2.3. Synthèse de la loi de commande par mode glissant

La loi de commande par mode glissant est construite de façon à ce que les trajectoires du système pointent vers la surface de glissement $s=0$ dans un premier lieu et ensuite maintenir ces trajectoires sur cette surface en deuxième lieu. Pratiquement, cette loi de commande est composée de deux composantes : une discontinue u_s permet de ramener les trajectoires vers la surface de glissement et d'assurer la robustesse vis à vis des perturbations externes et la seconde, continue, dite commande équivalente u_{eq} permet le maintien et le glissement le long de la surface.

Remarque III.4

La commande équivalente est déduite par les conditions d'invariance suivantes de la surface de glissement [67] :

$$\begin{cases} s = 0 \\ \dot{s} = \dfrac{\partial s}{\partial t} = \dfrac{\partial s}{\partial X}\dfrac{\partial X}{\partial t} = 0 \end{cases} \tag{III.22}$$

Pour le système (III.3), la loi de commande par mode glissant est choisie comme suit :

$$u = u_{eq} + u_{dis}$$

d'où

$$u_{eq} = -\left[\frac{\partial s}{\partial X} g(X)\right]^{-1} \frac{\partial s}{\partial X} f(X) \tag{III.23}$$

$$u_{dis} = -K\left[\frac{\partial s}{\partial X} g(X)\right]^{-1} \operatorname{sign}(s)$$

où K est une constante positive et sign est la fonction signe définie par :

$$\operatorname{sign}(s) = \begin{cases} 1 & \text{si } s > 0 \\ 0 & \text{si } s = 0 \\ -1 & \text{si } s < 0 \end{cases} \tag{III.24}$$

Si on considère la fonction candidate de Lyapunov suivante :

$$L = \frac{1}{2}s^2 \tag{III.25}$$

La dérivée par rapport au temps de cette fonction permet d'obtenir :

$$\dot{L} = s\dot{s} \leq -K|s|,\ K > 0 \tag{III.26}$$

et donc l'existence d'un régime glissant.

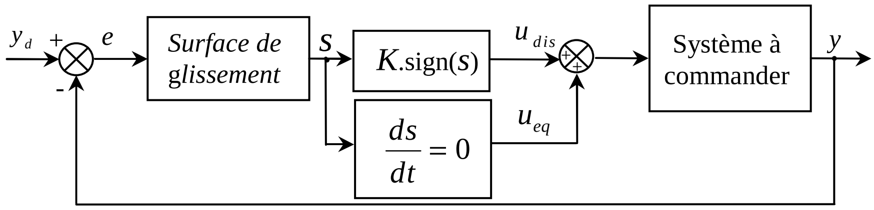


Fig. III.6. Diagramme schématique du principe de la commande par mode glissant

III.6.2.4. Phénomène de broutement

L'utilisation de la fonction signe dans la loi de commande pour assurer le passage de la phase d'approche à celle du glissement, excite de fortes oscillations autour de la surface de glissement (voir la figure III.7) qui entraînent l'apparition de ce qu'on appelle " broutement'' ou " réticence'' connu en anglais sous le nom de ''*chattering*'', qui consiste en des variations brusques et rapides du signal de commande, ce qui peut exciter les hautes fréquences du processus et l'endommager.

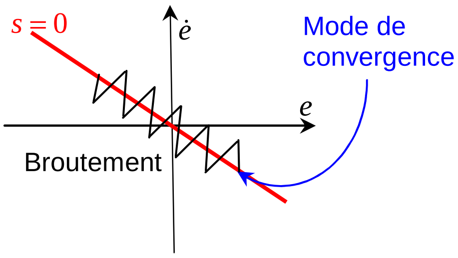


Fig. III.7. Mode de glissement avec broutement

Ce phénomène peut conduire à la dégradation des performances du système à commander, et peut même conduire à l'instabilité [67].

III.6.2.5. Quelques solutions pour le broutement

Afin de réduire ou d'éliminer le phénomène de chattering, de nombreuses techniques ont été proposées. Parmi les techniques les plus utilisées on cite :

a) ***Solution de la couche limite***

Dans ce cas la fonction signe est remplacée par des approximations continues [5][3] comme la fonction de saturation « sat » qui filtre les hautes fréquences. D'autres fonctions peuvent être aussi utilisées telles que les fonctions sigmoïdes, tangente hyperbolique « tanh », arc tangente « arctan », etc.

$$\operatorname{sat}(s/\phi)=\begin{cases} s/\phi & \text{si} \quad |s/\phi|\le 1 \\ \operatorname{sign}(s/\phi) & \text{si} \quad |s/\phi|>1 \end{cases} \tag{III.27}$$

où ϕ représente la bande de transition de la surface de glissement.

b) Solution basée sur un observateur

Il s'agit de placer un observateur asymptotique par mode glissant pour éliminer le broutement du à cette discontinuité de la loi de commande. Cette idée a été proposée par Bondarev *et al.* [6] et consiste à générer les modes glissants idéaux dans une boucle auxiliaire d'observation, telle que cette boucle d'observateurs n'intègre aucune dynamique non modélisée. La boucle principale est poursuivie de la boucle d'observateurs.

c) Mode glissant d'ordre supérieur

On remplace le mode glissant d'ordre un « $s=0$ » par le mode glissant d'ordre supérieur « $s=s^{(1)}=...=s^{(n)}=0$ » où n est l'ordre du mode glissant [7][8].

d) Régulateur PI type flou

Le terme discontinu est remplacé par un régulateur PI flou [32] comme suit:

$$u_s = K.\operatorname{sign}(s) = K_p s + K_i \int s(\tau)\,d\tau = \theta_p^T \xi(z) \tag{III.28}$$

où $\theta_p=\begin{bmatrix}K_p & K_i\end{bmatrix}^T$ est le vecteur des paramètres ajustables, $\xi(z)$ est le vecteur des fonctions floues de base avec $z=\begin{bmatrix}z_1 & z_2\end{bmatrix}^T$, telle que $z_1=s$ et $\dot{z}_2=s$.

III.6.3. Technique de commande par backstepping

La technique de commande par backstepping a été développée en 1991 par Kanella kopoulos *et al.* [68], elle est inspirée par les travaux de [69][70][71]. Cette technique offre une méthode systématique pour effectuer la conception d'un contrôleur pour les systèmes non-linéaires, l'idée consiste à calculer une loi de commande afin de garantir pour une certaine fonction (Lyapunov) définie positive que sa dérivée soit toujours négative. L'objectif de cette technique est de calculer, en plusieurs étapes, une commande qui garantit la stabilité globale du système

Dans un souci de faciliter la compréhension, on considère le cas des systèmes non-linéaires de la forme

$$\begin{cases} \dot{x}_1 = f_1 + g_1 x_2 \\ \dot{x}_2 = f_2 + g_2 x_3 \\ \vdots \\ \dot{x}_n = f_n + g_n u \end{cases} \tag{III.29}$$

où, f_i et g_i sont des fonctions non-linéaires connues tel que $f(0) = 0$ et $f_i(x) \neq 0$, On désire faire suivre à la sortie $y = x_1$ le signal de référence y_r. Le système étant d'ordre n, le design s'effectue en n étapes.

Étape 1 :

On considère d'abord le premier sous-système :

$$\dot{x}_1 = f_1(x_1) + g_1(x_1)\, x_2 \tag{III.30}$$

Définissons la première variable $z_1 = x_1$ et la deuxième variable $z_2 = x_2 - \alpha_1$, avec α_1 défini comme une loi de commande virtuelle.

La dérivée par rapport au temps de z_1 est exprimée par :

$$\dot{z}_1 = \dot{x}_1 = f_1 + g_1 z_2 + g_1 \alpha_1 \tag{III.31}$$

Pour trouver la loi de commande virtuelle α_1, nous construisons la fonction de Lyapunov suivante :

$$L_1(z_1) = \frac{1}{2} z_1^2 \tag{III.32}$$

Sa dérivée est exprimé par:

$$\dot{L}_1(z_1) = z_1 \dot{z}_1 = z_1 g_1 z_2 + z_1 \left(f_1 + g_1 \alpha_1 \right) \tag{III.33}$$

Un choix judicieux de α_1 rendrait $\dot{L}_1$ négative et assurerait la stabilité du sous système décrit par l'équation (III.30). Donc α_1 est choisi comme suit:

$$\alpha_1 = \frac{1}{g_1}\left(-f_1 - c_1 z_1\right) \tag{III.34}$$

où $c_1 > 0$ est un paramètre de design.

On substitue α_1 dans (III.33) et dans z_1, on trouve :

$$\dot{L}_1 = -c_1 z_1^2 + g_1 z_1 z_2 \tag{III.35}$$

$$\dot{z}_1 = -c_1 z_1 + g_1 z_2 \tag{III.36}$$

Pour une stabilité globale, le dernier terme $g_1 z_1 z_2$ dans $\dot{L}_1$ sera éliminé dans la prochaine étape.

Étape 2 :

On considère, dans ce cas, les deux premiers sous-systèmes :

$$\begin{cases} \dot{x}_1 = f_1 + g_1 x_2 \\ \dot{x}_2 = f_2 + g_2 x_3 \end{cases} \tag{III.37}$$

Dans cette étape nous utilisons une nouvelle variable $z_3 = x_3 - \alpha_2$

La dérivée par rapport au temps de z_2 est exprimée par :

$$\dot{z}_2 = f_2 + g_2 x_3 - \dot{\alpha}_1 \tag{III.38}$$

La fonction de Lyapunov pour le second sous système est choisie comme suit :

$$L_2(z_1, z_2) = L_1(z_1) + \frac{1}{2} z_2^2 \tag{III.39}$$

Sa dérivée par rapport au temps est exprimée par:

$$\begin{aligned} \dot{L}_2(z_1, z_2) &= \dot{L}_1 + z_2 \dot{z}_2 = \dot{L}_1 + z_2 \left(f_2 + g_2 x_3 - \dot{\alpha}_1 \right) \\ &= -c_1 z_1^2 + z_2 \left(f_2 + g_1 z_1 + g_2 x_3 - \dot{\alpha}_1 \right) \\ &= -c_1 z_1^2 + g_2 z_2 z_3 + z_2 \left(f_2 + g_1 z_1 + g_2 \alpha_2 - \dot{\alpha}_1 \right) \end{aligned} \tag{III.40}$$

Pour rendre $\dot{L}_2$ négative, la loi de commande virtuelle α_2 peut être choisie comme suit :

$$\alpha_2 = \frac{1}{g_2}\left(-f_2 - c_2 z_2 - g_1 z_1 - \dot{\alpha}_1 \right) \tag{III.41}$$

où $c_2 > 0$ est un paramètre de design.

On substitue α_2 dans (III.40) et dans z_2, on trouve :

$$\dot{L}_2\left(z_1, z_2\right) = -c_1 z_1^2 - c_2 z_2^2 + g_2 z_2 z_3 \tag{III.42}$$

$$\dot{z}_2 = -c_2 z_2 - g_1 z_1 + g_2 z_3 \tag{III.43}$$

Pour une stabilité globale, le dernier terme $g_2 z_2 z_3$ dans $\dot{L}_2$ sera éliminé dans la prochaine étape.

Étape n :

Le système (III.29) est maintenant considéré dans sa globalité. La nouvelle variable d'erreur est $z_n = x_n - \alpha_{n-1}$

La dérivée par rapport au temps de z_n est exprimée par :

$$\dot{z}_n = \dot{x}_n - \dot{\alpha}_{n-1} = f_n + g_n u - \dot{\alpha}_{n-1} \tag{III.44}$$

La fonction de Lyapunov globale est exprimée par :

$$L_n\left(z_1, z_2, ..., z_n\right) = L_{n-1}\left(z_1, z_2, ..., z_{n-1}\right) + \frac{1}{2} z_n^2 \tag{III.45}$$

Sa dérivée par rapport au temps est exprimée par :

$$\dot{L}_n = \dot{L}_{n-1} + z_n \dot{z}_n \tag{III.46}$$

Ce qui conduit à :

$$\dot{L}_n = \sum_{i=1}^{n-1}\left(-c_i z_i^2\right) + z_n\left(f_n + g_{n-1} z_{n-1} + g_n u - \dot{\alpha}_{n-1}\right) \tag{III.47}$$

Pour rendre la dérivée de la fonction de Lyapunov globale négative, la loi de commande u est choisie comme suit :

$$u = \frac{1}{g_n}\left(f_n + -c_n z_n - g_{n-1} z_{n-1} - \dot{\alpha}_{n-1}\right) \tag{III.48}$$

où $c_n > 0$ est un paramètre de design.

On substituant u dans (III.47) et dans z_n, on trouve :

$$\dot{L}_n = \sum_{i=1}^{n-1}\left(-c_i z_i^2\right) \leq 0 \tag{III.49}$$

$$\dot{z}_n = -c_n z_n - g_{n-1} z_{n-1} \tag{III.50}$$

d'où la stabilité globale du système original (III.29) en boucle fermée est alors assurée.

III.7. Observation de systèmes non-linéaires

L'élaboration d'une loi de commande nécessite souvent l'accès à la valeur d'un ou de plusieurs de ses états. Ceci n'est pas toujours possible à cause de l'inaccessibilité d'état et/ou le manque de capteurs. En plus les mesures des capteurs sont souvent entachées de bruit, ce qui limite les performances d'une boucle de commande. Pour des raisons physiques ou économiques, on a recours à un système dynamique auxiliaire appelé observateur qui est chargé d'estimer l'état du système à partir des entrées appliquées et des mesures fournies par des capteurs physiques.

Différentes techniques ont été proposées pour résoudre le problème de synthèse d'observateur [72][73]. L'idée générale derrière ces techniques est de transformer le système non-linéaire observable en un système plus simple, pour lequel la synthèse d'un observateur est possible. Ces techniques ont donné naissance à de nombreux algorithmes d'observation [74][75][76][77].

Un observateur est un système dynamique qui reconstruit l'état du système à partir des entrées et des sorties mesurées. Les entrées d'un observateur sont donc la commande et la sortie du système, et la sortie d'un observateur est l'état estimé. La figure III.8 donne un schéma qui décrit ce principe.

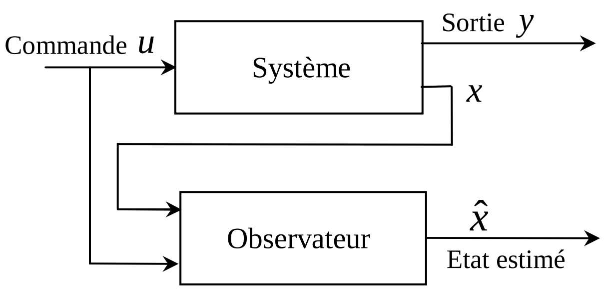


Fig. III.8. Ensemble système - observateur

Historiquement, les observateurs développés en premier lieu pour l'estimation d'état des systèmes dynamiques linéaires, les observateurs de Kalman [78][79] et de Luenberger [80] se sont rapidement révélés inadaptés pour observer l'état des systèmes non-linéaires.

De nombreuses méthodes ont ainsi été proposées depuis le début des années 1970 pour y remédier. Dans ce qui suit, nous présentons certaines des approches les plus étudiées dans la littérature ces dernières années, relatives à l'estimation d'état des systèmes non-linéaires.

III.7.1. Observateur à mode glissant

Cet observateur est basé sur les méthodes utilisées dans la commande par modes glissants [67]. Le principe des observateurs à mode glissant consiste à contraindre, à l'aide des fonctions discontinues, les dynamiques d'un système d'ordre n à converger vers une variété de surface S, sur laquelle l'erreur d'estimation de la sortie $e = y - \hat{y}$ tend vers zéro [75][81].

Considérons le système non-linéaire décrit par :

$$\begin{cases} \dot{x} = f(x,u) \\ y = h(x) \end{cases} \tag{III.51}$$

où $x \in \mathbb{R}^n$ représente le vecteur d'état. $u \in \mathbb{R}^m$ est le vecteur d'entrée ou la commande du système. $y \in \mathbb{R}^p$ dénote le vecteur de sortie. Les fonctions f et h sont des champs de vecteurs. Un observateur à modes glissants [82][83] s'écrit sous la forme :

$$\begin{cases} \dot{\hat{x}} = f(\hat{x},u) + \Gamma\, sign(y - \hat{y}) \\ \hat{y} = h(\hat{x}) \end{cases} \tag{III.52}$$

avec $\hat{x}$: l'état estimé de dimension $n \times 1$.

u : l'entrée ou commande de l'observateur de dimension $m \times 1$.

y et $\hat{y}$: sont respectivement, la sorties mesurée et estimée, de dimension $p \times 1$.

$sign(y - \hat{y}) = \left[sign(y_1 - \hat{y}_1) \quad sign(y_2 - \hat{y}_2) \quad \cdots \quad sign(y_p - \hat{y}_p) \right]$.

Γ : Matrice de gain de l'observateur, de dimension $n \times p$.

$f(.)$: Fonction non-linéaire d'évolution d'états, de dimension $n \times 1$.

$h(.)$: Fonction non-linéaire de sortie, de dimension $p \times 1$.

Les propriétés intéressantes dans ce type d'observateur sont celles liées à la convergence en temps fini vers la surface de glissement et à la réduction de la dynamique totale de n à $n - p$ états sur la surface de glissement.

III.7.2. Filtre de Kalman Étendu

Le filtre de Kalman développé à l'origine par Rudolf Emil Kalman en 1960 permet de résoudre le problème d’observation pour les systèmes linéaires. Le filtre de Kalman étendu ou observateur de Kalman étendu est une extension directe du filtre de Kalman standard. Il est l’une des techniques d’estimation les plus populaires et largement étudiées dans le domaine d’estimation d’état des systèmes dynamiques non-linéaires. Cet observateur consiste à utiliser les équations du filtre de Kalman standard sur le modèle non-linéaire linéarisé par la formule de Taylor au premier ordre. Malgré que les preuves de stabilité et de convergence établies dans le cas des systèmes linéaires ne peuvent pas être étendues de manière générale au cas des systèmes non-linéaires, cette méthode reste la plus populaire et largement étudiée dans le domaine d’observation des systèmes non-linéaires [84][85].
On considère le système non-linéaire suivant :

$$\begin{cases} x_{k+1} = f(x_k, u_k, w_k) \\ y_k = h(x_k,\ v_k) \end{cases} \tag{III.53}$$

où $x_k \in \mathbb{R}^n$ est le vecteur d’état, u_k est le vecteur d'entrée de commande, et y_k est le vecteur de mesure. $f(.)$ est la fonction d'évolution du système, alors que $h(.)$ représente la relation entre le vecteur d'état et le résultat de mesure. w_k et v_k sont respectivement, les bruits d’état et de mesure, supposés blancs et gaussiens avec des moyennes nulles, de covariances respectives :

$$Q = E[w_k, w_k^T] \tag{III.54}$$

$$R = E[v_k, v_k^T] \tag{III.55}$$

Pour appliquer EKF à la non-linéarité donnée par (III.53), il faut la linéariser en utilisant l'approximation de Taylor du premier ordre au voisinage du point de référence désiré $(\hat{x}_k, \hat{w}_k = 0, \hat{v}_k = 0)$ qui donne le modèle linéaire approximatif suivant:

$$\begin{cases} x_{k+1} = f(x_k, u_k, w_k) \approx f(\hat{x}_k, u_k, 0) + F_k(x_k - \hat{x}_k) + W_k(w_k - 0) \\ y_k = h(x_k,\ v_k) \approx h(\hat{x}_k,\ 0) + H_k(x_k - \hat{x}_k) + V_k(v_k - 0) \end{cases} \tag{III.56}$$

où, F_k, W_k, H_k et V_k sont les matrices jacobiennes données par:

$$F_k = \left.\frac{\partial f(x,0)}{\partial x}\right|_{x=\hat{x}_k},\ W_k = \left.\frac{\partial f(\hat{x}_k, w)}{\partial w}\right|_{w=0},\ H_k = \left.\frac{\partial h(x,0)}{\partial x}\right|_{x=\hat{x}_k},\ V_k = \left.\frac{\partial h(\hat{x}_k, v)}{\partial v}\right|_{v=0} \tag{III.57}$$

Avec cette définition on peut aussi résumer le filtre de Kalman étendu par deux étapes suivantes: la prédiction et la correction [86][87].

1ère étape: Equations de prédiction : On effectue une prédiction $\hat{x}_{k+1/k}$ d'état du processus à partir d'un modèle d'état conçu à partir des équations du système. Cette étape est généralement assez précise.

$$\begin{cases} \hat{x}_{k+1/k} = f(x_{k/k}, u_k, 0) \\ P_{k+1/k} = F_k P_{k/k} F_k^T + W_k Q W_k^T \end{cases} \tag{III.58}$$

2ème étape: Equations de correction : On effectue la correction de la valeur des variables d'état issue d'étape précédente pour obtenir $\hat{x}_{k+1/k+1}$ à partir des mesures effectuées sur le processus réel en corrigeant ainsi les résultats de la partie prédictive.

$$\begin{cases} K_k = P_{k+1/k} H_k^T (H_k P_{k+1/k} H_k^T + V_k R V_k^T)^{-1} \\ \hat{x}_{k+1/k+1} = \hat{x}_{k+1/k} + K_k (y_k - h(\hat{x}_{k+1/k}, 0)) \\ P_{k+1/k+1} = P_{k+1/k} - K_k H_k P_{k+1/k} \end{cases} \tag{III.59}$$

où la notation $\hat{x}_{k+1/k}$ représente le vecteur de prédiction d'état à priori, $\hat{x}_{k+1/k+1}$ est le vecteur de prédiction d'état postérieur, $P_{k+1/k}$ désigne la matrice de covariance d'erreur de prédiction à priori, $P_{k+1/k+1}$ est la matrice de covariance d'erreur de prédiction à posteriori, K est la matrice de gain. La représentation fonctionnelle de l'algorithme EKF est schématisée sur la figure III.9.

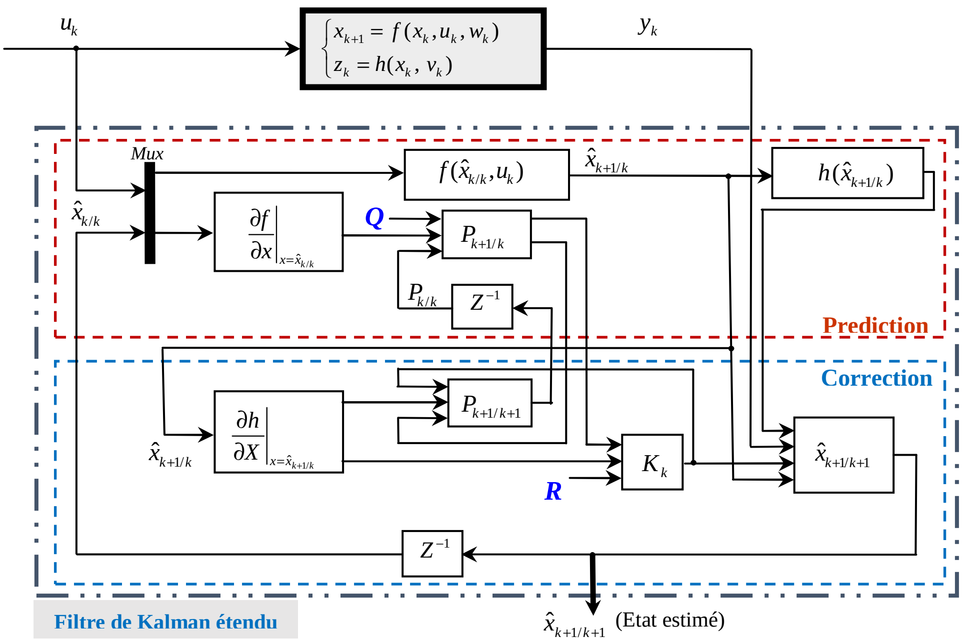


Fig. III.9. Schéma fonctionnel du filtre Kalman étendu [87]

III.8. Conclusion

Ce chapitre a été consacré à quelques rappels sur les propriétés générales des systèmes non-linéaires avec une mise au point de quelques concepts relatifs à la stabilité des systèmes non-linéaires (stabilité au sens de Lyapunov). Nous avons exposé un rappel sur la modélisation et la représentation entrée-sortie des systèmes non-linéaires. Nous avons aussi présenté les différentes méthodes, les plus utilisées pour la commande des systèmes non-linéaires telle que la commande linéarisante, la commande par mode glissant et la commande backstepping. D'autre part, nous avons présenté deux approches très étudiées dans la littérature qui concernent l'estimation d'état des systèmes non-linéaires, telles que l'observateur à mode glissant et le filtre de Kalman étendu.

Chapitre IV

Commande par backstepping-mode glissant floue type-1 : Application au quadrirotor

Sommaire

Commande par backstepping-mode glissant floue type-1 : Application au quadrirotor

Le plus difficile pour celui qui commande n'est pas de concevoir
de grands desseins, mais de les faire exécuter.

Édouard Alletz; Maximes politiques (1840).

IV.1. Introduction

Ces dernières années, les véhicules aériens sans pilote sont devenus un vaste sujet d'intérêt dans de nombreux organismes de recherche, en raison de leurs vastes applications dans plusieurs domaines. Ces véhicules aériens sont connus sous le nom de drones ou UAV. Parmi lesquels nous pouvons citer les mini drones de quelque kilogrammes, les micros drones de quelques dizaine de grammes et les nano drones de quelques grammes [88][89][90][91][92]. On s'intéressera ici plus particulièrement aux mini drones, type quadrirotor. Ce type de véhicules aériens a suscité un grand intérêt des chercheurs grâce à son versatilité, manœuvrabilité, capacité d’exécuter des décollages et atterrissages verticaux, conçu pour réaliser des missions de façon plus sûre et plus efficace qu'un véhicule piloté, etc., telle que, la surveillance de trafic routier, la surveillance et la protection de l’environnement, la recherche et le secours de blessés ou la gestion de grandes infrastructures telles que les lignes haute tension, les barrages et les ponts. Le développement de telles missions de façon plus sûre et plus efficace dépend de la capacité à contrôler le voyage de ces véhicules dans un environnement perturbé. Dans ce contexte, un contrôleur robuste et intelligent doit être conçu.
A travers ce chapitre, après une brève présentation du contexte historique et opérationnel dans lequel les mini drones sont utilisés, nous exposerons l'étude bibliographique qui nous a permis de présenter un état de l'art sur les différentes stratégies de commande mises en œuvre pour remplir les différentes missions du quadrirotor. Ensuite, nous donnerons une formulation du problème. Puis, nous synthétiserons une loi de commande robuste pour la commande de l'altitude et de l’attitude d'un quadrirotor en présence de perturbations externe élevées. Il s’agit

de combiner la logique floue type-1 avec la commande backstepping-mode-glissant, de manière adaptative selon certaines règles floues appropriées, afin d'adapter le gain de commutation pour réduire le phénomène de broutement, pour concevoir une structure de commande stable et robuste. Nous exposerons d'une manière constructive la procédure de développement d’un contrôleur non-linéaire permettant d'atteindre les objectifs désirés. Enfin, la performance de la commande développée à été mise en évidence à travers son application à un quadrirotor.

IV.1.1. Brève histoire des quadrirotors

Le Quadrirotor est un véhicule aérien sans pilote propulsé par quatre rotors. L'histoire du quadrirotor remonte au début du vingtième siècle avec diverses tentatives comme le gyroplane de Breguet-Richet (1907), l'Oehmichen (1922) et le *Flying Octopus* de De Bothezat (1924) [93]. Ces premiers prototypes présentaient de sérieuses restrictions en termes de contrôle et surtout d'endurance. Le *Flying Octopus* susmentionné pourrait rester en l'air pendant seulement 2 minutes et 45 secondes. Récemment, Les quadrirotors présentent plusieurs avantages par rapport aux hélicoptères de dimensions similaires. Premièrement, les quadrirotors standard sont mécaniquement moins complexes que les hélicoptères en raison de l'absence du moyeu de rotor sophistiqué. Ceci simplifie grandement la conception et la maintenance du quadrirotor. Deuxièmement, l'utilisation de quatre rotors de petit diamètre (généralement) permet aux quadrirotors de posséder moins d'énergie cinétique en vol que les hélicoptères. Récemment, les conceptions de quadrirotors sont devenues populaires pour de nombreuses applications d'UAV, Enfin, ils sont petits, peu coûteux. Ces avantages ont motivé ces dernières années le développement de quadrirotors pour de nombreux travaux coopératifs d'UAV [94][95].

IV.1.2. Utilisations dans le domaine civil

On recense de manière non exhaustive dans le paragraphe suivant l'usage de mini drones dans divers contextes civils.

Agriculture

Les utilisations agricoles des mini drones sont très diverses [94][96]. Dotés de capteurs optiques ou hyperspectraux ces engins permettent de collecter un grand nombre de données dont le traitement va ensuite faciliter l'analyse d’évolution des cultures, le repérage des mauvaises herbes ou des nuisibles. Ainsi, un suivi régulier peut être garanti pour un coût modeste. En outre, la mise en œuvre des drones est ici facilitée par l'absence de population sur les zones survolées.

Sécurité Civile

Les drones offrent un précieux secours dans la réponse immédiate à une catastrophe naturelle ou à un accident. Encore une fois, ils permettent le déploiement rapide de capteurs pour obtenir des informations sur la zone touchée. En plus de ces avantages, ils offrent des points de vue différents dans la collecte d'images par rapport à ceux d'un satellite. Cela facilite le post-traitement des images [97]. Des robots de toutes tailles ont été utilisés dans les récents événements [98]. Des quadrirotors ont permis de suivre l'évolution d'une marée noire en Alaska. Les ouragans Katrina et Wilma furent également des lieux d'expérimentations sur les modes opératoires des véhicules [91]. En mai 2013, un automobiliste canadien a pu être secouru grâce au déploiement rapide d'un drone équipé d'une caméra infrarouge. Celle-ci a permis de repérer le blessé égaré dans la nuit alors qu'une équipe au sol et un hélicoptère avaient auparavant échoué.

Inspection

L'inspection de bâtiments, de ponts ou d'éléments d'une architecture industrielle ainsi que des oléoducs ou des lignes à haute tension sont autant de tâches à effectuer régulièrement pour détecter le plus rapidement possible une dégradation ou défaillance potentielle [99][100].

Recherche scientifique

Les drones sont des plateformes idéales pour la collecte de données scientifiques [92][101]. L'archéologie bénéficie également de cette technologie pour le relevé de sites [102]. La surveillance de populations animales et d’écosystème d'une zone ont également été envisagés [103].

Média

Une société professionnelle de journalistes utilisant des drones [104] pour tenter à la fois de proposer un cadre éthique à cette pratique et la promouvoir. Les drones sont appréciés pour le tournage de séquences vidéo (films, publicités ou événements sportifs). Ils permettent de réaliser des plans avec un minimum de matériel.

Humanitaire

La documentation de situations de guerre est encore un exemple parfait de l'usage de drones dans des conditions dangereuses pour l'homme [105]. L'entreprise Matternet imagine distribuer des médicaments rapidement à des zones sinistrées ou difficiles d'accès à l'aide d'un réseau de drones [106]. En fin on peut encore citer l'initiative *Conservation Drones* [95] qui vise à développer des prototypes à bas coût, simple d'accès pour la surveillance de forêts, l'évaluation de leur biodiversité et la détection d'activités illégales.

IV.1.3. Configuration

Une configuration de mini-drone en particulier a connu un succès grandissant depuis plus de 10 ans. Il s'agit d'un drone possédant 4 hélices, d'où son nom de "quadrirotor" ("*quadrotor*", en anglais). Ce type de mini-drone est étudié dans de nombreuses universités ou écoles d'ingénieurs (figure IV.1) et est même vendu au grand public comme un jeu pour l'extérieur ou l'intérieur (un exemple est donné sur la figure IV.2). On considère dans la suite ce type de mini-drone qui a comme avantage, parmi d'autres, de pouvoir être modélisé et contrôlé automatiquement de manière relativement simple.

Fig. IV.1. Prototype étudié dans ce sujet

Fig. IV.2. AR.Drone de Parrot

Les sections précédentes ont permis d'établir le contexte historique et opérationnel dans lequel les mini drones sont utilisés. On s'attache à présent à comprendre les stratégies de commande mises en œuvre pour remplir ses missions. L'absence de pilote et les petites dimensions rendent possibles des manœuvres qu'un engin habité ne pourrait réaliser de manière récurrente. L'usage de la propulsion électrique et d'actionneur rapide rendent les prototypes agiles. Cependant, l'usage de ces capacités dépend de la capacité à maitriser la dynamique de tels systèmes suffisamment précisément et de la capacité à synthétiser des lois de commandes rapides et précises.

IV.1.4. Etat de l’art de la commande de drones

En général, la position dans l’espace de ce type de véhicule est commandée par un operateur (radio-pilotage) à travers d’un système de commande à distance, tandis que l'attitude peut être automatiquement stabilisée via un contrôleur embarqué. La commande d’attitude est une étape essentielle pour développer une commande complète pour le quadrirotor. Cette commande est considérée comme une application directe d’un problème plus général de la commande d’attitude d’un corps rigide. Ce problème de la commande d’attitude d’un corps rigide a attiré un intérêt considérable depuis les années 50 dans les communautés scientifiques de l’aéronautique, de l’automatique et la robotique.

La commande d’altitude et de l'attitude sont les principaux intérêts à l'étude des quadrirotors. Ce problème a attiré beaucoup d'attention des chercheurs en raison de ses importances dans les diverses applications [89][107][108][109][110]. Pour résoudre ce problème, différentes algorithmes de commande ont été proposées dans la littérature. Dans le cas de système linéaire, l’utilisation de la commande PD et PID dans [89][111] et, la commande LQ et LQR dans [112][113]. Tandis que dans le cas de système non-linéaire, la commande PID est appliquée dans [114] pour stabiliser l'altitude et l'attitude d'un quadrirotor en présence de conditions perturbatrices. Dans [115], la commande par linéarisation exacte est utilisée pour un quadrirotor. La commande inverse-direct basée sur les réseaux de neurones est présentée dans [116]. D’autres algorithmes de commande sont faits avec les techniques de logique flou [117][118] et réseaux de neurones [119][120].

La technique de backstepping est une méthode de commande récursive basée sur la fonction de Lyapunov [121][2]. Plusieurs algorithmes sur ce sujet peuvent être trouvés dans la littérature, à savoir la commande backstepping dans [88][122a][122b][123] pour la stabilisation et/ou la poursuite d’une trajectoire d'un UAV.

La commande par mode glissant est bien connue pour son efficacité à travers différentes études théorique et réalisations pratiques d'ingénierie, et une technique largement utilisée dans divers domaines tels que le contrôle des robots [124][125]. Plusieurs auteurs ont réussi à développer un contrôleur par mode glissant pour le quadrirotor [88][126][127][128][129]. L’inconvénient majeur de cette approche réside dans le problème de broutement (fréquence élevée de l'action de commande) qui peut dégrader la performance de la commande et peut même conduire à l'instabilité du quadrirotor. Cela a conduit au développement de diverses techniques afin de réduire les effets du phénomène de broutement, telles que la couche limite, le réseau de neurones et la logique floue [5][130][131], le mode de glissement d'ordre deux [132] et, le intégral backstepping mode glissant [133].

La contribution de ce chapitre peut être résumée comme suite: Un contrôleur d'altitude-attitude robuste a été développé pour la stabilisation d'un quadrirotor. Le contrôleur proposé combine l'avantage de SMC avec la synthèse backstepping pour construire backstepping-mode-glissant (BSMC), qui a été amélioré par un système flou type-1 pour développer backstepping-mode-glissant flou (FBSMC). L'objectif principal est l'utilisation de la logique floue type-1 afin d'adapter le gain de commutation pour réduire le phénomène de broutement induit par la commande discontinue dans la commande BSMC, et d'obtenir une bonne réponse dynamique en présence de perturbations élevées. Le système flou type-1 d'une manière générale comprend trois composantes principales : la fuzzification, le mécanisme la d'inférence avec la base de règles floues et la defuzzification. L'analyse de stabilité du système en boucle fermée a été étudiée en utilisant la seconde méthode de Lyapunov basée sur la technique de backstepping. La mise en ouvre de la commande proposée était sous l'environnement MatLab. Les résultats de la simulation ont montré la robustesse et l'efficacité de la commande proposée; et a présenté une performance plus supérieure que son homologue conventionnel.

IV.2. Contexte et formulation

Concéderons un système non-linéaire incertain, dont la dynamique est décrite par :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(X) + g(X)u + d(t) \end{cases} \tag{IV.1}$$

où $X = [x_1,\ x_2]^T$ est le vecteur d'état, u est l'entrée de commande du système, $f(X)$ et $g(X)$ sont des fonctions non-linéaires continues et bornées, $d(t)$ représente les perturbations externes.

L'objectif est la synthèse d'une loi de commande robuste u, de telle sorte que le vecteur d'état X suit le plus près possible, une trajectoire de référence continue et bornée prédéfinie en présence des perturbations externes élevées. Ainsi, l'objectif est de déterminer la commande u qui permet de conduire l'erreur de poursuite $e(t)$ à converger vers zéro, tout en garantissant la stabilité du système en boucle fermée.

Pour ce faire, les hypothèses suivantes sont retenues dans ce chapitre.

Hypothèses : Afin de développer notre contrôleur, pour le système (IV.1), nous adoptons les hypothèses de simplification et de réalisation suivantes :

- **Hyp IV.1:** le vecteur d'état du système est disponible à la mesure.
- **Hyp IV.2:** $f(X)$ et $g(X)$ sont des fonctions non-linéaires continues et bornées, $g(X) \neq 0$
- **Hyp IV.3:** La perturbation d est supposée connue et limitée, *i.e.* $\|d\| \leq D$ où $D > 0$.
- **Hyp IV.4:** La trajectoire de référence X_d et ses dérivées; première et seconde sont disponibles et avec des limites connues.

Concernant le problème du contrôle non-linéaire, nous proposons d'utiliser l'approche mode glissant [3].

La loi de commande par mode glissant peut être synthétisée facilement par :

Étape 1 Définir la surface de glissement

Nous définissons l'erreur de poursuit entre lacet réel et lacet désiré suivant :

$$e = x - r \tag{IV.2}$$

où r est la trajectoire de référence. Par conséquent, la dérive de (IV.2) est donnée par :

$$\dot{e} = \dot{x}_1 - \dot{r} = x_2 - \dot{r} \tag{IV.3}$$

D'après Slotine et Li [3], la surface de glissement sera donnée par :

$$s = \frac{\partial e}{\partial t} + \lambda e = \dot{e} + \lambda e \tag{IV.4}$$

où $\lambda > 0$ est un nombre réel positif.

Étape 2 Synthèse de la loi de commande par mode glissant

Il suffit maintenant de choisir la loi de commande pour que la condition de convergence (III.20) (mentionné dans le chap. III.) soit vérifiée. Pour cela, il faudra exprimer s en fonction de la commande u.

À partir de (IV.4), on a

$$\begin{aligned} \dot{s} &= \ddot{e} + \lambda\dot{e} \\ &= \ddot{x} - \ddot{r} + \lambda\dot{e} = \dot{x}_2 - \ddot{r} + \lambda\dot{e} \\ &= f(X) + g(X)u + d + \lambda\dot{e} - \ddot{r} \end{aligned} \tag{IV.5}$$

Afin de vérifier la condition de convergence (III.20), la loi de commande doit être choisie comme

$$u = \frac{1}{g(X)}\left(-K\,\mathrm{sign}(s) - f(X) - d - \lambda\dot{e} + \ddot{r}\right) \tag{IV.6}$$

où $K \geq 0$, $K > D$ sont des nombres réels positifs.

Par substitution de (IV.6) dans (IV.5), on peut avoir:

$$\dot{s} = -K\ \mathrm{sign}(s)$$

donc,

$$s\dot{s} = -s\ K\ \mathrm{sign}(s) = -K\left|s\right| < 0$$

ce qui donne, e tend vers zéro et $\dot{e}$ tend vers zéro quand t tend vers l'infini. Par conséquent, la stabilité du sous-système en boucle fermée le long de la surface de glissement $s = 0$ est garantie.

La commande par mode glissant est bien connue pour sa robustesse vis-à-vis des incertitudes et des perturbations. Malheureusement, la mise en œuvre de cette commande souffre d'un problème de broutement qui peut dégrader la performance de la commande et peut même conduire à l'instabilité du système à commander, ce qui rend difficile leur implémentation pratique. L'utilisation de techniques basées sur l'expertise humaine peut être une alternative à la résolution de ce problème. Dans ce contexte, et motivé par les travaux de Bouabdallah *et al.* [88] et Bouadi *et al.* [89], ce chapitre est consacré à la combinaison de la logique floue type-1 avec la commande backstepping-mode-glissant, de manière adaptative selon certaines des règles floues appropriées, afin d'adapter le gain de commutation pour réduire le phénomène de broutement, pour concevoir une structure de commande stable et robuste. L'analyse de stabilité en boucle fermée est étudiée en utilisant la seconde méthode de Lyapunov basée sur la technique de backstepping.

IV.3. Structure générale de la commande adoptée

La méthode de commande proposée dans ce chapitre est basée sur la synthèse de loi de commande robuste pour une classe de systèmes non-linéaires (IV.1) permettant de faire converger les états du système vers leurs valeurs du références en présence des perturbations externes élevées. Telle que schématisée par la figure IV.3.

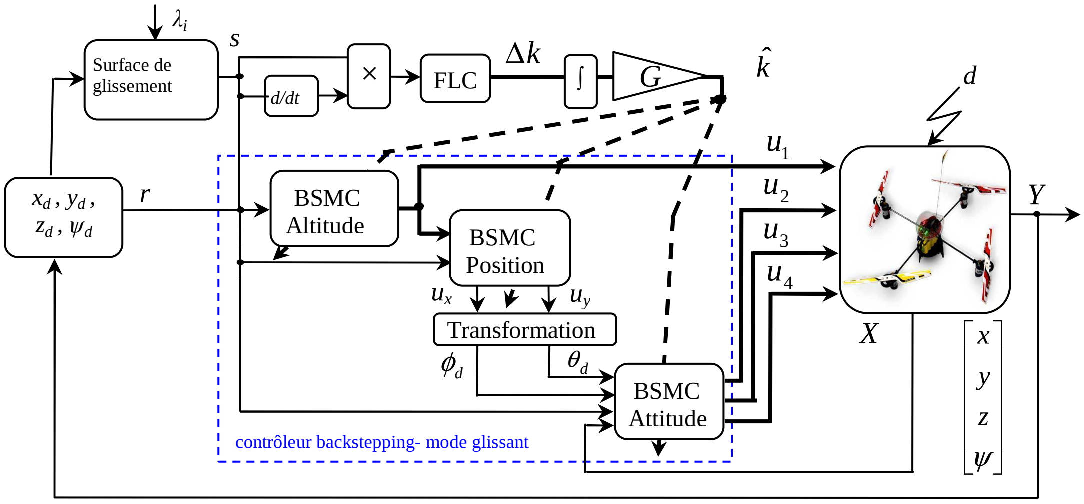


Fig. IV.3. Diagramme schématique de la commande proposée

La loi de commande par backstepping-mode-glissant est donnée par [134]:

$$u = \frac{1}{g(X)}\left(-K\,\mathrm{sign}(s) - f(X) - c_2 s - e_1 - c_1\dot{e}_1 + d + \ddot{r}\right) \tag{IV.7}$$

où $c_2 \geq 0$, $K > D$ sont des nombres réels positifs.

L'augmentation du gain de commutation élevé K de (IV.7) provoque une augmentation des oscillations dans le signal d'entrée de commande autour de la surface de glissement, par conséquent, un phénomène de broutement sera créé. De plus, un faible gain de commutation peut réduire le phénomène du broutement et améliorer la performance de poursuite malgré les incertitudes et les perturbations externes. Pour atteindre une performance plus appropriée, ce gain doit être ajusté. Ce réglage est basé sur la distance entre les états du système et la surface de glissement, c'est-à-dire que le gain devrait être élevé lorsque la trajectoire d'état est éloignée de la surface de glissement, et lorsque la distance diminue, il devrait être réduit. Cette idée peut être réalisée en combinant la logique floue type-1 avec la commande discontinue pour ajuster le gain K de manière adaptative selon certaines règles floues appropriées. Sous cette condition, le gain K est adapté par un gain flou Δk comme le montre la figure IV.3. L'architecture du FLS est représentée sur la figure I.3.

Le FLS se compose de quatre parties: le fuzzificateur, la base de connaissances, le moteur d'inférence flou, et le defuzzificateur. L'entrée du FLS est le produit de la surface glissante et sa dérivée $s\dot{s}$ et la sortie est Δk, les fonctions d'appartenance des variables linguistiques d'entrée et de sortie sont montrées dans la figure IV.4.

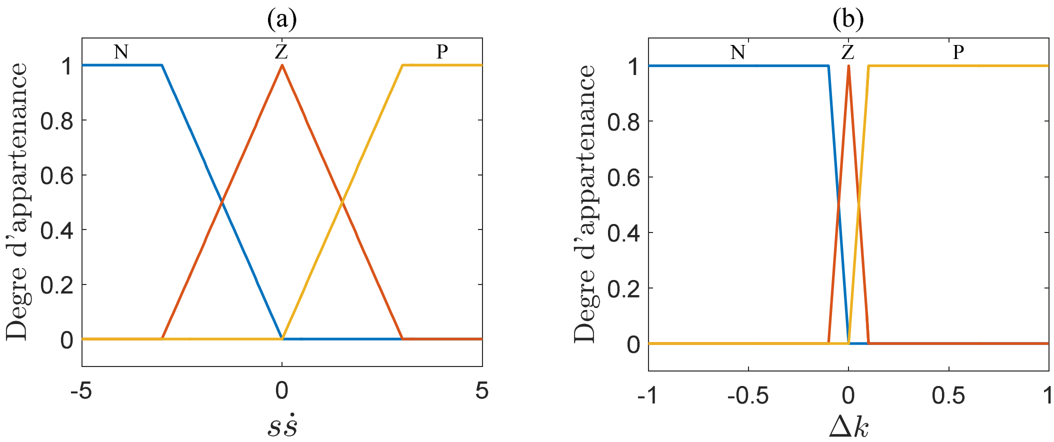


Fig. IV.4. (a) Fonctions d'appartenance d'entrée (b) Fonctions d'appartenance de sortie

La base de connaissance est composée d'une collection de règles ***SI … ALORS*** floues. Ces règles peuvent être énoncées de manière linguistique comme suit:

$$\text{Règle } i : \textbf{\textit{SI}} \;\; s \;\; \text{est } A^i, \textbf{\textit{ALORS}} \;\; \Delta k \;\; \text{est } B^i \;, \; i = 1, 2, ..., N$$

où N est le nombre total de règles. A^i représente un ensemble flou d'une variable d'entrée et qui peut être : **négatif (N)**, **zéro (Z)**, **positif (P)**, B^i est la sortie singleton de FLS, comprend les mêmes variables linguistiques.

La base de règles du FLS peut être exprimée comme dans le tableau IV.1

Tab. IV.1. Ensemble de règles floues

$s\dot{s}$	N	Z	P
Δk	N	Z	P

Notez que les fuzzifications singleton, la défuzzification avec la méthode centre de gravité, l'implication de Mamdani et le moteur d'inférence produit sont utilisés dans cette section. Par conséquent, la sortie du système flou type-1 peut être décrite par l'équation suivante

$$\Delta k = \frac{\sum_{i=1}^{N} \Delta k_i . \mu_{A_i^l}\left((s\dot{s})_i \right)}{\sum_{i=1}^{N} \mu_{A_i^l}\left((s\dot{s})_i \right)} \tag{IV.8}$$

où la fonction d'appartenance de l'ensemble de sortie est discrétisée sur N points. $\mu_{A_i^l}\left((s\dot{s})_i \right)$ est le degré d'appartenance résultant. Δk_i est la variable de sortie.

En utilisant la méthode intégrale, la limite supérieure de $\hat{k}_i(t)$ est adaptée par:

$$\hat{k}_i(t) = G \int_0^t \Delta k_i \, dt \tag{IV.9}$$

où G est le coefficient de proportionnalité et est ajusté selon les expériences d'essai-erreur.

IV.4. Synthèse de la loi de commande

IV.4.1. Synthèse de la commande backstepping-mode-glissant

Étape 1 : En tenant compte des équations de l'erreur et sa dérivée (IV.2) et (IV.3), respectivement, pour assurer la convergence de l'erreur de poursuite e vers 0, nous choisissons la première fonction de Lyapunov suivante :

$$L_1 = \frac{1}{2}e^2 \tag{IV.10}$$

dont la dérivée est donnée par :

$$\dot{L}_1 = e\dot{e} = e(x_2 - \dot{r}) \tag{IV.11}$$

Afin de réaliser $\dot{L}_1$ définie négative $(\dot{L}_1 \le 0)$, en utilisant la technique de backstepping (section III.6.3), nous considérons le système virtuel suivant :

$$x_2 = s - c_1 e + \dot{r} \tag{IV.12}$$

où s est la surface glissante, $c_1 > 0$ étant un paramètre de conception.
Par conséquent, l'équation (IV.12) nous donne

$$s = x_2 + c_1 e - \dot{r} = c_1 e - \dot{e}\, , c_1 > 0 \tag{IV.13}$$

En tenant compte des équations (IV.11) et (IV.12), on trouve l'expression de $\dot{L}_1$ suivante:

$$\dot{L}_1 = e.s - c_1 e^2$$

Si $s = 0$ alors $\dot{L}_1 \le 0$. Par conséquent, la prochaine étape est requise.
Étape 2 : La deuxième fonction de Lyapunov est donnée par :

$$L_2 = L_1 + \frac{1}{2}s^2 \tag{IV.14}$$

sa dérivée est donnée par :

$$\dot{L}_2 = \dot{L}_1 + s\dot{s} \tag{IV.15}$$

À partir de (IV.13), on a

$$\dot{s} = \dot{x}_2 + c_1\dot{e}_1 - \ddot{r}$$
$$= f(X) + g(X)u + d + c_1\dot{e}_1 - \ddot{r}$$

L'équation (IV.15) devient :

$$\dot{L}_2 = \dot{L}_1 + s\dot{s} = e.s - c_1e^2 + s\left(f\left(X\right) + g\left(X\right)u + d + c_1\dot{e}_1 - \ddot{r}\right)$$

Afin de réaliser $\dot{L}_2$ défini négative, la loi de commande backstepping-mode-glissant est choisi par:

$$u = \frac{1}{g\left(X\right)}\left(-\hat{k}\,\mathrm{sign}(s) - f\left(X\right) - c_2s - e - c_1\dot{e} + d + \ddot{r}\right) \tag{IV.16}$$

où $c_2 > 0$ étant un paramètre de conception.

Par conséquent, (IV.15) devient :

$$\dot{L}_2 = -\left(c_1e^2 + c_2s^2 + s.d + k_1\left|s\right|\right) \le 0 \tag{IV.17}$$

ce qui donne, e tend vers zéro et $\dot{e}$ tend vers zéro quand t tend vers l'infini. Par conséquent, la stabilité du sous-système en boucle fermée le long de la surface de glissement $s = 0$ est garantie.

IV.4.2. Application au quadrirotor

Dans le but de valider l'approche proposée, le modèle d'un quadrirotor est considéré. Ce type de quadrirotor a été étudié par plusieurs chercheurs (par exemple Castillo *et al.* [135], Derafa *et al.* [136], Derafa *et al.* [137] et Khebbache [138]) où la configuration est illustrée sur la figure IV.5.

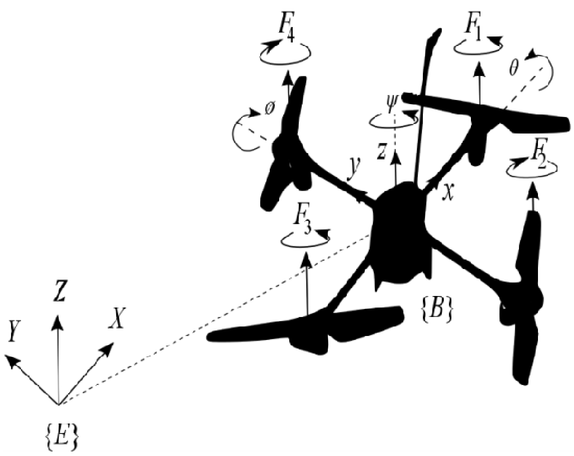


Fig. IV.5. Configuration du quadrirotor

En utilisant le formalisme de Newton-Euler, le quadrirotor peut être représenté par un système d'équations différentielles, données par :

$$\begin{cases} \ddot{\phi} = \dfrac{1}{I_x}\left\{ \dot{\theta}\dot{\psi}\left(I_y - I_z\right) - K_{fax}\dot{\phi}^2 - J_r\overline{\Omega}\dot{\theta} + l\,u_2 + d \right\} \\ \ddot{\theta} = \dfrac{1}{I_y}\left\{ \dot{\phi}\dot{\psi}\left(I_z - I_x\right) - K_{fay}\dot{\theta}^2 + J_r\overline{\Omega}\dot{\phi} + l\,u_3 + d \right\} \\ \ddot{\psi} = \dfrac{1}{I_z}\left\{ \dot{\theta}\dot{\phi}\left(I_x - I_y\right) - K_{faz}\dot{\psi}^2 + u_4 + d \right\} \\ \ddot{x} = \dfrac{1}{m}\left\{ (\cos\phi.\sin\theta.\cos\psi + \sin\phi.\sin\psi)u_1 - K_{fdx}\dot{x} + d \right\} \\ \ddot{y} = \dfrac{1}{m}\left\{ (\cos\phi.\sin\theta.\sin\psi - \sin\phi.\cos\psi)u_1 - K_{fdy}\dot{y} + d \right\} \\ \ddot{z} = \dfrac{1}{m}\left\{ \cos\phi\cos\theta.u_1 - K_{fdz}\dot{z} \right\} - g_a + d \end{cases} \tag{IV.18}$$

où m est la masse totale du quadrirotor. $J = diag\left(I_x, I_y, I_z\right) \in \mathbb{R}^{3\times3}$ est la matrice d'inertie. l est la distance entre le centre de masse du quadrirotor et l'axe de rotation du rotor. $K_{fa} = diag\left(K_{fax}, K_{fay}, K_{faz}\right) \in \mathbb{R}^{3\times3}$ représente la matrice des coefficients de frottements aérodynamiques. $K_{ft} = diag\left(K_{ftx}, K_{fty}, K_{ftz}\right) \in \mathbb{R}^{3\times3}$ représente la matrice des coefficients de traînée de translation. $J_r \in \mathbb{R}$ est l'inertie du rotor. La position du quadrirotor est représentée par x, y et z. Le roulis ϕ, le tangage θ, et le lacet ψ décrivent l'orientation du quadrirotor. d représente les perturbations appliquées au robot quadrirotor. u_x et u_y sont deux entrées de commande virtuelles.

$$\begin{cases} u_x = \cos\phi.\sin\theta.\cos\psi + \sin\phi.\sin\psi \\ u_y = \cos\phi.\sin\theta.\sin\psi - \sin\phi.\cos\psi \end{cases} \tag{IV.19}$$

u_1, u_2, u_3 et u_4 sont les entrées de commande du système (voir la figure IV.3) qui sont écrites en fonction des vitesses angulaires des quatre moteurs comme suit:

$$\begin{aligned} u_1 &= C_p\,(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ u_2 &= C_p\,(-\omega_1^2 + \omega_3^2) \\ u_3 &= C_p\,(-\omega_2^2 + \omega_4^2) \\ u_4 &= C_d\,(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{aligned} \tag{IV.20}$$

avec,

$$\overline{\Omega} = (\omega_1 - \omega_2 + \omega_3 - \omega_4) \tag{IV.21}$$

où $\overline{\Omega}$ sont les couples gyroscopiques totaux qui affectent le quadrirotor.

La modélisation dynamique (IV.20) est complétée par les contraintes d'entrées de commande suivantes [108]:

$$\begin{array}{r} 0 \le u_1 \le 4C_p\, \omega_{\max}^2 \\ -C_p\, \omega_{\max}^2 \le u_2 \le C_p\, \omega_{\max}^2 \\ -C_p\, \omega_{\max}^2 \le u_3 \le C_p\, \omega_{\max}^2 \\ -2C_d\, \omega_{\max}^2 \le u_4 \le 2C_d\, \omega_{\max}^2 \end{array} \tag{IV.22}$$

où $\omega_{\max}$ est la vitesse angulaire maximale du rotor.

A partir de (IV.20), nous extrayons deux contraintes non-holonomes (IV.23), qui représentent les angles de roulis et de tangage désirés, qui seront utilisés comme référence pour synthétiser la commande d'attitude (voir figure IV.3).

$$\begin{cases} \phi_d = \arcsin\left(u_x \sin(\psi_d) - u_y \cos(\psi_d)\right) \\ \theta_d = \arcsin\left(\left(u_x \cos(\psi_d) + u_y \sin(\psi_d)\right) \big/ \cos(\phi_d)\right) \end{cases} \tag{IV.23}$$

Généralement, un moteur DC standard est un système du 2ème ordre, il est possible de modéliser la dynamique d'un moteur à courant continu en tant que système du premier ordre [39]. Dans cette partie, Un moteur à courant continu DC (IV.24) est utilisé:

$$G(s) = \frac{k_{mi}}{\tau_{mi}\, s + 1}, \ (i = \overline{1,4}) \tag{IV.24}$$

où k_{mi} et τ_{mi} sont le gain et la constante de temps du moteur, respectivement.

Soit $X = \left[\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z}\right]^T \in \mathbb{R}^{12}$ et $u = [u_1, u_2, u_3, u_4]^T \in \mathbb{R}^4$ le vecteur l'état et la commande, respectivement. Le modèle dynamique (IV.18) peut être écrit en utilisant la méthode de l'espace d'état comme suit:

$$\dot{X} = f(X) + g(X) + d \tag{IV.25}$$

où

$f(X) = [f_1(X), f_2(X), f_3(X), f_4(X), f_5(X), f_6(X)]^T$, $g(X) = [g_1(X), g_2(X), g_3(X), g_4(X), g_5(X), g_6(X)]^T$

dans lesquels :

$$f_1(X)=\begin{pmatrix} x_2 \\ a_1x_4x_6+a_2x_2^2+a_3\overline{\Omega}x_4 \end{pmatrix},\quad g_1(X)=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & b_1 & 0 & 0 \end{bmatrix}$$

$$f_2(X)=\begin{pmatrix} x_4 \\ a_4x_2x_6+a_5x_4^2+a_6\overline{\Omega}x_2 \end{pmatrix},\quad g_2(X)=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & b_2 & 0 \end{bmatrix}$$

$$f_3(X)=\begin{pmatrix} x_6 \\ a_7x_2x_4+a_8x_6^2 \end{pmatrix},\qquad g_3(X)=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_3 \end{bmatrix}$$

$$f_4(X)=\begin{pmatrix} x_8 \\ a_9x_8 \end{pmatrix},\qquad g_4(X)=\begin{bmatrix} 0 & 0 & 0 & 0 \\ \dfrac{u_x}{m} & 0 & 0 & 0 \end{bmatrix}$$

$$f_5(X)=\begin{pmatrix} x_{10} \\ a_{10}x_{10} \end{pmatrix},\qquad g_5(X)=\begin{bmatrix} 0 & 0 & 0 & 0 \\ \dfrac{u_y}{m} & 0 & 0 & 0 \end{bmatrix}$$

$$f_6(X)=\begin{pmatrix} x_{12} \\ a_{11}x_{12}-g \end{pmatrix},\quad g_6(X)=\begin{bmatrix} 0 & 0 & 0 & 0 \\ \dfrac{\cos x_1.\cos x_3}{m} & 0 & 0 & 0 \end{bmatrix}$$

où

$a_{1}= (I_y-I_z)/I_x$, $a_{2}= -K_{fax}/I_x$, $a_{3}= -J_r/I_x$, $a_{4}= (I_z-I_x)/I_y$, $a_{5}= -K_{fay}/I_y$,
$a_{6}= J_r/I_y$, $a_{7}= (I_x-I_y)/I_z$, $a_{8}= -K_{faz}/I_z$, $a_{9}= -K_{ftx}/m$, $a_{10}= -K_{fty}/m$
$a_{11}= -K_{ftz}/m$, $b_{1}= l/I_x$, $b_{2}= l/I_y$, $b_{3}= 1/I_z$.

Nous obtenons la représentation d'état suivante :

$$\begin{cases} \dot{x}_1=x_2 \\ \dot{x}_2=a_1x_4x_6+a_2x_2^2+a_3\overline{\Omega}x_4+b_1u_2+d \\ \dot{x}_3=x_4 \\ \dot{x}_4=a_4x_2x_6+a_5x_4^2+a_6\overline{\Omega}x_2+b_2u_3+d \\ \dot{x}_5=x_6 \\ \dot{x}_6=a_7x_2x_4+a_8x_6^2+b_3u_4+d \\ \dot{x}_7=x_8 \\ \dot{x}_8=a_9x_8+u_xu_1/m+d \\ \dot{x}_9=x_{10} \\ \dot{x}_{10}=a_{10}x_{10}+u_yu_1/m+d \\ \dot{x}_{11}=x_{12} \\ \dot{x}_{12}=a_{11}x_{12}+\cos(x_1)\cos(x_3)u_1/m-g_a+d \end{cases} \tag{IV.26}$$

Pour synthétiser les lois de commande, les hypothèses suivantes sont retenues dans ce chapitre : Les angles de tangage, de roulis et de lacet doivent satisfaire les inégalités suivantes $-\pi/2\le\phi(t)\le\pi/2$, $-\pi/2\le\theta(t)\le\pi/2$, $-\pi\le\psi(t)\le\pi$. Le vecteur d'état est supposé disponible à la mesure. Les perturbations d sont inconnues mais bornées, c'est-à-dire $\|d\| \leq D$, où $D > 0$.

L'objectif est de synthétiser une loi de commande robuste afin que le vecteur d'état $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\}^T = \{\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z}\}^T \in \mathbb{R}^{12}$ peut suivre une référence donnée $X_d = \{x_{d1}, x_{d2}, x_{d3}, x_{d4}, x_{d5}, x_{d6}, x_{d7}, x_{d8}, x_{d9}, x_{d10}, x_{d11}, x_{d12}\}^T = \{\phi_d, \dot{\phi}_d, \theta_d, \dot{\theta}_d, \psi_d, \dot{\psi}_d, x_d, \dot{x}_d, y_d, \dot{y}_d, z_d, \dot{z}_d\}^T \in \mathbb{R}^{12}$ dans un temps fini, même en présence de perturbations externes élevées dans le modèle dynamique.

IV.4.2.1. Commande d'attitude via backstepping-mode-glissant floue type-1

Pour synthétiser la commande d'attitude (ou d'orientation), nous définissons l'erreur de poursuite entre lacet réel et le lacet désiré suivante :

$$e_1 = x_1 - \phi_d \tag{IV.27}$$

sa dérivée temporelle est:

$$\dot{e}_1 = \dot{x}_1 - \dot{\phi}_d = x_2 - \dot{\phi}_d = e_2 \tag{IV.28}$$

Étape 1 : Définir la surface de glissement

La surface de glissement est conçue comme suit [3]:

$$s_\phi(e_1) = \dot{e}_1 + c_1 e_1 \tag{IV.29}$$

où $c_1 \geq 0$ est un nombre réel positif.

Étape 2 : Synthèse de la loi de commande

L'objectif de la commande est de ramener les états du système à leurs références. Pour cela, considérons la première fonction de Lyapunov L_1 :

$$L_1 = \frac{1}{2} e_1^2 \tag{IV.30}$$

Ensuite, sa dérivée est donnée par :

$$\dot{L}_1 = e_1 \dot{e}_1 = e_1 (x_2 - \dot{\phi}_d) \tag{IV.31}$$

Afin de réaliser $\dot{L}_1$ définie négative $(\dot{L}_1 \leq 0)$, en utilisant la technique de backstepping, nous considérons le système virtuel suivant :

$$x_2 = s_\phi - c_1 e_1 + \dot{\phi}_d \tag{IV.32}$$

alors,

$$s_\phi = x_2 + c_1 e_1 - \dot{\phi}_d \tag{IV.33}$$

En tenant compte des équations (IV.30) et (IV.31), on trouve l'expression de $\dot{L}_1$ suivante:

$$\dot{L}_1 = e_1 s_\varphi - c_1 e_1^2$$

Si $s_\phi = 0$ alors $\left(\dot{L}_1 \leq 0\right)$. Par conséquent, la prochaine étape est requise.

La deuxième fonction de Lyapunov est donnée par :

$$L_2 = L_1 + \frac{1}{2} s_\phi^2 \tag{IV.34}$$

sa dérivée sera donc :

$$\dot{L}_2 = \dot{L}_1 + s_\varphi \dot{s}_\varphi \tag{IV.35}$$

À partir de (IV.33), on a

$$\begin{aligned} \dot{s}_\phi &= \dot{x}_2 + c_1 \dot{e}_1 - \ddot{\phi}_d \\ &= f_1(X) + g_1(X) u_2 + d + c_1 \dot{e}_1 - \ddot{\phi}_d \\ &= a_1 x_4 x_6 + a_2 x_2^2 + a_3 \overline{\Omega}\, x_4 + b_1 u_2 + d + c_1 \dot{e}_1 - \ddot{\phi}_d \end{aligned}$$

L'équation (IV.35) devient :

$$\dot{L}_2 = e_1 s_\phi - c_1 e_1^2 + s_\phi \left(a_1 x_4 x_6 + a_2 x_2^2 + a_3 \overline{\Omega}\, x_4 + b_1 u_2 + d + c_1 \dot{e}_1 - \ddot{\phi}_d \right)$$

Pour que le système atteigne sa stabilité, il faut que $\dot{L}_2$ soit négative.

Supposons que les variables d'état dans (IV.26) sont disponibles. A fin de réaliser $\dot{L}_2$ défini négative, la loi de commande backstepping-mode-glissant floue type-1 pour le roulis sera choisi comme:

$$u_2 = \frac{1}{b_1}\left(-\hat{k}_1 \mathrm{sign}(s_\phi) - a_1 x_4 x_6 - a_2 x_2^2 - a_3 \overline{\Omega}\, x_4 - d - c_2 s_\phi - e_1 - c_1 e_2 + \ddot{\phi}_d \right) \tag{IV.36}$$

où $c_2 \geq 0$, est un nombre réel positif. Par conséquent, (IV.35) devient :

$$\dot{L}_2 = -\left(c_1 e_1^2 + c_2 s_\phi^2 + s_\phi d + \hat{k}_1 \left|s_\phi\right|\right) \leq 0 \tag{IV.37}$$

ce qui donne, e_1 tend vers zéro et $\dot{e}_1$ tend vers zéro quand t tend vers l'infini. Par conséquent, la stabilité du sous-système en boucle fermée le long de la surface de glissement $s_\phi = 0$ est garantie.

Des étapes similaires peuvent être suivies pour synthétiser des lois FBSMC pour le tangage et le lacet. Les lois de commande correspondantes sont données par :

$$u_3 = \frac{1}{b_2}(-\hat{k}_2 \operatorname{sign}(s_\theta) - a_4 x_2 x_6 - a_5 x_4^2 - a_6 \overline{\Omega}\, x_2 - d - c_4 s_\theta - e_3 - c_3 e_4 + \ddot{\theta}_d) \tag{IV.38}$$

$$u_4 = \frac{1}{b_3}(-\hat{k}_3 \operatorname{sign}(s_\psi) - a_7 x_2 x_4 - a_8 x_6^2 - d - c_6 s_\psi - e_5 - c_5 e_6 + \ddot{\psi}_d) \tag{IV.39}$$

où $c_i \geq 0,\ i = \overline{3,6}$ sont des nombres réels positifs.

IV.4.2.2. Commande en translation via backstepping-mode-glissant floue type-1

1) Commande d'altitude

La commande d'altitude via FBSMC peut être obtenue par les mêmes procédures :

$$u_1 = \frac{m}{\cos(x_1)\cos(x_3)}(-\hat{k}_6 \operatorname{sign}(s_z) - a_{11} x_{12} + g_a - d - c_{12} s_z - e_{11} - c_{11} e_{12} + \ddot{z}_d) \tag{IV.40}$$

où $c_i \geq 0,\ i = \{11{,}12\}$ sont des nombres réels positifs.

2) Commande de position

À partir d'équations dynamique (IV.18), on peut voir que le mouvement à travers les axes x et y dépende de u_1. En fait, u_1 est le vecteur de poussée totale orienté pour obtenir le mouvement linéaire désiré, en considérant que, u_x et u_y sont la direction de u_1 responsable du mouvement par les axes x et y, respectivement. En utilisant l'FBSMC, les commandes au sens des déplacements x et y sont obtenus avec les mêmes étapes décrites ci-dessous

$$u_x = \frac{m}{u_1}(-\hat{k}_4 \text{sign}(s_x) - a_9 x_8 - d - c_8 s_x - e_7 - c_7 e_8 + \ddot{x}_d) \tag{IV.41}$$

$$u_y = \frac{m}{u_1}(-k_5 \text{sign}(s_y) - a_{10} x_{10} - d - c_{10} s_y - e_9 - c_9 e_{10} + \ddot{y}_d) \tag{IV.42}$$

où $c_i \geq 0,\ \{i = \overline{7,10}\}$ sont des nombres réels positifs.

La simplification de toutes les étapes de calcul concernant les erreurs de poursuites, les surfaces de glissement et les fonctions de Lyapunov sont définies comme suit:

$$\begin{cases} e_i = x_i - x_{id} \\ e_{i+1} = \dot{e}_i \end{cases},\ i \in [1,\ 11],\ \begin{cases} s_\phi = e_2 + c_1 e_1 = x_2 + c_1 e_1 - \dot{\phi}_d \\ s_\theta = e_4 + c_3 e_3 = x_4 + c_3 e_3 - \dot{\theta}_d \\ s_\psi = e_6 + c_5 e_5 = x_6 + c_5 e_5 - \dot{\psi}_d \\ s_x = e_8 + c_7 e_7 = x_8 + c_7 e_7 - \dot{x}_d \\ s_y = e_{10} + c_9 e_9 = x_{10} + c_9 e_9 - \dot{y}_d \\ s_z = e_{12} + c_{11} e_{11} = x_{12} + c_{11} e_{11} - \dot{z}_d \end{cases},\ L_i = \begin{cases} \frac{1}{2} e^2 & i \in \{1,3,5,7,9,11\} \\ L_{i-1} + \frac{1}{2} s_{i-1}^2 & i \in \{2,4,6,8,10,12\} \end{cases}$$

IV.5. Simulations et résultats

Afin d'évaluer les performances de la commande proposée, des simulations numériques sont donnés dans cette section. Les paramètres nominaux du quadrirotor peuvent être trouvés dans [135][136] et sont présentés dans le tableau A.1 dans l'annexe A. L'algorithme proposé, appliqué au quadrirotor ci-dessous, est simulé sur un PC en utilisant l'environnement MatLab/Simulink. Les conditions initiales du quadrirotor sont $X(0) = [0,1,0,2,0,2,1,0,1,2,0,2]^T$, les facteurs dynamiques des moteurs sont supposés être $k_m = 1,\ \tau_m = 0.15$, et $\omega_{\max} = 200\text{rad/s}$. Les incertitudes ont été injectées dans la structure pour vérifier la robustesse du contrôleur : Ils sont des perturbations externes élevées $d_i(\text{t}) = 5\exp\left(\frac{-(\text{t}-0.03)^2}{10^2}\right) \times \sin(\pi/4\,\text{t}) \times I_{6\times1}$, où $I_{6\times1}$ est une matrice d'identité et ils seront appliqués à $t \geq 5$, la limite supérieure des perturbations est supposée être $D = \max(|d|) = 5$ et $G = 0.01$. Le point désiré a été choisie comme : $x_d = 2\text{m},\ y_d = 2\text{m},\ z_d = 5\text{m},\ \psi_d = 45^\circ$.

Les paramètres du contrôleur sont sélectionnés comme suit : $\underline{k} = diag(k_1, \ldots, k_{12}) = diag(7, 7, 5, 5, 5, 5)$; $c = diag(c_1, \ldots, c_{12}) = diag(3, 20, 3, 20, 3, 1, 1, 4, 1, 4, 1, 1.5)$.

Le schéma global de la structure de commande est schématisé sur la figure IV.6.

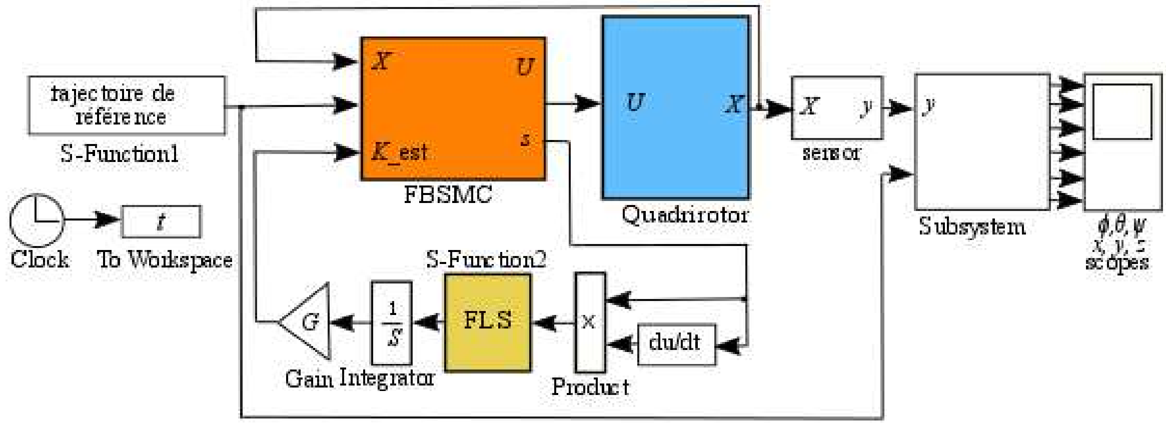


Fig. IV.6. Schémas de la commande globale

Les figures IV.7 et IV.8 montrent les résultats obtenus en simulation.

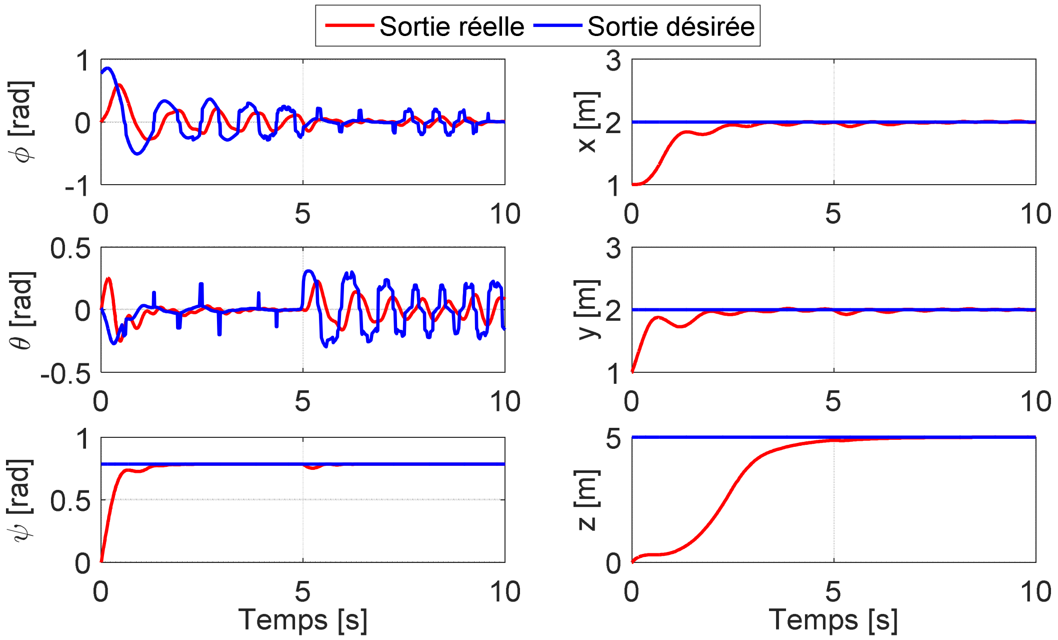


Fig. IV.7. Evolution des sorties du système sous perturbations élevées, en utilisant BSMC

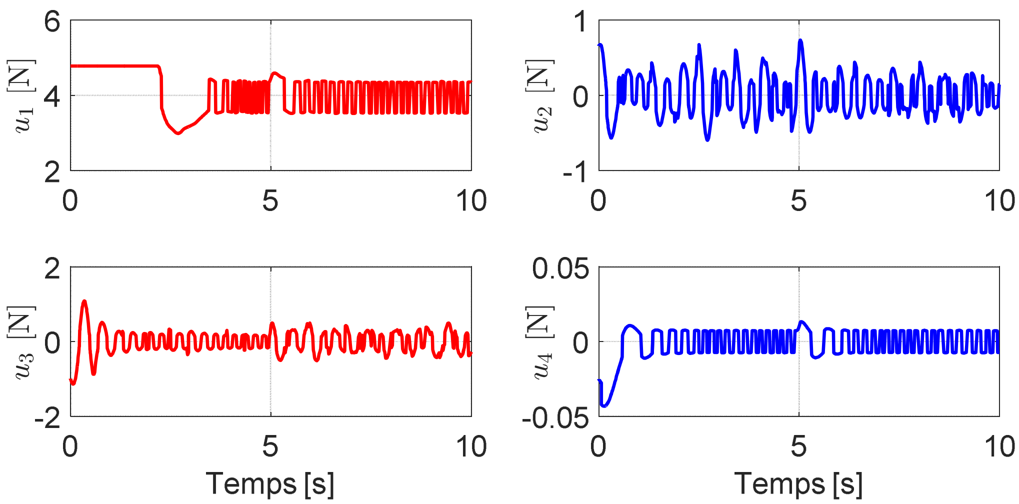


Fig. IV.8. Evolution des commandes BSMC

Il est important de remarquer que les angles de roulis et de tangage ne sont pas nuls, même lorsque le véhicule a rejoint la position de référence. Ce phénomène est dû à la présence des perturbations externes élevées qui viennent perturber le quadrirotor. Cependant, comme on peut le voir sur la figure IV.8, les signaux de commande présentent des discontinuités. Donc, les performances du contrôle ne sont pas satisfaisantes en raison du phénomène de broutement provoqué par la sélection inappropriée des gains de commutation. Afin de résoudre ce problème, et pour améliorer la poursuite des trajectoires, nous avons remplacé la loi de commande (IV.6) par (IV.16) en utilisant les équations (IV.36), (IV.38)-(IV.40). Les figures IV.9 – IV.12 montrent les résultats obtenus en simulation.

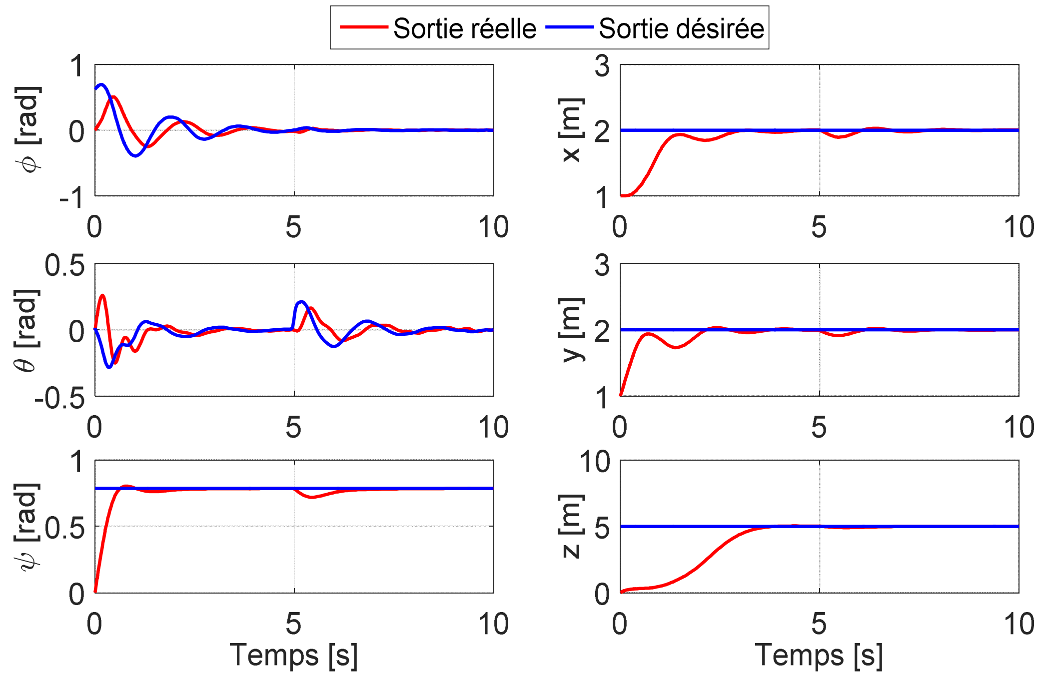


Fig. IV.9. Evolution des sorties du système sous perturbations élevées, en utilisant FBSMC

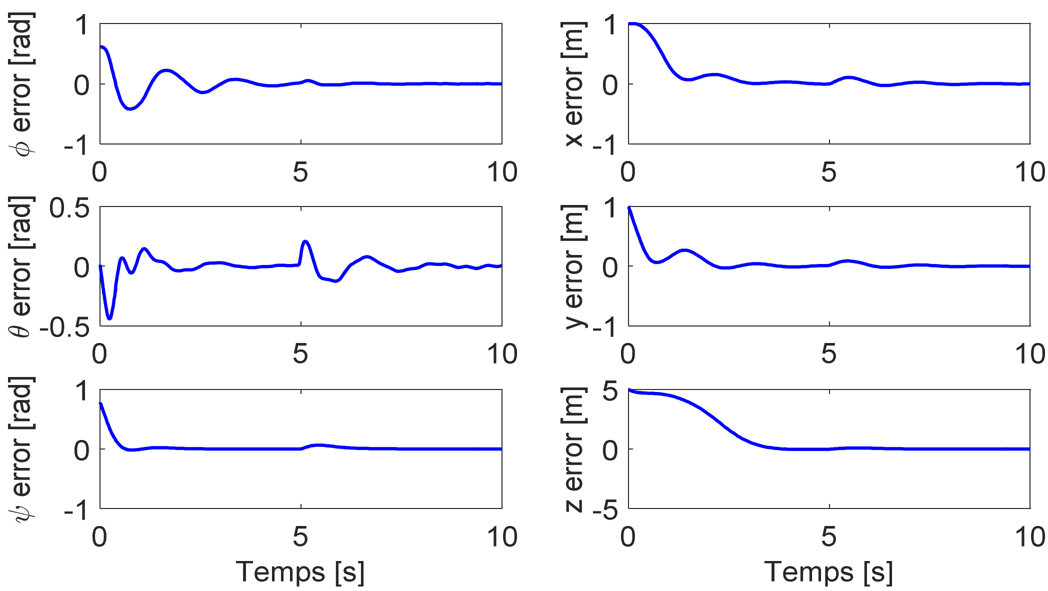


Fig. IV.10. Erreurs de poursuite pour FBSMC

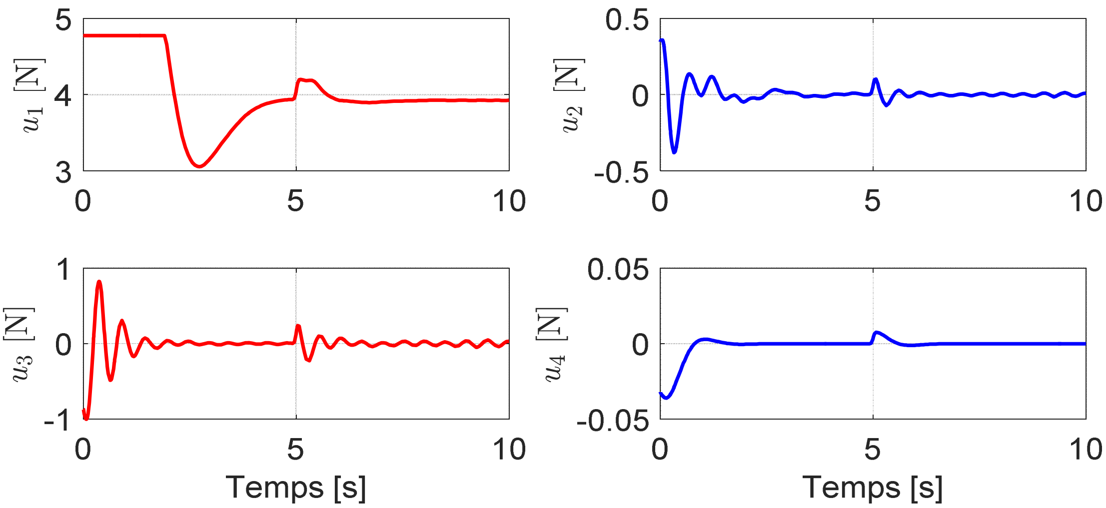


Fig. IV.11. Evolution des commandes FBSMC

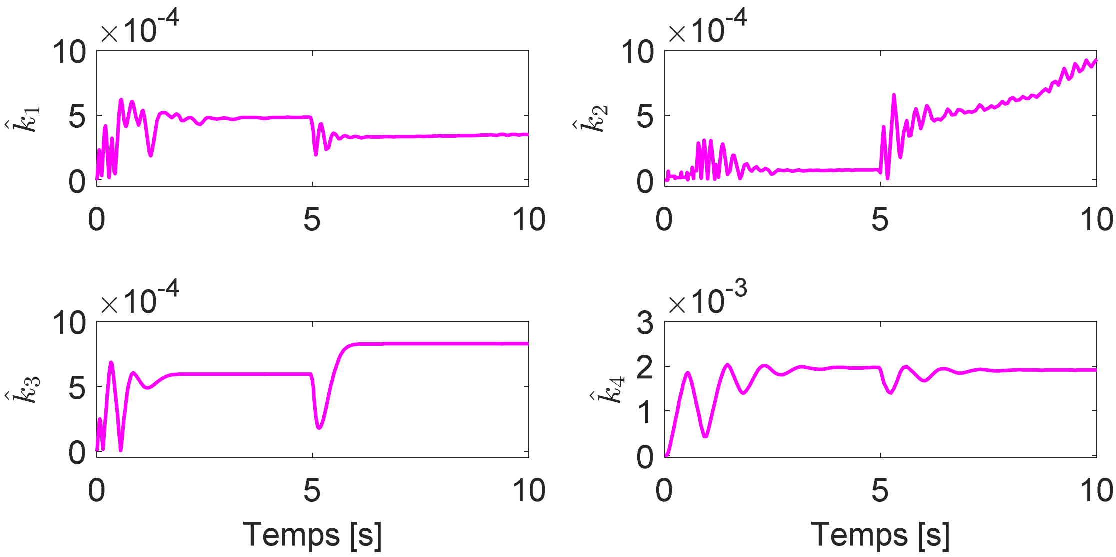


Fig. IV.12. Evolution des gains flous adaptés $\hat{k}_i$ pour FBSMC

Les figures IV.9 à IV.12 montrent les résultats de simulation correspondants à la performance du contrôleur backstepping-mode-glissement flou type-1. La figure IV.9 montre l'évolution des sorties du système par rapport à ses trajectoires de référence, et montre que la performance et la robustesse du contrôleur proposé, en présence de perturbations externes élevées, sont très acceptables. Les états convergent vers leurs références, ce qui montre une poursuite satisfaisante. La Figure IV.10 représente les erreurs de poursuite, qui tendent toutes vers zéro après un temps fini avec une convergence parfaite. Les nouvelles entrées de commande obtenues sont représentées dans la figure IV.11, comparées à la précédente dans la figure IV.8; on constate que les commandes sont lisses, et le phénomène de broutement est presque disparu. Par rapport aux études précédentes, par exemple, [3][108][139][20], la commande proposée réduit efficacement le phénomène de broutement et délivre une bonne réponse dynamique. Donc, à partir de ces résultats, nous constatons que nous avons de bonnes performances de poursuite réalisées grâce aux signaux de commande qui sont bornés et de forme lisse. L'évolution des gains de commutation floue adaptés sont représentés sur la figure IV.12. Il est important de souligner que les résultats sont aussi satisfaisants, même avec des plus grandes perturbations.

IV.6. Conclusion

Dans ce chapitre, après en avoir exposé une brève présentation du contexte historique et opérationnel, nous avons exposé l’état de l'art sur les différentes stratégies de commande mises en œuvre pour remplir les différentes missions du quadrirotor. Ensuite, nous avons proposé une loi de commande robuste pour la commande d’altitude et d’attitude d’un quadrirotor via backstepping-mode-glissant floue type-1, afin de garantir une bonne performance de poursuite en présence de perturbation externes élevées. Cette structure de commande a été capable d’assurer de bonnes performances de poursuite et de garantir la stabilité du système en boucle fermée. L’approche proposée combine trois termes, le premier est le mode glissant qui sert à stabiliser la dynamique du quadrirotor, le deuxième utilise la technique backstepping pour synthétiser la stabilité de la commande par mode glissant, le troisième est un système flou type-1 pour adapter le gain de commutation pour réduire le phénomène de broutement. La robustesse vis-à-vis des perturbations externes élevées est assurée par la commande backstepping-mode-glissant. La loi de commande a été déduite à travers l’analyse de la stabilité au sens de Lyapunov en boucle fermée. Enfin, les résultats de simulation obtenus montrent de bonnes performances de poursuite et de robustesse, et mettent en évidence la capacité et l’efficacité du système flou développé.

Chapitre V

Commande linéarisante robuste par mode glissant floue type-1 avec observateur optimisé : Application au bras manipulateur

Sommaire

Commande linéarisante robuste par mode glissant floue type-1 avec observateur optimisé : Application au bras manipulateur

Votre action positive combinée avec un esprit positif résultera en succès.

Shiv Khera.

V.1. Introduction

Les progrès enregistrés ces deux dernières décennies dans la théorie de la commande des systèmes non-linéaires ont donnés naissance à certaines méthodes systématiques de synthèse de lois de commande non-linéaires. L'une des méthodes de commande non-linéaires les plus connues est la commande dite linéarisante ou linéarisation par bouclage (*feedback linearization*, en anglais). Ce type de commande a fait son apparition dans les années 80s avec les travaux d'Isidori et les apports de la géométrie différentielle [1].

Cette technique de commande consiste à linéariser le système par compensation et appliquer à ce nouveau système une commande linéaire classique telle que la commande par retour d'état. Malheureusement, ce type de commande ne permet pas de maintenir de bonne performance de poursuite en présence d'incertitudes [1][2]. Pour résoudre ce problème, plusieurs travaux ont été focalisés sur la combinaison de la commande linéarisante avec d'autres techniques telle que : les Réseaux neuronaux, la technique H_∞, la logique floue, backstepping, etc. [140][141][142][143][144][28][29][145].

Cependant, pour améliorer la performance de poursuite et assurer la robustesse du système bouclé vis-à-vis des incertitudes structurelles et des perturbations externes, la plupart des approches de commande qui ont été proposées utilisent un terme de commande robuste. Dans la littérature, ce terme est soit une commande par mode de glissement et/ou par backstepping [3][32][146], soit un terme de commande par l'optimisation H_∞ [147][148] et/ou un terme de commande en utilisant l'approche LMI [27]. Cependant, cette approche de commande reste incontournable pour résoudre certains problèmes de commande [149][21].

L'élaboration d'une loi de commande, nécessite souvent l'accès à la valeur d'un ou de plusieurs de ses états. Ceci n'est pas toujours possible à cause de l'inaccessibilité d'état et/ou le manque des capteurs. En plus les mesures des capteurs sont souvent entachées de bruit, ce qui limite les performances d'une boucle de commande. Pour des raisons physiques ou économiques, on a recours à un système dynamique auxiliaire ; appelé observateur, qui est chargé d'estimer l'état du système à partir des entrées appliquées et des mesures fournies par des capteurs physiques. Dans le cas des systèmes non-linéaires, le problème de synthèse d'observateur est encore un problème ouvert et difficile à résoudre. Malgré cette difficulté, la résolution de ce problème est cruciale dans l'ingénierie de contrôle, ce qui a incité les chercheurs à développer des observateurs non-linéaires. Pour cette raison, plusieurs algorithmes sur ce sujet peuvent être trouvés dans la littérature, à savoir : l'observateur de Luenberger étendu [74][150], le filtre de Kalman étendu [151][152][153], l'observateur mode glissant [75][81], l'observateur basé sur la technique MRAS [154], l'observateur basé sur les réseaux de neurones [76] et l'observateur basé sur la logique floue [155][77].

Parmi ces algorithmes, EKF fournit l'estimateur d'état optimal en raison de sa capacité à considérer les incertitudes stochastiques. EKF est un algorithme récursif basé sur la connaissance des statistiques des bruits de mesure et d'état. Comparé à d'autres observateurs non-linéaires [84][85], l'algorithme EKF a un meilleur comportement dynamique, une possibilité de traiter d'une manière relativement objective le cas des bruits dont on est capable de caractériser leurs propriétés statistiques, et il peut fonctionner même en présence de conditions d'arrêt. La performance d'estimation est le problème majeur associé à EKF, il a forte influence sur les paramètres du système et les matrices de covariance du bruit d'état et du bruit de mesure Q et R, respectivement.

D'après la théorie du filtre de Kalman, les matrices de covariances Q et R doivent être obtenues en considérant les propriétés stochastiques des bruits correspondants [156], c'est pour ça la plupart des cas, Q et R sont des matrices généralement inconnues. Cependant, comme ceux-ci ne sont généralement pas connus, dans la plupart des cas, les matrices de covariance sont utilisées comme paramètres de pondération (ajustement des paramètres). De plus, ces matrices ont d'abord été ajustées de manière empirique par la méthode essai-erreur, qui est une méthode très fastidieuse en raison d'une consommation de temps importante [157]. Pour éviter la complexité computationnelle de la méthode essai-erreur, et lorsque les valeurs de ces matrices ne sont pas connues précisément, l'amélioration de performance de EKF peut être assimilée à un problème d'optimisation. Les auteurs dans [151] ont utilisé des GAs pour optimiser ces matrices. Plusieurs études ont montré que l'algorithme PSO peut produire des solutions de haute qualité dans un temps de calcul plus court, facile à mettre en œuvre, et converge plus vite par rapport à GA [158][87].

Motivé par les travaux de (Laamari *et al.*) [87], ce chapitre, a pour objectif de construire un schéma de commande et d'observation assurant à la fois, robustesse et performance, adaptés à une classe des systèmes non-linéaires (robot manipulateur).

Premièrement, nous proposons une loi de commande pour une classe de systèmes non-linéaires. La loi de commande proposée consiste en une loi de commande linéarisante classique augmentée d'une composante de robustification par mode glissant, où le gain de commande est un système flou type-1. La combinaison de ces techniques avec celles de la commande linéarisante permet de concevoir une commande linéarisante robuste vis-à-vis aux variations paramétriques du système et aux perturbations externes pour une classe de systèmes non-linéaires. Dans l'approche proposée, la commande linéarisante est d'abord appliquée en utilisant le placement de pôles. Ensuite, une composante de robustification par la commande discontinue est additionnée (la commande discontinue peut être trouvée dans la commande par mode glissant), pour assurer la robustesse de la structure de commande par rapport aux effets des variations paramétriques et des perturbations externes. Nous allons encore introduire un système flou type-1 afin d'adapter le gain de commutation pour réduire/ou faire disparaitre le phénomène de broutement induit par la commande discontinue dans le mode glissant. Deuxièmement, afin d'améliorer la performance de l'observateur EKF pour garantir la stabilité et la poursuite en boucle fermée, les matrices de covariance Q et R doivent être optimisées, cette dernière est assurée par une méthode évolutif inspiré par les interactions sociales qui est l'algorithme d'optimisation PSO.

Afin de tester la faisabilité du schéma de commande proposé et d'évaluer la performance en poursuite de trajectoires, une application à la commande d'un bras manipulateur a été effectuée. Les résultats de simulation pour des testes de performance et de robustesse seront également présentées.

V.2. Contexte et formulation

Concéderons un système non-linéaire incertain donné par :

$$\begin{cases} \dot{X} = f(X,t) + g(X,t)u + d(t) + w(t) \\ Y = h(X,t) + v(t) \end{cases} \tag{V.1}$$

où $X \in \mathbb{R}^n$ est l'espace d'état, $u \in \mathbb{R}^n$ est le vecteur d'entrée de commande, $Y \in \mathbb{R}^m$ est le vecteur de mesure. Les fonctions $f(X)$, $g(X)$ et $h(X)$ sont des champs de vecteurs. d représente la somme des incertitudes paramétriques et des perturbations externes, supposée bornée. w_k et v_k sont respectivement, des bruits d'état et de mesure, supposés blancs et Gaussiens avec des moyennes nulles, de covariances respectives $Q = E[w_k, w_k^T]$ et $R = E[v_k, v_k^T]$. Nous supposons que le vecteur d'état du système $X = (x_1, x_2, ..., x_n)^T = \left(x, \dot{x}, ..., x^{(n-1)}\right)^T \in R^n$ est indisponible à la mesure.

Notre objectif par la suite sera de concevoir une loi de commande $u = u\left(\hat{X}\right)$ tel que même en présence des variations paramétriques et des perturbations externes, le vecteur d'état X suivra une trajectoire de référence désirée bornée $X_d = \left(x_d, \dot{x}_d, ..., x_d^{(n-1)}\right)^T$.

Ainsi, notre objectif est de déterminer la commande $u\left(\hat{X}\right)$ qui permet de conduire l'erreur de poursuite estimée $e(t)$ définie par :

$$e = \hat{x} - x_d \tag{V.2}$$

à converger asymptotiquement vers zéro et garantir la stabilité du système de commande.
Pour atteindre l'objectif de commande, les hypothèses suivantes sont retenues dans ce chapitre.

Hypothèses : Afin de développer notre contrôleur, pour le système (V.1), nous adoptons les hypothèses de simplification et de réalisation suivantes :

- **Hyp V.1:** La fonction f est supposée connue et l'erreur sur son estimation est bornée, c'est-à-dire $\left|f(\hat{X}) - f(X)\right| \le E$ où $f(\hat{X})$ est une estimation de $f(X)$.
- **Hyp V.2:** Le gain d'entrée g est supposé connu, positif et borné, c'est-à-dire $0 < g_{\min} \le g(X) \le g_{\max}$.
- **Hyp V.3:** $span\left\{dh; df \circ h; ....; df^{n-1} \circ h\right\}$ est de rang n.
- **Hyp V.4:** La perturbation d est supposée bornée, *i.e.* $\|d\| \le D$ où $D > 0$.
- **Hyp V.5:** La trajectoire désirée X_d et ses dérivées par rapport au temps, $x_d^{(i)}$, $i = 1, ..., n$ sont supposées bornées.

La condition donnée dans l'hypothèse V.3 est appelée condition du rang d'observabilité (*Observabilité Rang Condition,* en anglais) pour les systèmes non-linéaires. Cette condition vérifie la condition d'observabilité (voir l'annexe C pour plus de détails).

Toutefois, la dynamique du système (V.1) est non-linéaire, de plus les variables d'états ne sont généralement pas toutes disponibles. Une solution est d'utiliser un observateur non-linéaire pour estimer tous les états à partir des mesures disponibles. Dans cette section, nous proposons d'utiliser le filtre de Kalman étendu [86] qui fournit l'estimateur d'état optimal en raison de sa capacité à considérer les incertitudes stochastiques. Cet algorithme met à jour les états du système en ligne en utilisant les observations précédentes et actuelles.

En utilisant les équations décrites par (III.57) à (III.59) qui ont été mentionnées dans la section III.7.2, le filtre de Kalman étendu est mis à jour selon les étapes suivantes :

1ère étape: équations de prédiction :

$$\begin{aligned} \hat{X}_{k+1/k} &= f(\hat{X}_{k/k}, u_k, 0) \\ P_{k+1/k} &= F_k P_{k/k} F_k^T + W_k Q W_k^T \end{aligned} \tag{V.3}$$

avec, F_k et W_k sont les matrices jacobiennes données par:

$$F_k = \left.\frac{\partial f(X,0)}{\partial X}\right|_{X=\hat{X}_k} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, \quad W_k = \left.\frac{\partial f(\hat{X}_k, w)}{\partial w}\right|_{w=0} = \begin{bmatrix} \frac{\partial f_1}{\partial w_1} & \frac{\partial f_1}{\partial w_2} & \cdots & \frac{\partial f_1}{\partial w_n} \\ \frac{\partial f_2}{\partial w_1} & \frac{\partial f_2}{\partial w_2} & \cdots & \frac{\partial f_2}{\partial w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial w_1} & \frac{\partial f_n}{\partial w_2} & \cdots & \frac{\partial f_n}{\partial w_n} \end{bmatrix} \tag{V.4}$$

2ème étape: équations de correction :

$$K_k = P_{k+1/k} H_k^T (H_k P_{k+1/k} H_k^T + V_k R V_k^T)^{-1} \tag{V.5}$$

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_k (Z_k - h(\hat{X}_{k+1/k}, 0)) \tag{V.6}$$

$$P_{k+1/k+1} = P_{k+1/k} - K_k H_k P_{k+1/k} \tag{V.7}$$

avec, H_k et V_k sont les matrices jacobiennes données par:

$$H_k = \left.\frac{\partial h(X,0)}{\partial X}\right|_{x=\hat{x}_k} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \frac{\partial h_n}{\partial x_2} & \cdots & \frac{\partial h_n}{\partial x_n} \end{bmatrix}, \quad V_k = \left.\frac{\partial h(\hat{X}_k, v)}{\partial v}\right|_{v=0} = \begin{bmatrix} \frac{\partial h_1}{\partial v_1} & \frac{\partial h_1}{\partial v_2} & \cdots & \frac{\partial h_1}{\partial v_n} \\ \frac{\partial h_2}{\partial v_1} & \frac{\partial h_2}{\partial v_2} & \cdots & \frac{\partial h_2}{\partial v_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial v_1} & \frac{\partial h_n}{\partial v_2} & \cdots & \frac{\partial h_n}{\partial v_n} \end{bmatrix} \tag{V.8}$$

où la notation $\hat{X}_{k+1/k}$ dénote le vecteur de prédiction d'état a priori,

$\hat{X}_{k+1/k+1}$ est le vecteur de prédiction d'état postérieur,

$P_{k+1/k}$ désigne la matrice de covariance d'erreur de prédiction a priori,

$P_{k+1/k+1}$ est la matrice de covariance d'erreur de prédiction a posteriori,

K est la matrice du gain.

Le filtre de Kalman se caractérise principalement par deux matrices Q et R qui contiennent la valeur de la covariance des bruits d'état et de mesure, respectivement. D'après la théorie du filtre de Kalman, Q et R doivent être obtenus en considérant les propriétés stochastiques des bruits correspondants [156], c'est pour ça la plupart des cas, Q et R sont des matrices généralement inconnues. Cependant, comme elles ne sont pas connus, dans la plupart des cas elles sont utilisées comme de paramètres de pondération à ajuster. De plus, ces matrices ont d'abord été ajustées de manière empirique par la méthode essai-erreur, qui est une méthode très fastidieuse en raison d'une consommation du temps importante [157]. Pour éviter la complexité computationnelle de cette méthode, et lorsque les valeurs de ces matrices ne sont pas connues précisément, l'amélioration de performance d'EKF peut être assimilée à un problème d'optimisation.

Concernant le problème du contrôle non-linéaire, nous proposons d'utiliser l'approche de la commande linéarisante [3].

La loi de commande par la commande linéarisante avec l'observateur EKF peut être calculée comme suite :

Par l'utilisation d'équation (III.4), en dérivant Y par rapport au temps, on obtient:

$$\dot{Y} = \frac{\partial h}{\partial \hat{X}} f(\hat{X}) + \frac{\partial h}{\partial \hat{X}} g(\hat{X}) u$$

$$\dot{Y} = L_f h(\hat{X}) + L_g h(\hat{X}) u \tag{V.9}$$

où $L_f h(\hat{X}) : \mathbb{R}^n \to \mathbb{R}$ et $L_g h(\hat{X}) : \mathbb{R}^n \to \mathbb{R}$, définis comme des dérivées de Lie de h par rapport à f et g, respectivement. Soit U un ensemble ouvert contenant le point d'équilibre X_0, c'est-à-dire un point où $f(\hat{X})$ devient nul $\left(f(X_0)\right)$. Ainsi, si dans l'équation (V.9), la dérivée de Lie de h par rapport à « $L_g h(\hat{X})$ » est bornée en s'éloignant de zéro pour tout $X \in U$ [159].

La loi de commande est alors défini par :

$$u^* = \frac{1}{L_g h(\hat{X})}\left(-L_f h(\hat{X}) + x_d^{(n)}\right) \tag{V.10}$$

Pour résoudre le problème d'instabilité, la loi de commande par placement de pôles est choisie comme suit [3] :

$$u^* = \frac{1}{L_g h(\hat{X})}\left(-L_f h(\hat{X}) - k_{feed}^T \underline{e} + x_d^{(n)}\right) \tag{V.11}$$

où le vecteur d'état estimé est définie par $\hat{X} = (\hat{x}_1, \hat{x}_2, ..., \hat{x}_n)^T$, $\underline{e} = (e, \dot{e}, ..., e^{(n-1)})^T$ est le vecteur d'erreur de poursuite estimée, $k_{feed} = (k_0, k_1, ..., k_{n-1})^T$ est choisi de sorte que toutes les racines du polynôme $p^n + k_{n-1}\, p^{(n-1)} + ... + k_0$ sont situées dans le demi-plan gauche ouvert du plan complexe.

Si la sortie est $Y = h(X) = x_1$, on obtient : $L_g h(X) = g(X)$ et $L_f h(X) = f(X)$, la loi de commande devient :

$$u^* = \frac{1}{g(\hat{X})}\left(-f(\hat{X}) - k_{feed}^T \underline{e} + x_d^{(n)}\right) \tag{V.12}$$

En combinant la loi de commande (V.12) et le système (V.1), nous obtenons la dynamique d'erreur suivante:

$$e^{(n)} + k_{n-1}\, e^{(n-1)} + ... + k_0\, e = 0 \tag{V.13}$$

dans lequel l'objectif principal est $\lim\limits_{t \to \infty} e(t) = 0$.

La présence d'incertitudes et de perturbations peut perturber le fonctionnement de la commande linéarisante; par conséquent, la dynamique du système peut conduire à des instabilités telles que des erreurs statiques (voir [1]). Pour résoudre ce problème, nous proposons d'augmenter cette commande. En général, l'élaboration d'une loi de commande peut être effectuée principalement en deux étapes, à savoir : le choix de la structure de la commande et la détermination d'une loi de commande assurant la stabilité du système bouclé. Dans notre cas, la commande linéarisante que nous améliorions, s'articule sur un terme de la commande discontinue et un système à logique floue type-1.

V.3. Structure générale de la commande adoptée

L'approche proposée dans ce chapitre est basée sur la synthèse de loi de commande robuste pour une classe des systèmes non-linéaires (V.1) permettant de forcer les sorties du système à suivre ses références, en présence de variations paramétriques et de perturbations externes, comme que schématisée sur la figure V.1.

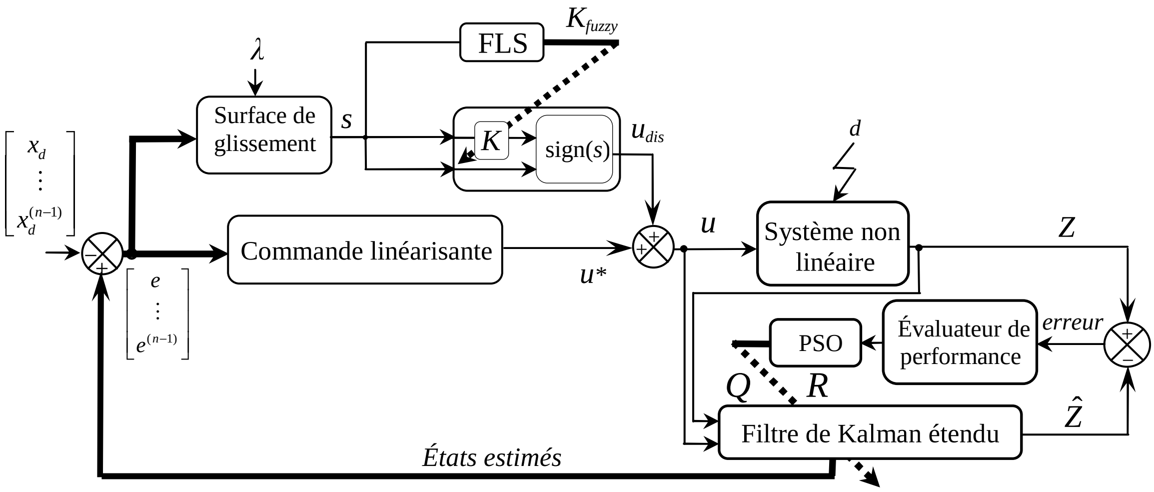


Fig. V.1. Diagramme schématique de commande proposée avec EKF optimisé par PSO

La détermination des matrices de covariance Q et R est une tâche difficile, en particulier lorsque les bruits correspondants ont des propriétés stochastiques inconnues. Pour éviter la complexité computationnelle de la méthode essai-erreur lorsque les valeurs de ces matrices ne sont pas connues précisément, nous considérerons ces matrices comme des paramètres libres à ajuster par l'utilisation de l'algorithme PSO.

Dans cette partie, la tâche principale du PSO est représentée sur la figure V.1, où nous l'exécutons de manière hors-ligne avec un EKF afin de trouver la solution optimale pour les matrices de covariance Q et R. Nous appellerons cette combinaison PSO-EKF. Le critère d'erreur quadratique moyenne (MSE) est utilisé comme fonction objective, entre la sortie réelle et la sortie estimée, selon un certain nombre d'itérations N à effectuer pour chaque étape d'estimation.

L'entrée de commande u et la réponse mesurée Z seront considérées comme des signaux d'entrée pour l'observateur EKF, où u est appliquée à la fois au système non-linéaire et au filtre de Kalman étendu.

La sortie réelle Z et la sortie estimée $\hat{Z}$ sont définies comme étant les entrées d'évaluateur de performance du module PSO via un comparateur. La fonction objective MSE est calculée par l'évaluateur de performance. Ensuite, les valeurs obtenues du MSE seront utilisées dans l'algorithme PSO. Sur la base de ces valeurs, l'optimiseur PSO va calculer et optimiser les paramètres inconnus des matrices de covariance Q et R. Nous obtenons alors, le meilleur ensemble de particules, en mettant à jour les solutions de particules selon les équations (II.1) et (II.2).

Une fois la vitesse de chaque particule calculée, l'équation (II.1) actualise la vélocité du nouveau. La nouvelle position est alors déterminée par la somme de la nouvelle vitesse et de la position précédente par l'équation (II.2).

La nouvelle position et la mise à jour de Q et R sont ensuite utilisées pour adapter EKF à l'itération suivante jusqu'à ce qu'un nombre prédéfini d'itérations soit atteint, c qui nous donne les valeurs optimales de Q et R. Enfin, les matrices Q et R optimisées sont injectés dans l'observateur EKF pour une future exécution en ligne.

V.4. Synthèse de la loi de commande

Deux termes sont additionnés pour améliorer la commande linéarisante. Le premier est un terme de robustification "commande discontinu" u_{dis}. Le deuxième est un système flou type-1, comme indiqué sur la figure V.1, où la commande discontinue peut être obtenue par mode glissant, ce choix est motivé par sa grande robustesse vis-à-vis les incertitudes et les perturbations. Le terme u_{dis} doit forcer la sortie du système à suivre un signal de référence désiré. Donc, la commande résultante sera constituée de deux termes: un terme de la commande linéarisante classique et un terme de la commande discontinue (terme de robustification), elle est donnée par :

$$\begin{aligned} u &= u^* + u_{dis} \\ &= \frac{1}{g(\hat{X})}\left(-f(\hat{X}) - k_{feed}^T\, \underline{e} + x_d^{(n)}\right) - K\, sign(s) \end{aligned} \tag{V.14}$$

où s est la surface de glissement associée à la commande, qui peut être représentée par (III.19).

La condition de glissement (Attractivité de la surface) qui garantit que les états du système atteignent la surface de glissement en temps fini est donnée par [160][3] :

$$\frac{1}{2}\frac{d}{dt}s^2 \le -\eta|s|,\ \eta > 0 \tag{V.15}$$

où

$$\dot{s} = e^{(n)} + \lambda_{n-1}e^{(n-1)} + \ldots + \lambda_2\ddot{e} + \lambda_1\dot{e} \tag{V.16}$$

L'utilisation de la fonction discontinue signe provoquera un phénomène indésirable appelé broutement. Généralement, il existe trois méthodes différentes pour résoudre ce problème :

(i) La première consiste à utiliser le mode de glissement d'ordre supérieur [161][162].

(ii) La deuxième, une couche limite est introduite autour de la surface de glissement [163][164].

(iii) La troisième, concerne l'adaptation du gain selon certaines règles floues appropriées.

Dans ce chapitre, la troisième approche (iii) est adoptée. Dans ce contexte, un gain de commutation élevé K de (V.14) conduira à une augmentation des oscillations du signal d'entrée de commande, et par conséquent une excitation de dynamique à haute fréquence, par conséquent, un phénomène de broutement sera créé. De plus, un faible gain de commutation peut réduire le phénomène de broutement et améliorer la performance de poursuite malgré les incertitudes et les perturbations externes. Cependant, l'augmentation du gain provoque une augmentation des oscillations dans le signal de commande autour de la surface de glissement. Pour atteindre une performance plus appropriée, ce gain doit être ajusté. Ce réglage est basé sur la distance entre les états du système et la surface de glissement, c'est-à-dire que le gain devrait être élevé lorsque la trajectoire d'état est éloignée de la surface de glissement, et lorsque la distance diminue, il devrait être réduit. Cette idée peut être réalisée en combinant la logique floue type-1 avec la commande discontinue pour ajuster le gain K de manière adaptative selon certaines règles floues appropriées. Sous cette condition, le gain K est adapté par un gain flou K_{fuzzy} comme le montre la figure V.1.

Pour cette raison, un FLS avec une seule entrée et une seule sortie est conçu, l'entrée s reflète la distance de la trajectoire d'erreur par rapport à la surface de glissement. La sortie du système flou type-1 est désignée par K_{fuzzy}.

Le FLS se compose de quatre parties: le fuzzificateur, la base de règles, le moteur d'inférence flou et le defuzzificateur. La base de règles est composée d'une collection de règles ***SI… ALORS*** floues dont ils peuvent être énoncées de manière linguistique suivante:

$$\text{Règle } l : \textbf{\textit{SI}} \;\; s \;\text{ est }\; A^l , \textbf{\textit{ALORS}} \; K_{fuzzy} \;\text{ est }\; B^l \;,\; l = 1,2,...,N$$

où A^l et B^l sont des ensembles flous, N est le nombre total de règles.

Notez que les fuzzifications singleton, la défuzzification avec la méthode du centre de gravité, l'implication de Mamdani et le moteur d'inférence de produit sont utilisés dans ce chapitre. Par conséquent, la sortie du système flou type-1 peut être décrite par l'équation suivante:

$$K_{fuzzy}(s) = \frac{\sum_{i=1}^{N} K_{fuzzy\,i} . \mu_{A_i^l}\left(s_i\right)}{\sum_{i=1}^{N} \mu_{A_i^l}\left(s_i\right)} \tag{V.17}$$

où la fonction d'appartenance de l'ensemble de sortie est discrétisée sur N points. $\mu_{A_i^l}\left(s_i\right)$ est le degré d'appartenance résultante. K_{fuzzy_i} est la variable de sortie.

Par conséquent, la loi de commande (V.14) devient:

$$\begin{aligned} u &= u^* + K_{fuzzy} u_{dis} \\ &= \frac{1}{g(\hat{X})}\left(-f(\hat{X}) - k_{feed}^T \underline{e} + x_d^{(n)}\right) - \hat{K}\,\text{sign}\left(s\right) \end{aligned} \tag{V.18}$$

où K_{fuzzy} est la sortie du FLS comme indiqué sur la figure V.1 et donc le gain final devient $\hat{K} = K_{fuzzy} \times K$.

En se basant sur l'hypothèses V.1 à V.4 et en considérant que l'estimation $g(\hat{X})$ pourrait être choisie en fonction de la moyenne géométrique $g(\hat{X}) = \sqrt{g_{\min} g_{\max}}$, les bornes de $g(X)$ peuvent être exprimées par $\beta^{-1} < g(\hat{X}) / g(X) < \beta$ avec $\beta = \sqrt{g_{\max} / g_{\min}}$.

Sous cette condition, le gain $\hat{K}$ doit être choisi en fonction de:

$$\hat{K} \geq \beta\, g^{-1}(\hat{X})\left(\eta + D + E\right) + g^{-1}(\hat{X})\left(\left|\hat{u}_1\right| - \beta\left|\hat{u}_2\right|\right) \tag{V.19}$$

Afin de dominer les états du système pour arriver à la surface de glissement $s = 0$ dans un temps limité et y rester, la loi de commande doit être conçue de sorte que la condition de glissement décrite en (V.15) soit satisfaite. Cet objectif est assuré par le théorème suivant :

Théorème V.1:

Considérant le système non-linéaire incertain (V.1). Supposons que les hypothèses V.1 à V.5 sont satisfaites. Si on choisit la loi de commande (V.18) et considérant $\hat{K}$ *comme (V.19),* *alors, la condition précédente (V.15) est satisfaite, ce qui assure la convergence du vecteur d'erreur de poursuite sur la surface de glissement* s

Preuve:

Considérons la fonction candidate de Lyapunov suivante :

$$L = \frac{1}{2}s^2 \tag{V.20}$$

Sa dérivée temporelle est donnée par :

$$\begin{aligned}
\dot{L} &= \frac{1}{2}\frac{d}{dt}s^2 = s\,\dot{s} = \left(e^{(n)} + \lambda_{n-1}e^{(n-1)} + \ldots + \lambda_2\ddot{e} + \lambda_1\dot{e}\right)s \\
&= \left(\left(x^{(n)} - x^{(n)}{}_d\right) + \lambda_{n-1}e^{(n-1)} + \ldots + \lambda_2\ddot{e} + \lambda_2\dot{e}\right)s \\
&= \left(\sum_{i=1}^{n-1}\lambda_i e^{(i)} + f(X) + g(X)u + d - x_d^{(n)}\right)s \\
&= \left(\sum_{i=1}^{n-1}\lambda_i e^{(i)} + f(X) + g(X)\left[g^{-1}(\hat{X})\left(-f(\hat{X}) - k_{feed}^T\,\underline{e} + x_d^{(n)}\right) - \hat{K}\,\mathrm{sign}(s)\right] + d - x_d^{(n)}\right)s \\
&= \left(\sum_{i=1}^{n-1}\lambda_i e^{(i)} + f(X) + g(X)\left[g^{-1}(\hat{X})\left(-f(\hat{X}) - \sum_{i=1}^{n}K_{i-1}e^{(i-1)} + x_d^{(n)}\right) - \hat{K}\,\mathrm{sign}(s)\right] + d - x_d^{(n)}\right)s
\end{aligned}$$

Notant que: $f(X) = f(\hat{X}) - \left[f(\hat{X}) - f(X)\right]$, $\hat{u}_1 = -f(\hat{X}) - \sum_{i=1}^{n}K_{i-1}e^{(i-1)} + x_d^{(n)}$ et $\hat{u}_2 = -f(\hat{X}) - \sum_{i=1}^{n-1}\lambda_i e^{(i)} + x_d^{(n)}$ on obtient :

$$\dot{L} = \left[-\left(f(\hat{X}) - f(X)\right) + d + g(X)\,g^{-1}(\hat{X})\hat{u}_1 - \hat{u}_2 - g(X)\,K_T\,\mathrm{sign}(s)\right]s \tag{V.21}$$

Par conséquent, en considérant les hypothèses V.1 - V.4 et en définissant $\hat{K}$ selon (V.19), $\dot{L}$ devient

$$\dot{L} \le 0 \tag{V.22}$$

ce qui implique $L(t) \le L(0)$. À partir de la définition de s dans (III.19), on peut vérifier que e est bornée. Ainsi, l'hypothèse V.5, et les équations (III.19) et (V.16) impliquent que s et $\dot{s}$ sont également bornés.

Pour une convergence en temps fini, la condition (V.22) qui ne garantie qu'une convergence asymptotique vers la surface de glissement, est remplacée par une condition plus restrictive dite de η-attractivité et donnée dans [66]:

$$\dot{L} = s\dot{s} \le -\eta|s| \tag{V.23}$$

où η est une constante positive.

Ensuite, en divisant par $|s|$ et en intégrant les deux côtés sur l'intervalle $0 < t < t_s$, où t_s est le temps nécessaire pour atteindre s, on obtient:

$$\int_0^t \frac{s}{|s|}\dot{s}\,d\tau \le -\int_0^t \eta\,d\tau$$

$$|s(t=t_s)| - |s(t=0)| < -\eta\, t_s$$

Dans ce cas on peut assurer que la surface $s=0$ va être rejointe avec un temps finie tel que :

$$t_{conv} \le \frac{|s(t_{conv}) = 0|}{\eta} \tag{V.24}$$

et par conséquent, la convergence à temps fini vers la surface de glissement s, où t_{conv} est le temps nécessaire pour atteindre la surface de glissement s. □

V.5. Simulation et résultats

Pour illustrer la performance de la méthode de commande proposée, nous considérons la commande d'un bras manipulateur à 2-articulations représenté par la figure V.2 où son modèle dynamique est donné par [165] :

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix}\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} \tag{V.25}$$

la matrice $M_0 = [\mathrm{m}_{ij}]_{2\times 2}$ est donnée par :

$$m_{11} = m_1 l_{c1}^2 + m_2\left(l_1^2 + l_{c2}^2 + 2m_2 l_1 l_{c2}\cos(\theta_2)\right) + I_1 + I_2$$

$$m_{12} = m_2\left(l_{c2}^2 + l_1 l_{c2}\cos(\theta_2)\right) + I_2$$

$$m_{22} = m_2 l_2^2 + I_2$$

la matrice $C_0 = [c_{ij}]_{2\times 2}$ est donnée par:

$$c_{11} = a\dot{\theta}_2, \quad c_{12} = a\dot{\theta}_1 + a\dot{\theta}_2, \quad c_{21} = -a\dot{\theta}_1, \quad c_{22} = 0,$$

avec $a = -m_2 l_1 l_{c2}\sin\theta_2$,

le vecteur $G_0 = [\mathrm{G}_1, \mathrm{G}_2]^T$ est donné par:

$G_1 = (m_1 l_{c1} + m_2 l_1)\, g \cos(\theta_1) + m_2 l_{c2}\, \mathrm{g} \cos(\theta_1 + \theta_2)$

$G_2 = m_2 l_{c2}\, \mathrm{g} \cos(\theta_1 + \theta_2)$

Le vecteur $\delta = [\delta_1, \delta_2]^T$ représente les perturbations externes.

En raison d'une erreur de modélisation (variations paramétriques et perturbations externes), on suppose que le modèle dynamique du manipulateur (V.25) présente une certaine incertitude. Par conséquent, peut être écrit par :

$$M(\theta) = M_0(\theta) + \Delta M(\theta) \in R^{2\times 2} \tag{V.26}$$

$$C(\theta,\dot{\theta}) = C_0(\theta,\dot{\theta}) + \Delta C(\theta,\dot{\theta}) \in R^{2\times 2} \tag{V.27}$$

$$G(\theta) = G_0(\theta) + \Delta G(\theta) \in R^{2\times 1} \tag{V.28}$$

où $M_0(\theta)$, $C_0(\theta,\dot{\theta})$ et $G_0(\theta)$ sont des parties nominales, alors que $\Delta M(\theta)$, $\Delta C(\theta,\dot{\theta})$ et $\Delta G(\theta)$ sont les paramètres d'incertitudes.

Le modèle dynamique du robot (V.25) avec des paramètres incertains et des perturbations externes peut être réécrit comme suit:

$$M_0(\theta)\ddot{\theta} + C_0(\theta,\dot{\theta})\dot{\theta} + G_0(\theta) + d(t) = \tau \tag{V.29}$$

où $d(t) = \Delta M(\theta) + \Delta C(\theta,\dot{\theta}) + \Delta G(\theta) + \delta(t) \in R^2$ représente la somme des incertitudes paramétriques et des perturbations externes.

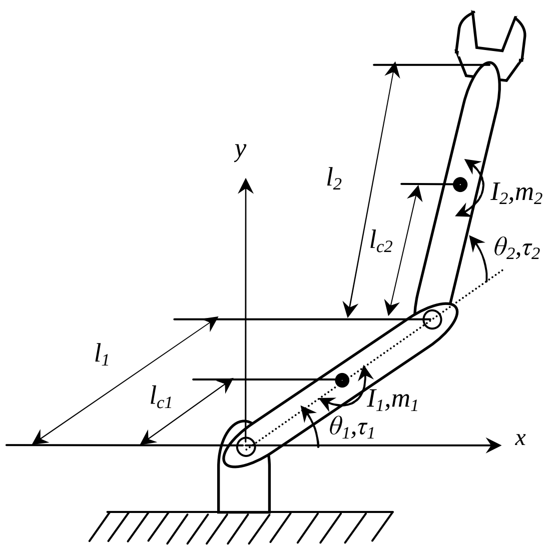


Fig. V.2. Robot manipulateur à 2-articulations

Le modèle dynamique non-linéaire du robot manipulateur (V.29) peut être donné par la représentation non-linéaire générale continue, décrite dans (V.1). Nous écrivons ces équations d'état en considérant le vecteur d'état $X=[\theta_1,\dot{\theta}_1,\theta_2,\dot{\theta}_2]^T=[x_1,x_2,x_3,x_4]^T$ et le vecteur de sortie $y=[\theta_1,\theta_2]^T=[x_1,x_3]^T$. Les équations d'état sont alors données par :

$$\begin{cases}\dot{x}_1=x_2\\ \dot{x}_2=f_1(X)+g_1(X)\tau_1+d_1\\ \dot{x}_3=x_4\\ \dot{x}_4=f_2(X)+g_2(X)\tau_2+d_2\\ y_1=x_1\\ y_2=x_3\end{cases} \tag{V.31}$$

où les fonction $f(X)$ et $g(X)$ sont données comme suit

$$f=\begin{bmatrix}f_1\\ f_2\end{bmatrix}=M_0^{-1}\left(-C_0[\dot{x}_1,\ \dot{x}_2]-G_0\right),\quad g=\begin{bmatrix}g_1\\ g_2\end{bmatrix}=M_0^{-1} \tag{V.32}$$

Puisque le filtre de Kalman est un algorithme discret, la discrétisation du modèle est nécessaire. Cette discrétisation a été faite en utilisant la méthode d'Euler directe qui fournit une approximation acceptable de la dynamique des systèmes pour une courte période d'échantillonnage.

Nous obtenons la représentation sous forme d'état discrète suivante :

$$\begin{cases}x_1(k+1)=x_1(k)+\Delta t.\ x_2(k)+w_1(k)\\ x_2(k+1)=x_2(k)+\Delta t\left[f_1(X,k)+g_1(X,k)\tau_1(k)+d_1(k)\right]+w_2(k)\\ x_3(k+1)=x_3(k)+\Delta t.\ x_4(k)+w_3(k)\\ x_4(k+1)=x_4(k)+\Delta t\left[f_2(X,k)+g_2(X,k)\tau_2(k)+d_2(k)\right]+w_4(k)\\ z_1(k)=x_1(k)+v_1(k)\\ z_2(k)=x_2(k)+v_2(k)\end{cases} \tag{V.33}$$

où $w(k)=[w_1(k)\ w_2(k)\ w_3(k)\ w_4(k)]$ et $v(k)=[v_1(k)\ v_2(k)]$ sont les vecteurs de bruits blancs Gaussiens de processus et de mesure avec des moyennes nulles. Δt est la période d'échantillonnage. k est l'instant discret.

EKF est implémenté comme représenté par les équations (III.57) à (III.59) où les matrices jacobéennes sont définies en annexe B.

EKF fournira le vecteur d'estimation d'état $\hat{X} = [\hat{\theta}_1, \hat{\dot{\theta}}_1, \hat{\theta}_2, \hat{\dot{\theta}}_2]^T = [\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4]^T$. Les conditions initiales d'état d'EKF sont choisies telles que $\hat{X}_{0/0} = [0,0,0,0]^T$.

Dans la simulation, la matrice de covariance d'erreur P est initialement définie comme une matrice unitaire de dimension 4×4, alors que les matrices de covariance de bruit Q et R ont des dimensions 4×4 et 2×2, respectivement, et sont supposés avoir la forme suivante:

$$Q = diag(q_{x_1}, q_{x_2}, q_{x_3}, q_{x_3}) = \begin{bmatrix} q_{x_1} & 0 & 0 & 0 \\ 0 & q_{x_2} & 0 & 0 \\ 0 & 0 & q_{x_3} & 0 \\ 0 & 0 & 0 & q_{x_4} \end{bmatrix} \tag{V.34}$$

$$R = diag(r_1, r_2) = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \tag{V.35}$$

Les valeurs de q_{x_i} et r_i dans les matrices Q et R sont généralement déterminées en utilisant une méthode d'essai-erreur. A des fins de comparaison, la performance de l'observateur EKF avec diverses compositions de Q et R est évaluée en utilisant l'erreur quadratique moyenne entre la position estimée et la position réelle comme suit:

$$MSE = \frac{1}{T}\sum_{k=1}^{N}\left[\theta_i(k) - \hat{\theta}_i(k)\right]^2,\ i = 1,2 \tag{V.36}$$

Dans ce qui suit, l'algorithme proposé a été simulé sous un PC en utilisant l'environnement du logiciel MatLab. Notez que tous les codes sont écrits en langage MatLab, dans les fichiers M-files. Dans cette simulation, les paramètres nominaux du robot sont donnés par: $m_1 = m_2 = 1\text{Kg}$ $l_1 = l_2 = 0.5\text{m}$, $l_{c1} = l_{c2} = 0.25\text{m}$, $I_1 = I_2 = 0.1\text{Kg.m}^2$, $g = 0.81\text{m/s}^2$.

Un total de $N = 4000$ données de mesure sont simulées sur un intervalle de temps de 0 à 4 secondes avec un pas d'échantillonnage $\Delta t = 0.001\text{s}$.

Les trajectoires de référence désirées pour x_1 et x_3 sont choisis $x_{1d}(t) = 70^\circ$ and $x_{3d}(t) = 90^\circ$.

Les conditions initiales choisies sont : $x_1(0) = 0$, $x_2(0) = 0$, $x_3(0) = 0$ et $x_4(0) = 0$.

Trois types d'incertitudes sont injectés dans la structure pour vérifier la robustesse du contrôleur: (1) Les incertitudes des paramètres ($+10\%$ sur les valeurs des paramètres nominaux du modèle). (2) Les perturbations externes sont choisies aléatoirement comme suit: $\delta_1 = \delta_2 = 1\times rand$, et elles ont appliqué à $t > 2\text{s}$, la limite supérieure des perturbations est supposée être $D=1$. (3) Bruits gaussiens aléatoires pour les états et les mesures avec des valeurs moyennes nulles et avec les covariances $q=10^{-2}$ et $r=10^{-5}$, respectivement.

Dans un premier temps, on va simuler le système complet sous la commande linéarisante traditionnelle afin de montrer son inconvénient en présence d'incertitudes paramétriques et de perturbations externes. En appliquant la loi de commande (V.12), nous avons choisi $k_{feed}=[200\,I_{2\times2};50\,I_{2\times2}]$ où $I_{i\times i}$ est une matrice d'identité de démontions $i\times i$.

Tableau V.1. Performances d'EKF pour un robot à 2-articulations par l'estimation essai-erreur

Cas	entrées ***Q*** et ***R***						**MSE_EKF [deg]**	**Qualité d'estimation**
	q_{x_1}	q_{x_2}	q_{x_3}	q_{x_4}	r_1	r_2		
1	1	1	1	1	1	1	1.6131	Mauvaise
2	0.2	10^{-1}	0.2	10^{-1}	10^{-1}	1	3.3692×10^{-5}	Bien
3	10^{-1}	10^{-1}	0.2	0.2	1	10^{-1}	3.2587×10^{-5}	Bien
4	10^{-1}	10^{-1}	10^{-1}	10^{-1}	1	1	3.1821×10^{-6}	Très bien

Le tableau V.1 montre les performances typiques d'EKF avec leurs matrices de covariance correspondantes (avec les entrées $q_{x_1}, q_{x_2}, q_{x_3}, q_{x_4}, r_1$ et r_2) obtenues par la méthode essai-erreur. On constate que de bons résultats de performance d'estimation ont été obtenus lorsque Q et R sont égaux (cas 2 et 3 dans le tableau V.1), mais une mauvaise sélection de ($q_{x_1}, q_{x_2}, q_{x_3}, q_{x_4}, r_1$ et r_2) peut produire une mauvaise performance d'estimation (cas 1). Notez que la meilleure performance d'estimation est obtenue avec les matrices Q et R ($q_{x_1}=q_{x_2}=q_{x_3}=q_{x_4}=10^{-1}$ and $r_1=r_2=1$) (cas 4) qui correspondent au la plus petit MSE. Les résultats de simulation par rapport au meilleur cas (cas 4) sont montrés sur les figures V.3 V.3(a) et (b), où nous présentons respectivement, la position de l'articulation-1 et celle de l'articulation-2.

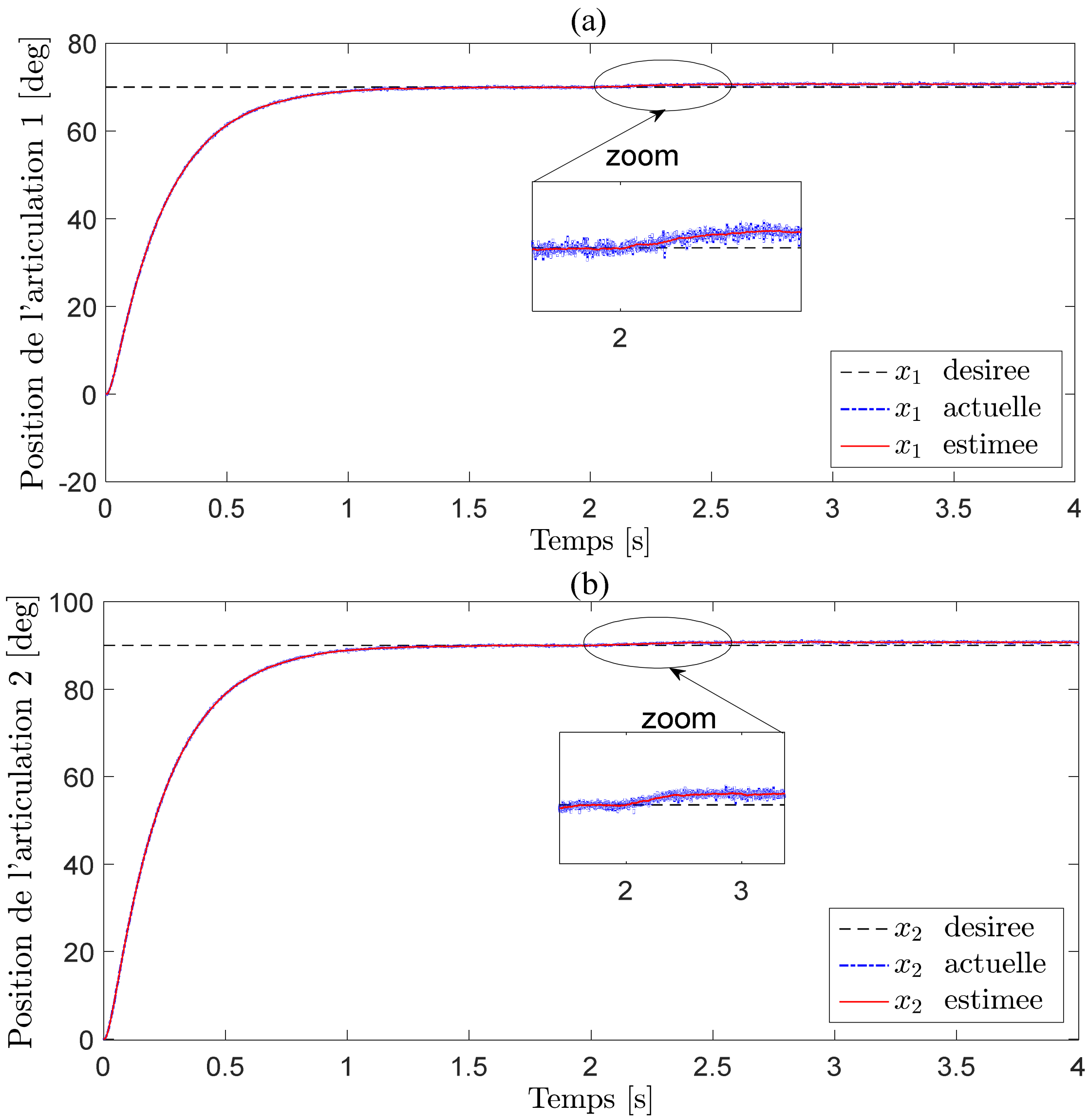


Fig. V.3. Positions des angles articulaires en utilisant la commande linéarisante classique avec EKF

Tableau V.2. Performances du SMO pour un robot à 2-articulations par l'estimation essai-erreur

Cas	gains du SMO		MSE_SMO [deg]	Qualité d'estimation
	γ_1	γ_2		
1	10^{-5}	10^{-3}	1.3306	Mauvaise
2	1	1	7.1961×10^{-5}	Bien
3	2	10	4.3120×10^{-5}	Bien
4	30	70	3.2612×10^{-6}	Très bien

Pour comparer les performances d'EKF avec d'autres observateurs en termes du MSE, nous avons donné dans le tableau V.2 les performances relatives à l'observateur mode glissant

[75]. En comparant les tableaux V.1 et V.2 (voir la colonne MSE), nous voyons clairement que, les deux méthodes ont donné de petits MSE, mais l'MSE obtenu par EKF est plus petit que celui obtenu par l'observateur SMO.

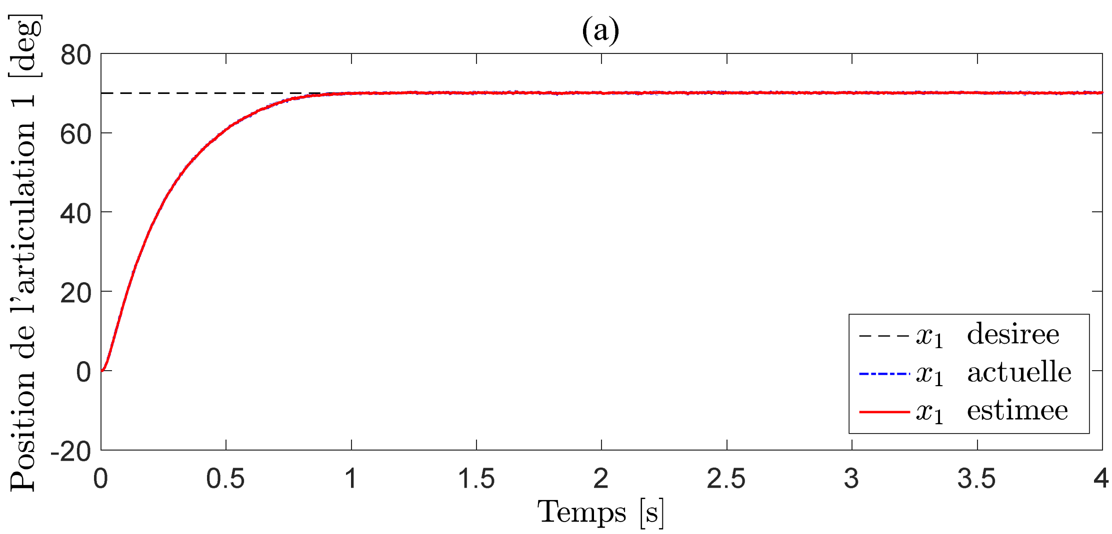
La figure V.3 montre que la commande linéarisante traditionnelle n'assure pas la robustesse en performance surtout après l'instant $t = 2\,s$ auquel les perturbations ont été appliquées ($+10\%$ variation des paramètres et perturbation externe). Nous constatons aussi que, la précision de prédiction d'EKF n'est pas tout à fait satisfaisante en raison du choix des entrées des matrices Q et R par la méthode essai-erreur.

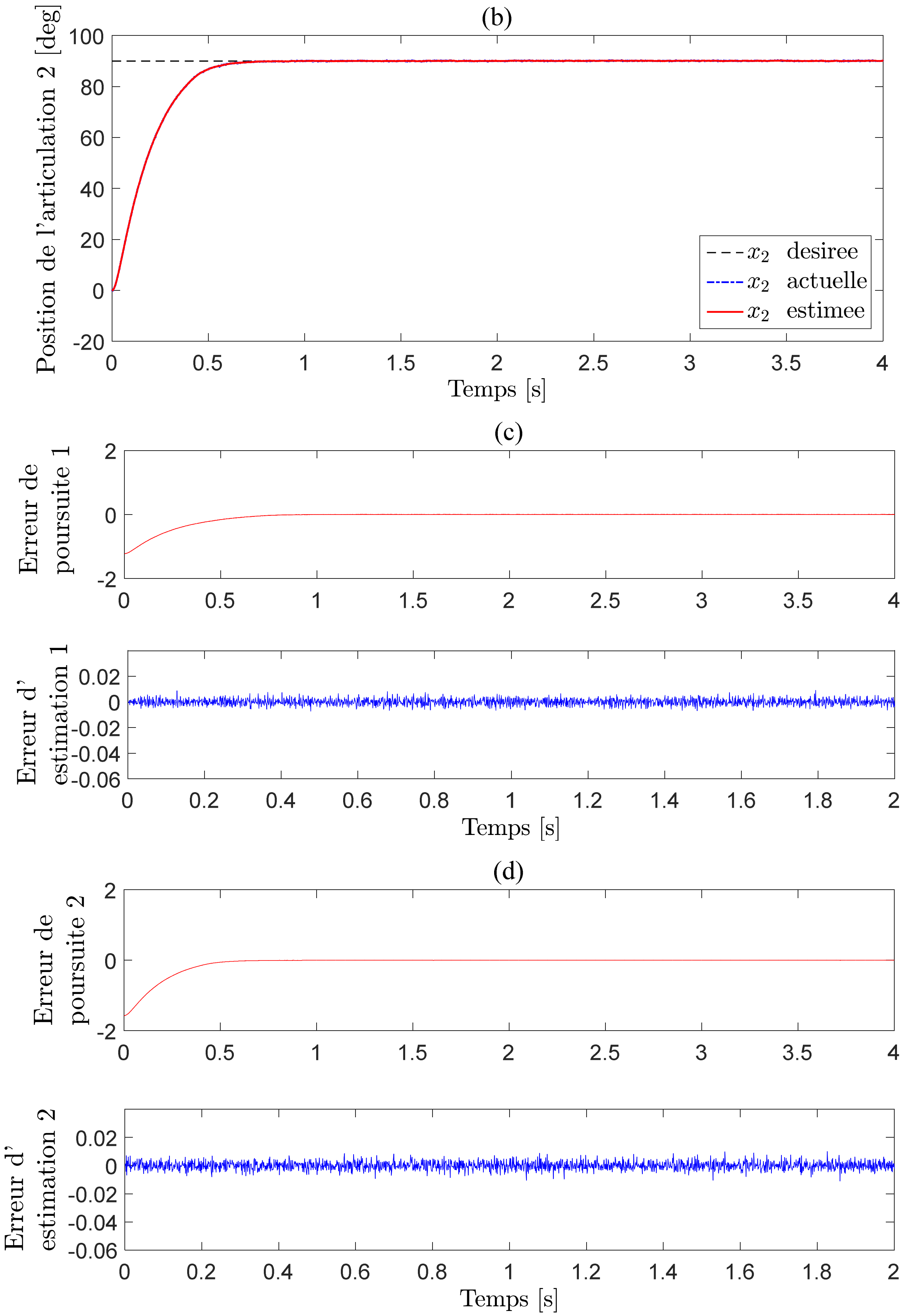
Dans le reste de cette section, nous proposons notre approche à appliquer afin de résoudre les deux problèmes ci-dessus.

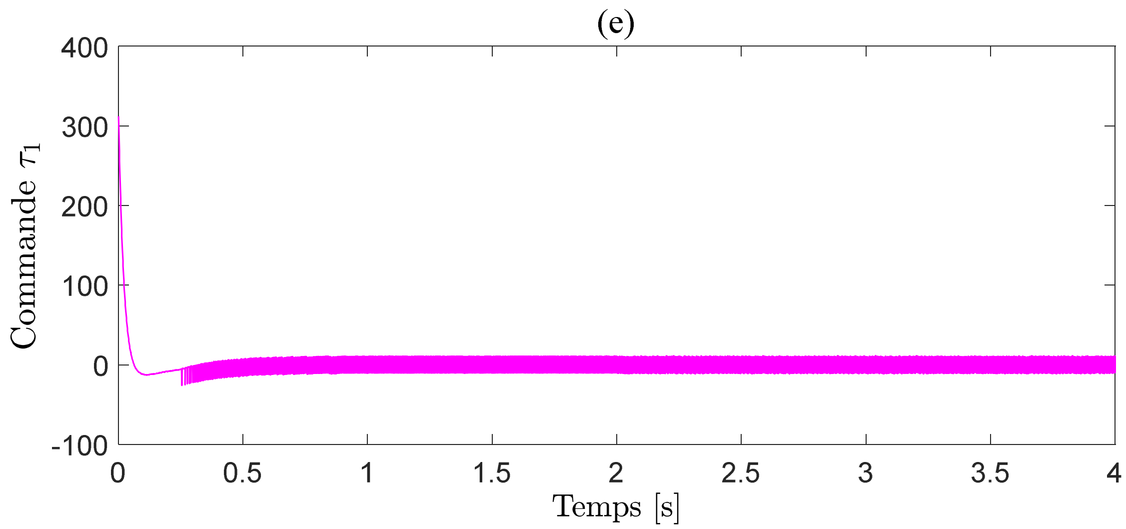
V.5.1. Problème de robustesse

Pour résoudre le problème de la robustesse et acquérir une meilleure réponse à ce système, la loi de commande donnée par (V.14) est utilisée dans laquelle la commande linéarisante classique a été augmentée par la commande discontinue. Dans ce cas, nous avons choisi $K = 10 I_{2\times2}$ et $\lambda = 5 I_{2\times2}$.

Les figures V.4(a) à (f) exposent les résultats de simulation issus de cette application.




(b)
Position de l'articulation 2 [deg]
Temps [s]
x_2 desiree
x_2 actuelle
x_2 estimee
(c)
Erreur de poursuite 1
Erreur d' estimation 1
Temps [s]
(d)
Erreur de poursuite 2
Erreur d' estimation 2
Temps [s]



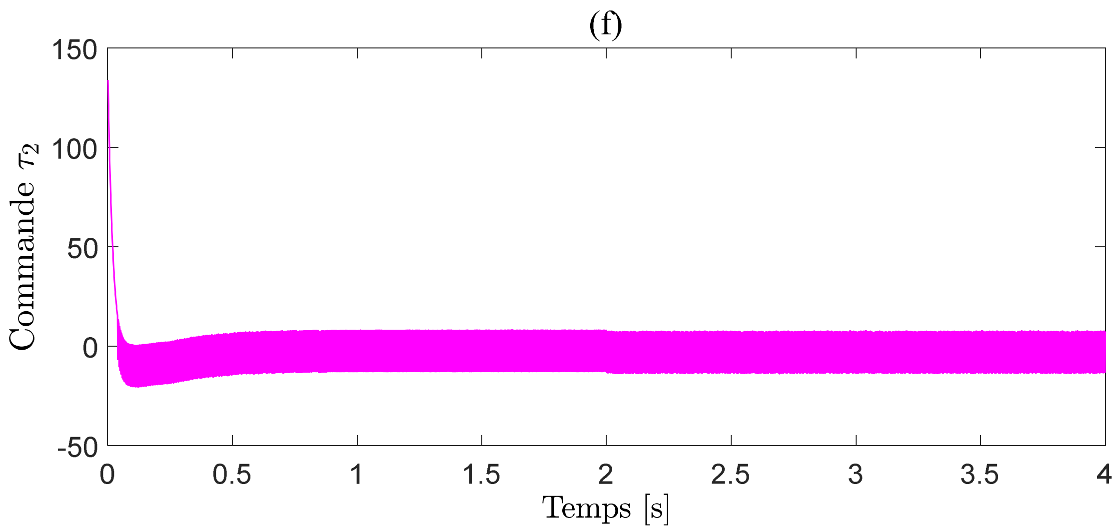


Fig. V.4. Résultats de simulation utilisant la commande linéarisante améliorée par le terme discontinu (a) Position de l'angle d'articulation-1 (b) Position de l'angle d'articulation-2 (c) Erreurs de poursuite et d'estimation en position-1 (c) Erreurs de poursuite et d'estimation en position-2 (e) Commande appliquée à l'articulation-1 (f) Commande appliquée à l'articulation-2.

Les figures V.4(a) et (b) montrent que les sorties du système rejoint rapidement ses références en utilisant la commande linéarisante augmentée par une composante de commande discontinue, même en présence des variations paramétriques et des perturbations externes. Ainsi, le système maintient un temps de réponse moins de $t_r = 1s$. Les figures V.4(c) et (d) montres que les erreurs de poursuite convergent toutes vers zéro après un court temps. Comme nous l'avons vu, la performance sous l'occurrence de variations paramétriques et de perturbations externes sont satisfaisantes. Contrairement aux réponses de sorties présentées du système dans les figures V.3(a) et (b) qui montrent que la performance n'est pas satisfaisante, surtout après $t = 2s$ lorsque les perturbations s'appliquent. Sur la base de ces résultats, on peut voir que, la commande proposée est capable de forcer les sorties du système à converger vers ses références. Les Figure V.4(e) et (f) montrent l'évolution des entrées de commande appliquées au robot, où on constate que les signaux présentent des discontinuités. Donc, la performance de la commande n'est pas satisfaisante en raison du phénomène de broutement provoqué par la sélection inappropriée du gain de commutation K .

Afin de résoudre ce problème de broutement, la propriété de lissage de la logique floue type-1 sera exploitée comme nous l'avons vu dans le chapitre I en utilisant (V.17). Les fonctions d'appartenance de s et de K_{fuzzy} sont choisies comme illustré sur la figure V.5(a) et (b) respectivement, dans laquelle les variables linguistiques suivantes ont été utilisées: **négatif (N)**, **zéro (Z)**, **positif (P)**, **positif petit (PP)** et **positif grand (PG)**.

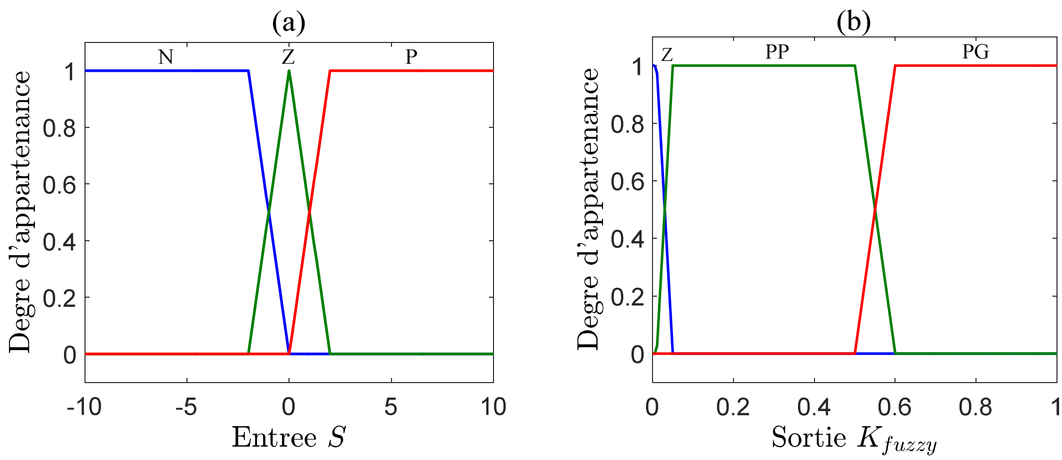


Fig. V.5. (a) Fonctions d'appartenance d'entrée (b) Fonctions d'appartenance de sortie

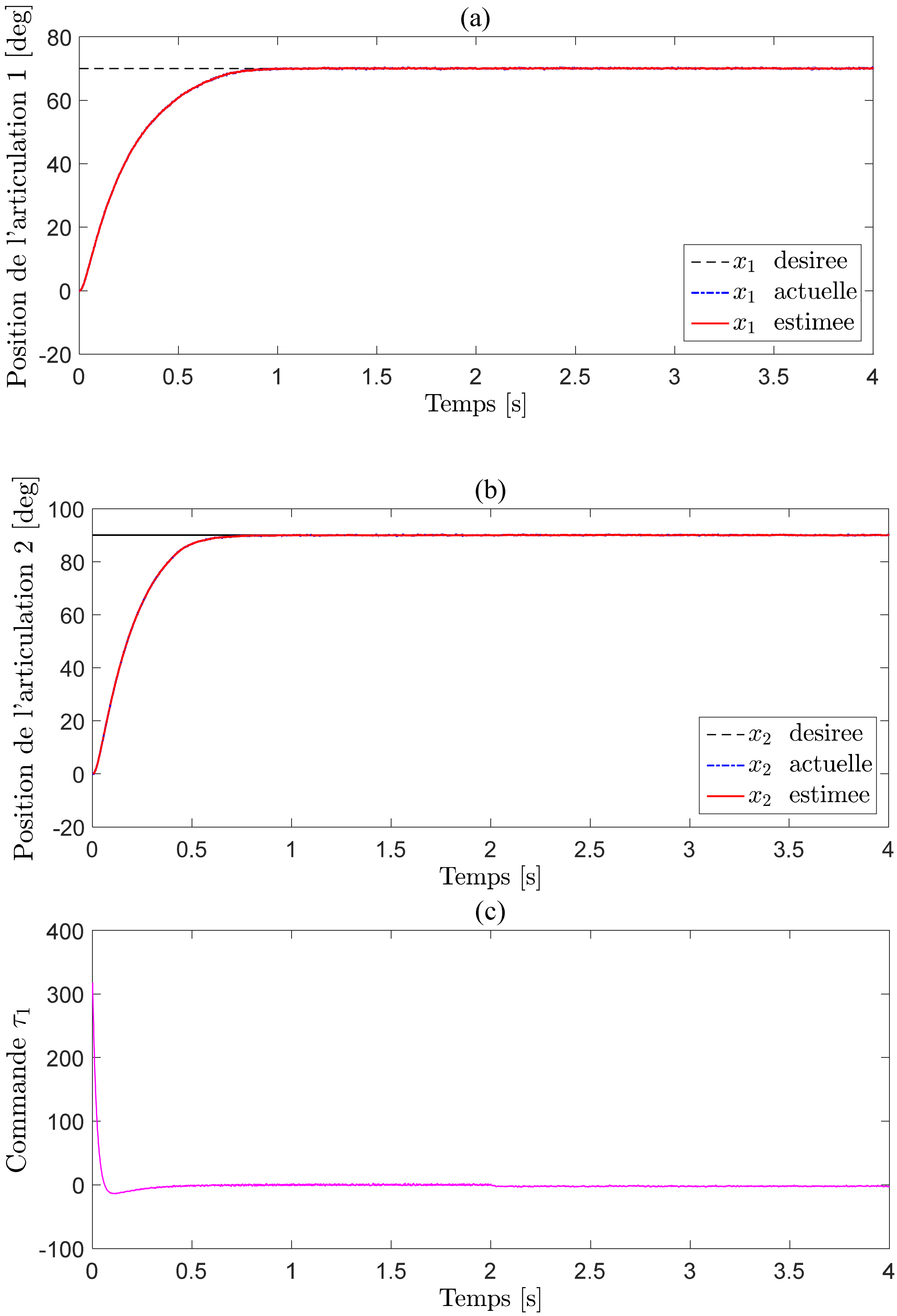
L'ensemble de règles floues adoptées contient 3 règles définies comme suit:

Règle 1 : **Si** s est N , **Alors** K_{fuzzy} est PG

Règle 2 : **Si** s est Z , **Alors** K_{fuzzy} est Z

Règle 3 : **Si** s est P , **Alors** K_{fuzzy} est PP

Ces règles gouvernent la relation entrée-sortie entre s et K_{fuzzy} en adoptant le moteur d'inférence de type Mamdani, dans lequel la méthode du centre de gravité est utilisée pour la défuzzification.


(a)
Position de l'articulation 1 [deg]
Temps [s]
x_1 desiree
x_1 actuelle
x_1 estimee
(b)
Position de l'articulation 2 [deg]
Temps [s]
x_2 desiree
x_2 actuelle
x_2 estimee
(c)
Commande τ_1
Temps [s]

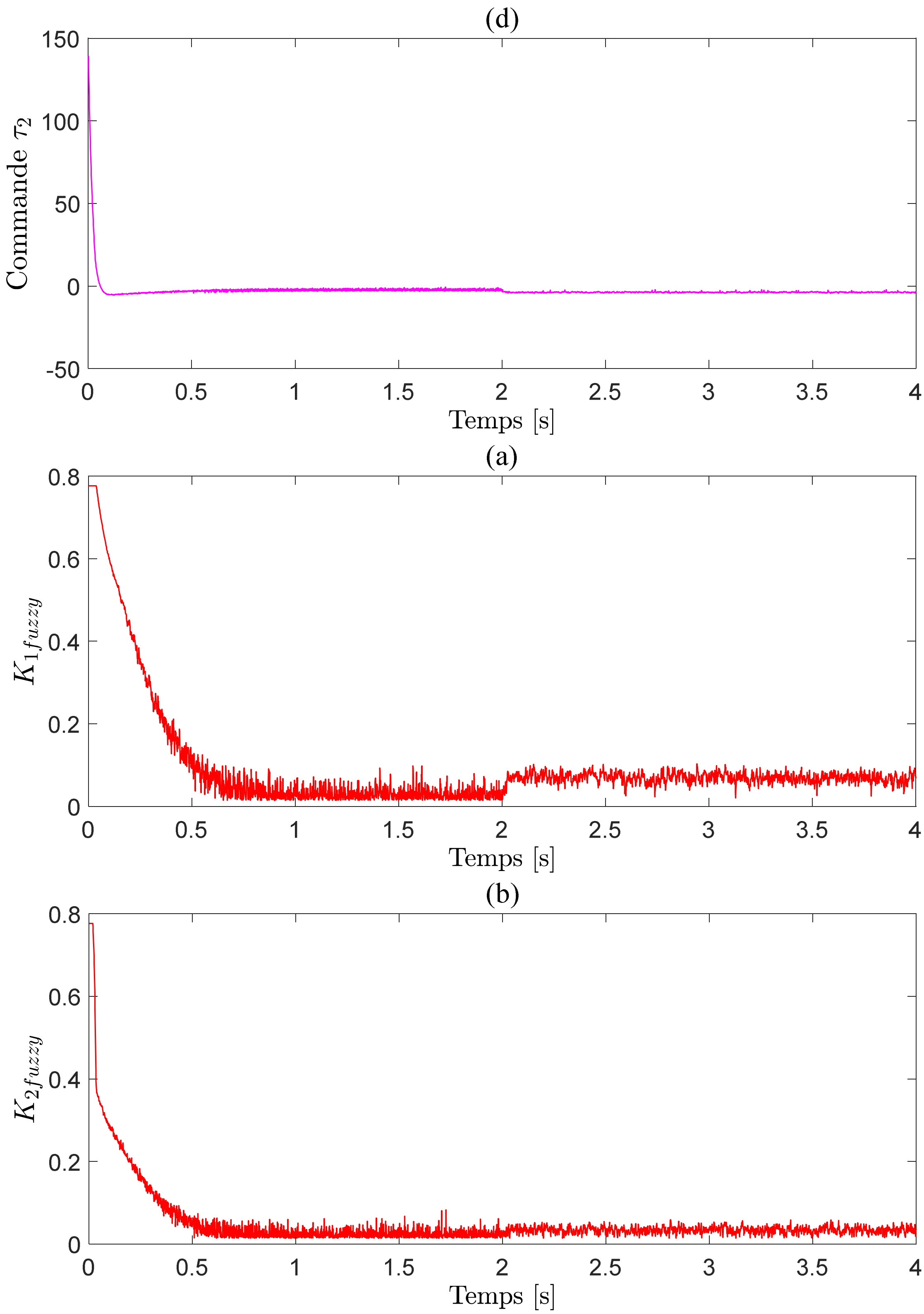


Fig. V.6. Résultats de simulation utilisant la commande linéarisante améliorée par la commande discontinu floue type-1 (a) Position de l'angle d'articulation-1 (b) Position de l'angle d'articulation-2 (c) Entrée de commande appliquée à l’articulation-1 (d) Entrée de commande appliquée à l’articulation-2 (e) Evolution temporelle du gain flou $K_{1\ fuzzy}$ (f) Evolution temporelle du gain flou $K_{2\ fuzzy}$

La figure V.6 est obtenue par simulation en utilisant la loi de commande (V.18), où nous avons présenté sur les sous figures V.6(a) et (b) les positions des angles articulaires. En comparant les nouvelles entrées de commande associées présentées dans les sous figures V.6(c) et (d) avec les précédentes dans les figures V.4(e) et (f), on peut voir clairement que le phénomène de broutement est presque disparu, on constate que les amplitudes de discontinuités sont presque disparues. Les gains flous estimés sont représentés sur les sous figures V.6(e) et (f).

V.5.2. Problème de prédiction

Notez que dans toutes les simulations ci-dessus, les matrices de covariance ont été ajustées en utilisant la méthode essai-erreur qui est simple à réaliser mais consomme plus de temps. Pour obtenir de performance plus satisfaisante, l'ajustement a sera effectué par l'algorithme PSO et sera comparer à la technique des algorithmes génétiques tels que décrits dans la section II.3.

V.5.2.1. Méthode PSO-EKF

Nous suggérons de rechercher simultanément, la combinaison optimale de six variances $q_{x_1}, q_{x_2}, q_{x_3}, q_{x_4}, r_1$ et r_2 en utilisant l'algorithme II.2 (voir chapitre II) pour trouver les matrices de covariance optimales Q et R d'EKF, ce qui permettra d'obtenir de meilleures estimations avec une plus grande précision que la méthode essai-erreur.

Dans cette partie, nous avons exécuté l'algorithme d'optimisation PSO de manière hors-ligne avec EKF afin de trouver efficacement les matrices de covariances optimales Q et R. Le critère a été utilisé comme fonction objectif pour évaluer la qualité de prédiction est l'erreur quadratique moyenne (V.36) entre la sortie réelle et la sortie estimée en fonction d'un certain nombre d'itérations à effectuer pour chaque étape d'estimation.

Après avoir exécuté le PSO-EKF, les matrices de covariance optimisées Q et R et leurs performances MSE sont illustrées dans le tableau V.3. Il convient de noter que la convergence de la méthode PSO vers la solution optimale dépend des paramètres c_1, c_2 et w dans lesquels le coefficient d'auto-reconnaissance $c_1 = 15$, le coefficient social $c_2 = 2$ et le poids d'inertie varie entre $w = 0.3$ à 1. Puisque nous avons six paramètres à optimiser, la dimension de la simulation sera de 6. Notez également que l'essaim simulé a une taille de 20 avec un nombre maximal de générations égal à 100.

Tableau V.3. Performances d'EKF optimisé à l'aide du PSO

Taille de l'essaim	Itérations	Eléments de *Q* et *R*						MSE_PSO [deg]
		q_{x_1}	q_{x_2}	q_{x_3}	q_{x_4}	r_1	r_2	
20	5	10^{-4}	0.0316	10^{-4}	0.0534	0.0685	0.0745	1.5401×10^{-6}
	10	10^{-4}	0.0432	10^{-4}	0.0416	0.0515	0.0621	1.3712×10^{-6}
	50	10^{-4}	0.0005	10^{-3}	0.0001	0.0400	0.0601	1.2049×10^{-6}
	100	10^{-5}	10^{-5}	10^{-5}	10^{-5}	0.4800	0.8637	1.1637×10^{-6}

Le tableau V.3 illustre la convergence de l'algorithme PSO-EKF, où l'MSE est diminué à 1.1637×10^{-6} après 100 itérations, ce qui est inférieur à la valeur obtenue par la méthode essai-erreur ($MSE_{essai\text{-}erreur}$ $=3.1821\times10^{-6}$) ce qui confirme le l'efficacité de cette méthode.

V.5.2.2. Méthode des algorithmes génétiques

À des fins de comparaison, nous allons présenter dans ce qui suit l'optimisation d'EKF en utilisant les GAs. Notez que nous avons utilisé l'algorithme génétique avec les paramètres suivants: taille de la population = 20, nombre maximal de génération = 100, dimension = 6, probabilité de croisement = 0,8 et probabilité de mutation = 0,01.

Les matrices de covariance optimisées en utilisant les algorithmes GA sont présentées dans le tableau V.4 où nous avons vu que la valeur de l'MSE est diminuée à 2.4959×10^{-6} après 100 itérations. Notons que cette MSE est proche de celui obtenu par la méthode essai-erreur qui est égale à 3.1821×10^{-6}.

Tableau V.4. Performances d'EKF optimisé à l'aide du GA

Taille de l'essaim	Itérations	Eléments de *Q* et *R*						MSE_GAs [deg]
		q_{x_1}	q_{x_2}	q_{x_3}	q_{x_4}	r_1	r_2	
20	5	0.0153	0.0416	10^{-4}	0.0456	0.0772	0.0753	7.4531×10^{-6}
	10	0.0106	0.0112	10^{-4}	0.0324	0.0456	0.0568	6.2235×10^{-6}
	50	0.0081	0.0153	0.001	0.0248	0.0440	0.0654	2.7500×10^{-6}
	100	0.0010	0.1000	10^{-4}	0.0010	0.5000	0.4001	2.4959×10^{-6}

A partir des résultats obtenus présentés par les deux méthodes dans les tableaux V.3 et V.4, si on s'intéresse uniquement à la minimisation de la fonction objectif qui est l'erreur quadratique moyenne, la comparaison des approches PSO-EKF et GAs montre que les deux méthodes sont capables de trouver les matrices de covariance optimales *Q* et *R*. On observe que PSO-EKF donne des résultats plus précis que ceux obtenus par l'approche GAs lorsque le nombre d'itérations (génération) est élevé. Par conséquent, il peut être confirmé que PSO-EKF est plus performant que l'approche GAs. Notons que la comparaison a été faite dans les mêmes conditions (nombre de génération, taille de la population, population initiale). De plus, la figure V.7 montre l'évolution de la fonction objectif pour les méthodes PSO et GA, où l'on remarque que la convergence du PSO est plus rapide que la convergence de GA.

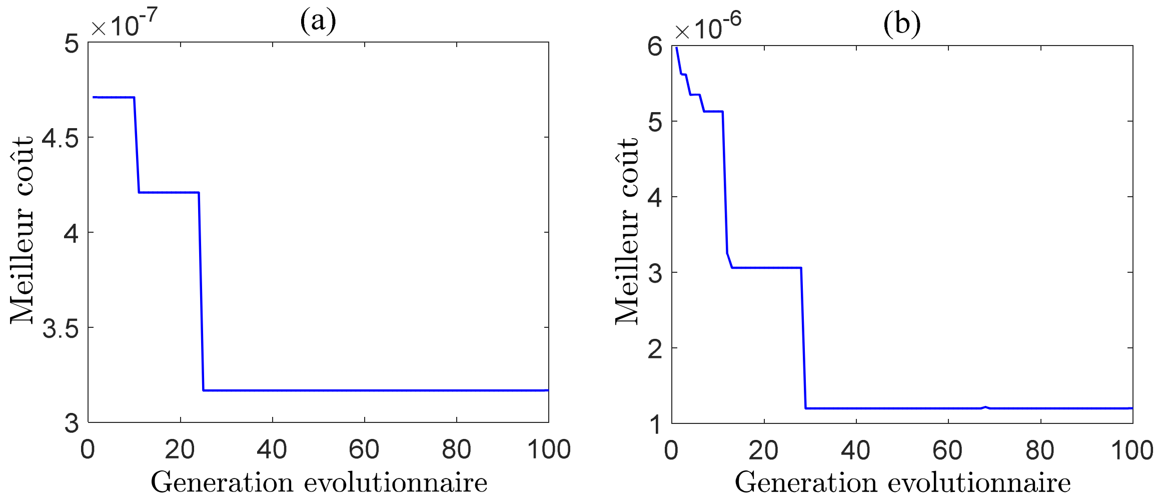
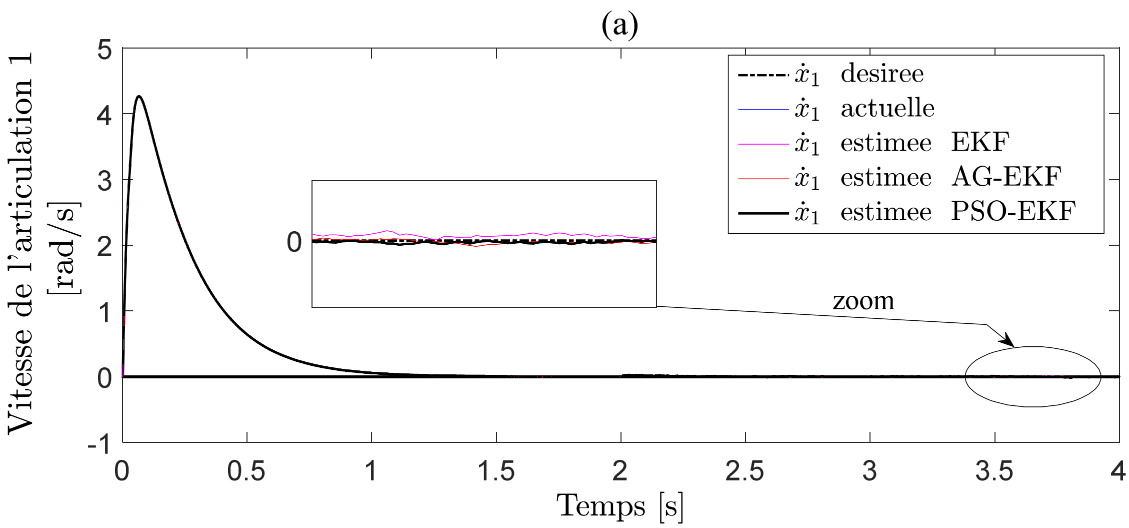
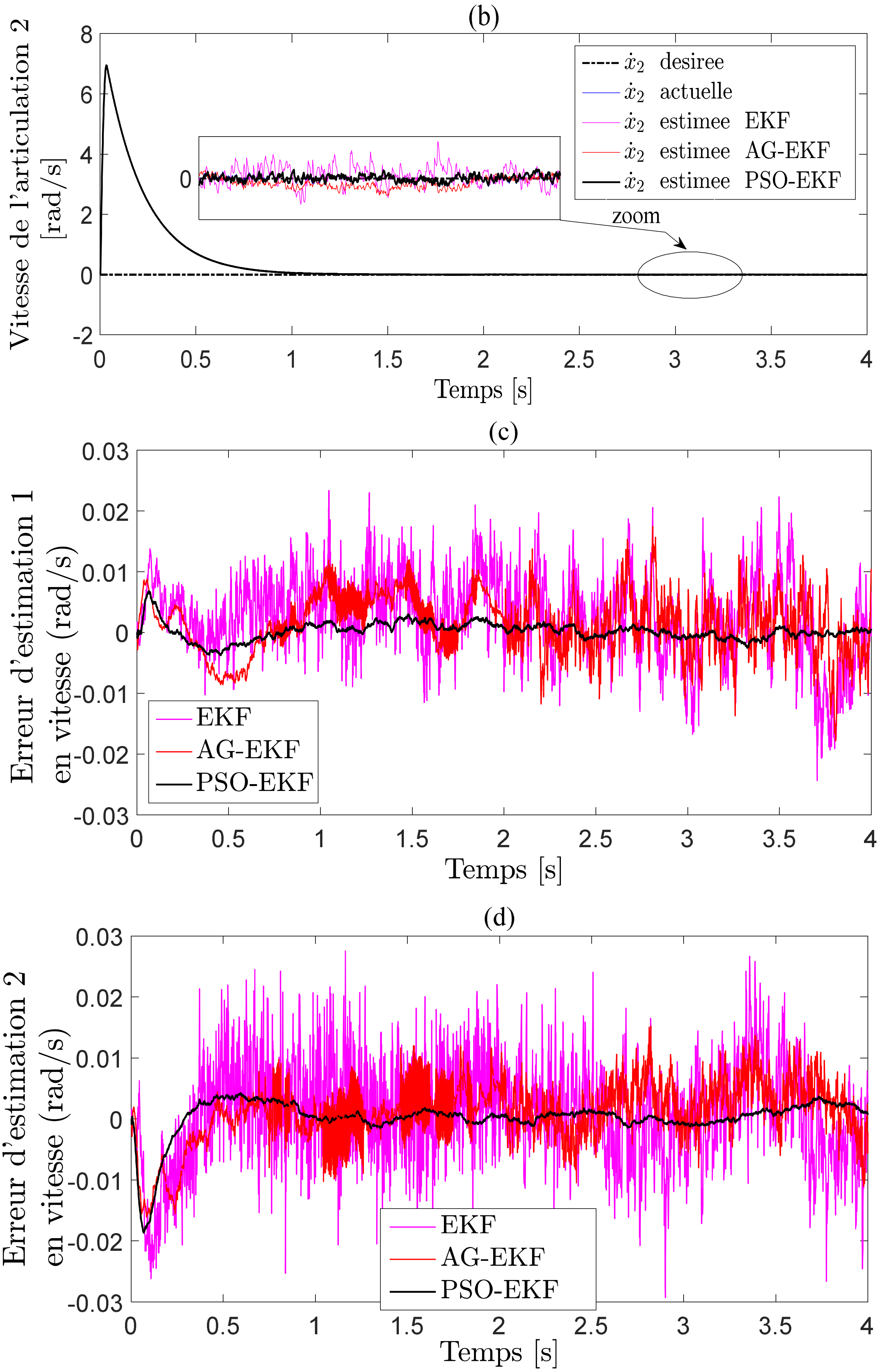


Fig. V.7. Evolution de la fonction fitness / objectif en 100 itérations.
(a) PSO par rapport au Tab. 3, (b) GA par rapport au Tab. 4

Dans ce qui suit, nous présentons les résultats de simulation finaux relatifs à la commande linéarisante augmentée par la commande discontinue floue type-1, avec les meilleurs paramètres optimaux d'EKF (voir la figure V.8).




(b)
Vitesse de l'articulation 2 [rad/s]
$\dot{x}_2$ desiree
$\dot{x}_2$ actuelle
$\dot{x}_2$ estimee EKF
$\dot{x}_2$ estimee AG-EKF
$\dot{x}_2$ estimee PSO-EKF
zoom
Temps [s]
(c)
Erreur d'estimation 1 en vitesse (rad/s)
EKF
AG-EKF
PSO-EKF
Temps [s]
(d)
Erreur d'estimation 2 en vitesse (rad/s)
EKF
AG-EKF
PSO-EKF
Temps [s]

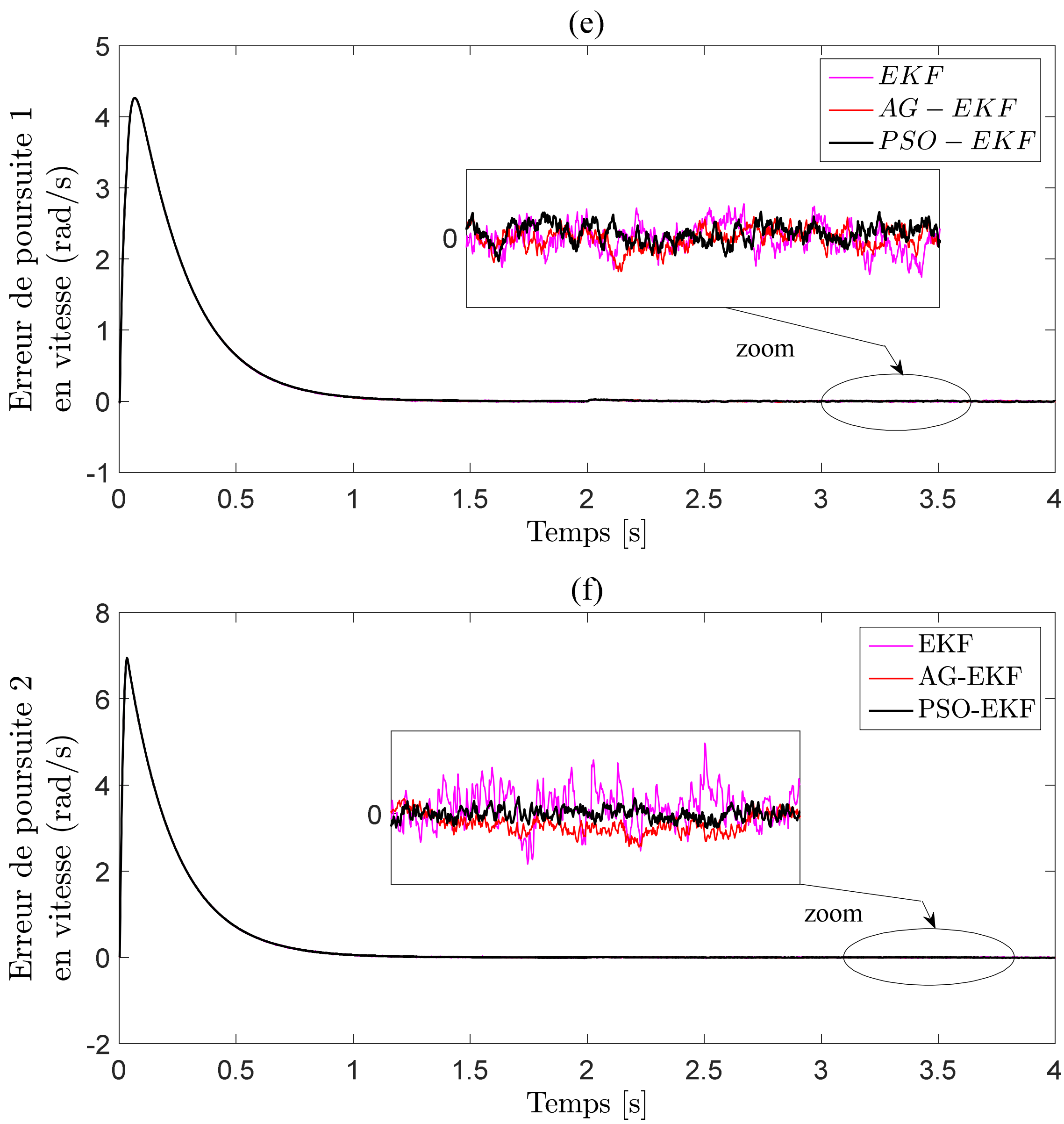


Fig. V.8. Résultats de simulation en utilisant la commande proposée pour différents algorithmes d'optimisation (a) Vitesse de l'angle d'articulation-1, (b) Vitesse de l'angle d'articulation-2, (c) Erreurs d'estimation en vitesse pour l'articulation-1, (d) Erreurs d'estimation en vitesse pour l'articulation-2, (e) Erreurs de poursuite en vitesse pour l'articulation-1, (f) Erreurs de poursuite en vitesse pour l'articulation-2.

Dans les figures V.8(a) et (b), nous avons présenté respectivement, les réponses des vitesses des angles d'articulations 1 et 2 obtenues avec les valeurs optimales des matrices de covariance d'EKF données dans les tableaux V.1, V.3 et V.4 pour la méthode essai-erreur, PSO et GA, respectivement. Les erreurs d'estimation en vitesse correspondantes sont présentées sur les figures V.8(c) et (d), respectivement. De plus, les erreurs de poursuite en vitesse correspondantes sont présentées dans les figures V.8(e) et (f), respectivement. D'après les figures V.8(a) et (b), on peut voir une bonne performance de poursuite de trajectoires en vitesse. Les états $\dot{x}_1(t)$ et $\dot{x}_2(t)$ convergent rapidement vers leurs références. Cette constatation est renforcée par les figures V.8(c) - (f), qui représentent la convergence des erreurs de poursuite et d'estimation vers zéro. On voit bien que, les meilleurs résultats sont obtenus par la commande proposée avec PSO-EKF où l'on peut voir que PSO-EKF s'adapte aux bonnes variables d'état avec une grande précision pour un robot manipulateur à deux articulations, et cela montre l'efficacité de l'approche proposée. Dans toutes ces figures, les résultats de simulation obtenus sont très satisfaisants et montrent l'intérêt de la structure proposée.

V.6. Conclusion

Dans ce chapitre, nous avons proposé une loi de commande pour une classe de systèmes non-linéaires. La loi de commande proposée consiste en une loi de commande linéarisante classique, augmentée d'une composante de robustification par mode de glissement. La composante de robustification a été introduite pour assurer la robustesse de la structure de commande, par rapport aux effets des variations paramétriques et les perturbations externes. La combinaison de la commande linéarisante avec la composante de robustification, montre une meilleure performance. La composante de robustification a été renforcée également par un système flou type-1 pour réduire le phénomène de broutement. La structure de base d'un système flou type-1 a été constituée essentiellement d'un fuzzificateur, d'une base de règles, d'un moteur d'inférence floue et d'un defuzzificateur.

Nous avons supposé que tous les états ne sont pas mesurés. Par conséquent, un EKF a été introduit pour estimer ces états. La performance d'estimation obtenue avec l'observateur EKF est également comparée à celle obtenue par l'observateur SMO. L'ajustement des paramètres de l'estimateur EKF a été fait par l'ajustement des matrices de covariance Q et R via l'algorithme d'optimisation PSO, la performance obtenue a été comparée avec la technique GA.

La stabilité de l'approche proposée était garantie par le critère de stabilité de Lyapunov. Des simulations ont été alors présentées pour illustrer d'une part la robustesse de la commande proposée et d'autre part les performances d'observateur optimisé, pour les systèmes non-linéaires en présence des variations paramétriques et les perturbations externes simultanément.

Chapitre VI

Commande floue type-2 intervalle par mode glissant avec observateur optimisé : Application au bras manipulateur

Sommaire

Commande floue type-2 intervalle par mode glissant avec observateur optimisé : Application au bras manipulateur

Mesure ce qui est mesurable, et rend mesurable ce qui ne peut être mesuré ...

Galilée.

VI.1. Introduction

L'un des meilleures commandes non-linéaires classiques est la commande par mode glissant (SMC), basé sur la théorie des systèmes à structures variables; elle a été introduite par Uktin en 1977. Cette commande appelée en anglais sliding mode control (SMC) est connu pour sa robustesse vis-à-vis les incertitudes et les perturbations externes. Au cours des deux dernières décennies, les systèmes à logique floue (FLS) ont été un sujet dominant dans les systèmes de contrôle intelligents. De nombreux schémas FLS ont été développés pour gérer les systèmes non-linéaires afin d'améliorer les performances de l'SMC, e.g., Kapoor et Ohri [166], ont proposé une commande par mode glissant flou avec stabilisation globale; utilisant une fonction de saturation pour le contrôle de trajectoire; ils ont utilisé des systèmes flous pour ajuster le gain de commutation du SMC, et la fonction de saturation linéaire d'une couche limite pour résoudre le problème de broutement (vibrations à haute fréquence). Soltanpour *et al.* [167], ont combiné la commande linéarisante avec la commande mode glissant floue en utilisant le modèle flou de Takagi-Sugeno, et le résultat obtenu était exempt du phénomène de broutement indésirable. De plus, Baklouti *et al.* [168], ont combiné l'SMC, le PI et les systèmes flous adaptatifs pour commander un bras manipulateur d'un robot; Le système flou a été utilisé pour approximer les fonctions non-linéaires inconnues et l'action PI é été utilisée pour réduire le phénomène de broutement. Cependant, Naoual *et al.* [169], ont conçu un contrôleur par mode glissant flou pour commander un bras manipulateur, dans lequel la logique floue a été utilisée pour approximer seulement les parties dynamiques inconnues du système. Chen *et al.* [170], ont utilisé une technique d'approximation de fonction pour commander un moteur DC, cette technique a été utilisée pour transformer le terme incertain en combinaisons linéaires finie de la fonction de base orthogonale. Notez que dans tous ces travaux cités, le problème d'observation n'était pas pris en compte. Pour cette raison nous avons dans [171] conçu un contrôleur par mode glissant flou type-1 basé sur un EKF; pour commander un bras manipulateur à deux articulations, dans lequel la logique floue type-1 a été utilisée pour approximer le gain de commutation du SMC.

La logique floue type-2 est une généralisation de la logique floue type-1 (logique floue conventionnelle), dans laquelle les valeurs des fonctions d'appartenance sont elles-mêmes floues [37][172]. Le type de système flou type-2 le plus couramment utilisé est le système flou type-2 intervalle (IT2FLS), qui utilise des degrés d'appartenance de formes intervalles [173]. Les contrôleurs basés sur IT2FLS sont capables de maintenir les performances en présence de bruit très élevé et de fortes non-linéarités [173][174][175]. Par conséquent, l'intégration de l’IT2FLS dans la commande SMC conventionnelle, peut être réalisée avec un contrôleur de poursuite intelligent hybride, avec une robustesse vis-à-vis du bruit de mesure. Bien que l’SMC fonctionne bien dans les systèmes non-linéaires, il présente un inconvénient majeur, le phénomène de broutement, qui est causé par une sélection inappropriée du gain de commutation. Cependant, plusieurs recherches ont été consacrées à l’étude pour éviter ce problème [176][177][178][179].

La commande non-linéaire robuste d'un système donné nécessite la connaissance des variables d'état, qui sont rarement disponibles pour la mesure directe. Dans la plupart des cas, il existe un réel besoin d'états non mesurés estimés de manière fiable; l'élaboration d'une loi de commande d'un système donné nécessite souvent l'accès à la valeur d'un ou de plusieurs de ses états. Pour cette raison, il est nécessaire de concevoir un système dynamique auxiliaire; appelé observateur, qui est chargé d'estimer l'état du système à partir des entrées appliquées et des mesures fournies par les capteurs physiques. Plusieurs algorithmes sur ce sujet peuvent être trouvés dans la littérature [75][150][153][77]. L’EKF fournit un estimateur d'état optimal en raison de sa capacité à considérer les incertitudes stochastiques. La performance d'estimation est le problème majeur associé à EKF; il a forte influence surles paramètres du système et les matrices de covariance du bruit d'état et du bruit de mesure Q et R, respectivement. D’après la théorie du filtre de Kalman, Q et R doivent être obtenus en considérant les propriétés stochastiques des bruits correspondants [156], c'est pour ça que dans la plupart des cas, on considère que les matrices Q et R sont généralement inconnues.

Cependant, comme elles ne sont généralement pas connues précisément, l'amélioration de performance d'EKF peut être assimilée à un problème d'optimisation. Ces matrices ont été ajustées par les AGs [151] et PSO [87]. Notez que dans ces travaux, le problème traité était seulement l'estimation. Par ailleurs, dans notre papier [180] on a utilisé l'algorithme PSO pour ajuster ces matrices pour un contexte de commande et d'observation à la fois.

Ce chapitre présente, une commande par mode glissant floue type-2 (IT2FSMC), combiné avec un observateur EKF optimisé, en présence d'incertitudes et de perturbations pour les manipulateurs robotiques. La contribution principale est la proposition d'une nouvelle méthode basée sur une combinaison du filtre de Kalman étendu avec optimisation basée sur la biogéographie (BBO) [56] afin d'obtenir une haute performance d'estimation d'états du système à commandé. Les résultats sont comparés avec ceux obtenus par l'algorithme PSO. Les paramètres à optimiser sont les matrices de covariance du bruit d'état et du bruit de mesure, qui jouent un rôle important dans les performances de l'observateur EKF. Le système flou type-2 intervalle est utilisé pour réduire le phénomène de broutement dans la commande SMC. Le théorème de Lyapunov est utilisé pour prouver la stabilité du système en boucle fermée. Afin de tester la faisabilité du schéma de commande proposé et d'évaluer la performance en poursuite de trajectoire, une application à la commande d'un bras manipulateur est effectuée. Les résultats de simulation pour des testes de performance et de robustesse seront également présentées.

VI.2. Contexte et formulation

Considérons un robot manipulateur donné sous la forme standard suivante [165]:

$$M(\theta)\ddot{\theta}+C\left(\theta,\dot{\theta}\right)\dot{\theta}+G(\theta)=\tau \qquad \text{(VI.1)}$$

où $\theta,\dot{\theta},\ddot{\theta}\in\mathbb{R}^n$ sont respectivement, la position conjointe, la vitesse et l'accélération.

$M(\theta)=M_0(\theta)+\Delta M(\theta)\in\mathbb{R}^{n\times n}$ est la matrice d'inertie symétrique, définie positive.

$C\left(\theta,\dot{\theta}\right)=C_0\left(\theta,\dot{\theta}\right)+\Delta C\left(\theta,\dot{\theta}\right)\in\mathbb{R}^{n\times n}$ est le vecteur de Coriolis et les forces centrifuges.

$G(\theta)=G_0(\theta)+\Delta G(\theta)\in\mathbb{R}^n$ est le vecteur des forces gravitationnelles.

$\tau\in\mathbb{R}^n$ est le vecteur des couples de commande, appliqués aux actionneurs articulaires.

$M_0(\theta),C_0\left(\theta,\dot{\theta}\right),G_0(\theta)$ sont des termes nominaux, $\Delta M(\theta),\Delta C\left(\theta,\dot{\theta}\right),\Delta G(\theta)$ sont les paramètres d'incertitudes.

Le modèle dynamique d'un manipulateur robotique (VI.1) avec des incertitudes et des perturbations peut être réécrit comme suit:

$$M_0(\theta)\ddot{\theta} + C_0(\theta,\dot{\theta})\dot{\theta} + G_0(\theta) = \tau + d(\theta,\dot{\theta},\ddot{\theta}) \tag{VI.2}$$

où $d(\theta,\dot{\theta},\ddot{\theta}) = -\Delta M(\theta) - \Delta C(\theta,\dot{\theta}) - \Delta G(\theta) + \delta \in \mathbb{R}^n$ représente la somme des incertitudes paramétriques et des perturbations externes (δ).

Notre objectif est de concevoir une loi de commande bornée pour l'entrée τ, de sorte que tous les signaux soient bornés, et que les trajectoires d'états $X = (\theta_1, \theta_2)^T$ convergent vers les trajectoires désirées $X_d = (\theta_{1d}, \theta_{2d})^T$ le plus près possible, pour tous les intervalles de temps $T = [0, t_f]$ sous la présence d'incertitudes paramétriques et de perturbations externes.

Concernant le problème de contrôle non-linéaire, nous proposons d'utiliser l'approche de la commande par mode glissant [3][181].

Deux propriétés classiques des robots manipulateurs sont considérées :

Propriété 1 $M(\theta)$ est définie symétrique et positive, $M^T = M$.

Propriété 2 $(\dot{M} - 2C)$ est dissymétrique, c'est-à-dire pour tout vecteur X, on a

$$M^T(\dot{M} - 2C)M = 0$$

Pour atteindre l'objectif de la commande, les hypothèses suivantes sont retenues dans ce chapitre.

Hypothèses : En ce qui concerne le système dynamique présenté en (VI.2), nous adoptons les hypothèses de simplification et de réalisation suivantes :

- **Hyp VI.1:** Les positions et les vitesses des articulations ne sont pas disponibles.
- **Hyp VI.2:** Les perturbations d sont inconnues et supposées bornées, *i.e.* $\|d\| \le D$ où $D > 0$.
- **Hyp VI.3:** Les trajectoires désirées $\theta, \dot{\theta}$ et $\ddot{\theta}$ sont disponibles et avec des bornes connues.
- **Hyp VI.4:** Les valeurs admissibles de l'entrée de commande $\tau(t)$ sont limitées entre une borne supérieure et une borne inférieure $\underline{\tau}$ et $\overline{\tau}$, telles que $\underline{\tau} \le \tau(t) \le \overline{\tau}$.

Toutefois, la dynamique du système (VI.2) est non-linéaire, de plus les variables d'états ne sont généralement pas toutes disponibles. Une solution est d'utiliser un observateur non-linéaire pour estimer tous les états à partir des mesures disponibles. Dans cette section, nous proposons d'utiliser le filtre de Kalman étendu [86] qui peut fournir des estimations d'état optimales en raison de sa capacité à considérer les incertitudes stochastiques. Cet algorithme peut mettre à jour les états du système en ligne en utilisant les observations précédentes et actuelles en utilisant les équations décrites par (III.57) à (III.59) qui sont mentionné dans la section III.7.2.

Ainsi, notre objectif est de déterminer la commande $\tau(\hat{X})$ qui permet de conduire l'erreur de poursuite estimée $\hat{e}(t)$ définie par :

$$\hat{e} = \hat{\theta} - \theta_d \tag{VI.3}$$

à converger asymptotiquement vers zéro et garantir la stabilité du robot, où $\hat{X}$ est l'estimé de X , $\hat{\theta}$ est la position estimée et θ_d est la position désirée.

Comme nous avons mentionné précédemment dans le chapitre V, le filtre de Kalman se caractérise principalement par deux matrices Q et R qui contiennent les valeurs des covariances des bruits d'état et de mesure respectivement, et lorsque les valeurs de ces matrices ne sont pas connues précisément, l'amélioration de performance de l'estimation d'EKF peut être assimilée à un problème d'optimisation. Pour résoudre ce problème, nous avons proposé précédemment (dans le chapitre V) d'optimiser ces matrices en utilisant PSO et AGs. Dans ce chapitre, nous proposons d'utiliser l'algorithme BBO, qui est un algorithme métaheuristique, inspiré par des modèles mathématiques de la biogéographie, afin d'obtenir une haute performance d'estimation des états.

VI.3. Structure générale de la commande adoptée

Dans cette section, nous proposons une nouvelle alternative pour l'ajustement et l'optimisation de Q et R basée sur l'algorithme BBO. La structure de la méthode est constituée de deux étapes. Dans la première étape, nous présentons une structure BBO-EKF (la combinaison de BBO avec EKF) qui fonctionne en hors-ligne et permet de trouver les valeurs optimales de Q et R. Dans un second temps, les valeurs obtenues Q et R dans l'étape 1 sont injectées dans l'estimateur EKF, qui va ensuite fonctionner en ligne pour estimer le vecteur d'état, telle que schématisée par la figure VI.1.

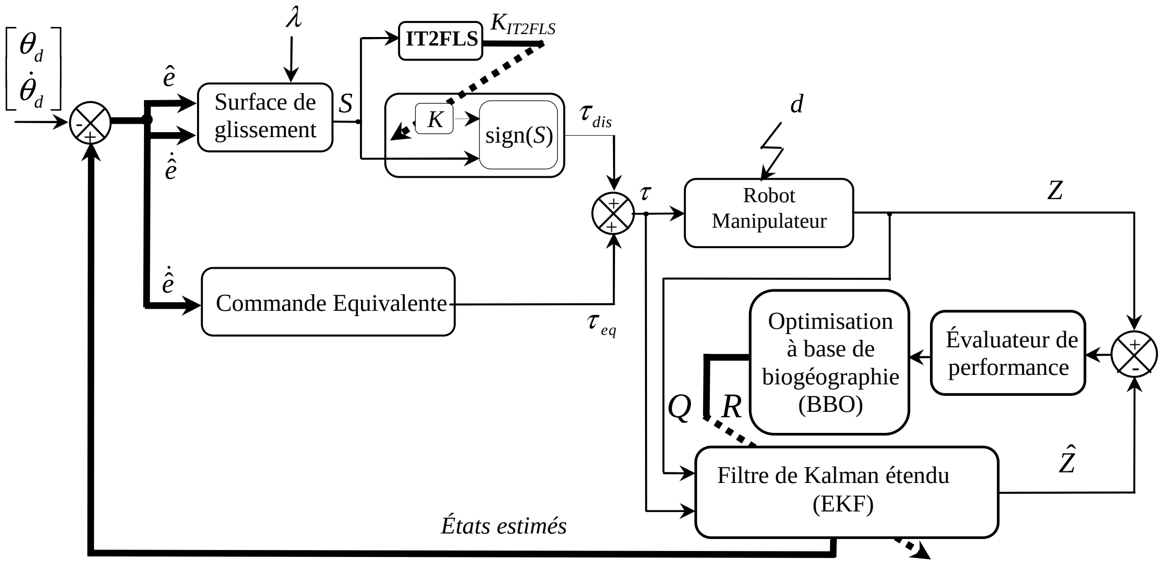


Fig. VI.1. Diagramme schématique de commande avec EKF optimisé par BBO

La détermination des matrices de covariance Q et R est une tâche difficile, en particulier lorsque les bruits correspondants ont des propriétés stochastiques inconnues. Pour éviter la complexité computationnelle de la méthode essai-erreur, et lorsque les valeurs de ces matrices ne sont pas connues précisément, nous considérerons ces matrices comme des paramètres libres à ajuster. Nous proposons d'utiliser pour ce propos l'algorithme BBO.

Dans cette partie, la tâche principale du BBO est représentée sur la figure V.1, dans laquelle nous l'exécutons de manière hors-ligne avec EKF afin de trouver la solution optimale pour les matrices de covariance Q et R. Le critère d'erreur quadratique moyenne (MSE) est utilisé comme fonction objectif, entre la sortie réelle et la sortie estimée, selon un certain nombre d'itérations N à effectuer pour chaque étape d'estimation.

L'entrée de commande τ et la réponse mesurée Z seront considérées comme des signaux d'entrée pour l'observateur EKF, où τ est appliqué à la fois au système non-linéaire et au filtre de Kalman étendu.

La sortie réelle Z et la sortie estimée $\hat{Z}$ sont définies comme étant les entrées d'évaluateur de performance du module BBO via un comparateur. La fonction objectif MSE est calculée par l'évaluateur de performance. Ensuite, les valeurs obtenues de l'MSE sont utilisées dans l'algorithme BBO. Sur la base de ces valeurs, l'optimiseur BBO va calculer et optimiser les paramètres inconnus des matrices de covariance Q et R.

Les nouvelles solutions et la mise à jour de Q et R sont ensuite utilisées pour adapter EKF à l'itération suivante jusqu'à ce qu'un nombre prédéfini d'itérations soit atteint.

Enfin, les matrices Q et R optimisées sont injectées dans l'observateur EKF pour une future exécution en ligne.

VI.4. Synthèse de la loi de commande

La première étape de l'SMC consiste à concevoir la surface de glissement S, qui peut être définie comme

$$S\left(\hat{X},t\right)=\dot{\hat{e}}+\lambda\hat{e} \tag{VI.4}$$

où λ est une matrice diagonale, définie positive.

La dérivée temporelle de (VI.4) est donnée comme:

$$\dot{S}\left(\hat{X},t\right)=\ddot{\hat{e}}+\lambda\dot{\hat{e}} \tag{VI.5}$$

La deuxième étape de l'SMC est de concevoir la loi de commande. Dans l'SMC traditionnel, la loi de commande discontinue est définie comme suit:

$$\tau_{dis}=-K\,\mathrm{sign}(S) \tag{VI.6}$$

où K est une matrice diagonale, définie positive.

La commande équivalente τ_{eq} est donnée par la condition suivante [3]:

$$S=0 \text{ et } \dot{S}=0 \Rightarrow \tau\left(t\right)=\tau_{eq}\left(t\right) \tag{VI.7}$$

Afin de gouverner les états du système, il suffit de suivre la trajectoire de référence désirée pour rendre $S=0$ attractif. Par conséquent, $\hat{e}\to 0$ quand $t\to\infty$.

Une condition suffisante pour réaliser ce comportement est de sélectionner la stratégie de commande de sorte que la condition de glissement suivante [160][3] soit satisfaite:

$$\frac{1}{2}\frac{d}{dt}S^2\leq-\eta\left|S\right|,\ \eta>0 \tag{VI.8}$$

Considérons le problème de contrôle du système non-linéaire incertain (VI.2), la loi de commande par mode glissant traditionnel est alors défini par :

$$\begin{aligned}\tau&=\tau_{eq}+\tau_{dis}\\&=M_0(\ddot{\theta}_d-\lambda\dot{\hat{e}})+C_0(\dot{\theta}_d-\lambda\hat{e})+G_0-d-K\,\mathrm{sign}(S)\end{aligned} \tag{VI.9}$$

L'utilisation de la fonction discontinu signe provoquera le phénomène de broutement. En plus de ce que nous avons présenté dans les chapitres précédents (chapitre IV et V), pour résoudre ce problème, nous adoptons l'intégration de la logique floue type-1 avec la commande discontinue.

Dans ce chapitre, un système flou type-2 intervalle [182][14][183][19] a été utilisé pour réduire le phénomène de broutement, cette idée consiste en la combinaison d'un IT2FLS avec la commande discontinu de la commande SMC selon certaines règles floues type-2 appropriées.

Dans ce contexte, un gain de commutation élevé K de (VI.9) conduira à une augmentation des oscillations du signal d'entrée de commande, et par conséquent une excitation de dynamique à haute fréquence, ce qui va créer le phénomène de broutement. De plus, un faible gain de commutation peut réduire le phénomène du broutement et améliorer la performance de poursuite malgré les incertitudes et les perturbations externes. Cependant, l'augmentation du gain provoque une augmentation des oscillations dans le signal d'entrée de commande autour de la surface de glissement. Pour atteindre une performance plus appropriée, ce gain doit être ajusté. Ce réglage est basé sur la distance entre les états du système et la surface de glissement, c'est-à-dire que le gain devrait être élevé lorsque la trajectoire d'état est éloignée de la surface de glissement, et lorsque la distance diminue, il devrait être réduit. Cette idée peut être réalisée en intégrant la logique floue type-2 avec la commande discontinue pour ajuster le gain K de manière adaptative selon certaines règles floues type-2 appropriées. Sous cette condition, le gain K est adapté par un gain flou K_{IT2FLS} comme le montre la figure VI.1.

Pour cette raison, un IT2FLS à une entrée et une sortie est conçu, l'entrée S reflète la distance de la trajectoire d'erreur à la surface de glissement, et la sortie du système flou type-2 est désignée par K_{IT2FLS}.

Le système flou type 2 (T2FLS) [14][183] se compose de cinq parties: le fuzzificateur, la base de règles, le moteur d'inférence floue, le réducteur de type et le defuzzificateur. La base de règles est composée d'une collection de règles ***SI…ALORS*** floues dont elles peuvent être énoncées de manière linguistique suivante:

$$\text{Règle } i: \textbf{\textit{SI}} \;\; S \;\; \text{est} \;\; A^i, \textbf{\textit{ALORS}} \;\; K_{IT2FLS} \;\; \text{est} \;\; B^i \;\;, \; i = 1, 2, ..., N$$

où A^i et B^i sont des ensembles flous type-2 intervalle, N est le nombre total de règles.

Notez que les fuzzifications singleton, la défuzzification avec la méthode du centre de gravité, l'implication de Mamdani type-2 intervalle et le moteur d'inférence de produit sont utilisés dans ce chapitre.

Le processus de correspondance entrée-sortie peut être formulé comme suit:

1. Calculez l'intervalle de poids de chaque règle:

$$\omega_i \in \left[\min(\mu_{A_i}) \quad \max(\mu_{A_i}) \right] \tag{VI.10}$$

2. Calculer la sortie pondérée de toutes les règles (réduction de type) basé sur le centre des ensembles du réducteur de type [175]

$$\begin{aligned} y_l(S) &= \min_{\omega_i} \left(\frac{\sum y_l(S).\omega_i}{\sum \omega_i} \right), \\ y_r(S) &= \min_{\omega_i} \left(\frac{\sum y_r(S).\omega_i}{\sum \omega_i} \right) \end{aligned} \tag{VI.11}$$

Par conséquent, la sortie du système flou type-2 intervalle (defuzzification) peut être calculée sur la base de la moyenne arithmétique, par l'équation suivante:

$$K_{IT2FLS}(S) = \frac{y_l + y_r}{2} \tag{VI.12}$$

Par conséquent, la loi de commande (VI.9) devient:

$$\begin{aligned} \tau &= \tau_{eq} + K_{IT2FLS}\, \tau_{dis} \\ &= M_0(\ddot{\theta}_d - \lambda\dot{\hat{e}}) + C_0(\dot{\theta}_d - \lambda\hat{e}) + G_0 - d - K_{IT2FLS}.K.\mathrm{sign}(S) \\ &= M_0(\ddot{\theta}_d - \lambda\dot{\hat{e}}) + C_0(\dot{\theta}_d - \lambda\hat{e}) + G_0 - d - \hat{K}.\mathrm{sign}(S) \end{aligned} \tag{VI.13}$$

où $\hat{K} = K_{IT2FLS}.K$.

Par conséquent, il peut être facilement vérifié que (VI.13) est suffisant pour imposer la condition de glissement

$$\frac{1}{2}\frac{d}{dt}S^2 = S\dot{S} \le -\hat{K}\left|S\right| \tag{VI.14}$$

ce qui assure, la convergence en temps fini de l'erreur de poursuite vers la surface de glissement S et, par conséquent, sa stabilité exponentielle.

Afin de dominer les états du système pour arriver à la surface de glissement $s = 0$ dans un temps limité et y rester, la loi de commande doit être conçue de sorte que la condition de glissement décrite en (VI.14) soit satisfaite. Cet objectif est assuré par le théorème suivant.

Théorème VI.1:

Considérant le système non-linéaire incertain (VI.2). Supposons que les hypothèses VI.1 à VI.4 sont satisfaites. Si on choisit la loi de commande (VI.13), alors, la condition (VI.14) est satisfaite, ce qui assure la convergence du vecteur d'erreur de poursuite sur la surface de glissement S

Preuve:

Considérons la fonction candidate de Lyapunov suivante :

$$L = \frac{1}{2} S^T M S \tag{VI.15}$$

Sa dérivée temporelle est donnée par :

$$\dot{L} = \frac{1}{2} S^T \dot{M} S + S^T M \dot{S} \tag{VI.16}$$

Considérant la propriété 2, alors

$$S^T \left(\frac{1}{2} \dot{M} - C \right) S = 0 \tag{VI.17}$$

En combinant (VI.15) - (VI.17), on peut obtenir

$$\begin{aligned}
\dot{L} &= \frac{1}{2} S^T (\dot{M} - 2C) S + S^T C S + S^T M \dot{S} \\
&= S^T (CS + M\dot{S}) = S^T \left[CS + M(\ddot{\hat{e}} + \lambda \dot{\hat{e}}) \right] \\
&= S^T \left[CS + M(\ddot{\hat{\theta}} - \ddot{\theta}_d) + M\lambda(\dot{\hat{\theta}} - \dot{\theta}_d) \right] \\
&= S^T \left[C(\dot{\hat{\theta}} - \dot{\theta}_d) + C\lambda(\hat{\theta} - \theta_d) - M\ddot{\theta}_d - C\dot{\hat{\theta}} - G + \tau + M\lambda(\dot{\hat{\theta}} - \dot{\theta}_d) \right] \\
&= S^T \left[M[\lambda(\dot{\hat{\theta}} - \dot{\theta}_d) - \ddot{\theta}_d] + C[\lambda(\hat{\theta} - \theta_d) - \dot{\theta}_d] - G + \tau \right] \\
&= S^T \left[M_0[\lambda(\dot{\hat{\theta}} - \dot{\theta}_d) - \ddot{\theta}_d] + C_0[\lambda(\hat{\theta} - \theta_d) - \dot{\theta}_d] - G_0 + \tau + d \right]
\end{aligned} \tag{VI.18}$$

Pour obtenir la stabilité en boucle fermée, la dérivée de la fonction de Lyapunov (VI.18) doit être négative $\left(\dot{L} \leq 0\right)$, la loi de commande τ peut être choisie comme:

$$\tau = M_0\left[\ddot{\theta}_d - \lambda\left(\dot{\hat{\theta}} - \dot{\theta}_d\right)\right] + C_0\left[\dot{\theta}_d - \lambda\left(\hat{\theta} - \theta_d\right)\right] + G_0 - d - \hat{K}\,\text{sign}\left(S\right) \qquad \text{(VI.19)}$$

Par conséquent, l'équation (VI.18) devient:

$$\dot{L} = -S^T \hat{K}\,\text{sign}\left(s\right) = -\hat{K}\left|S\right| \leq 0 \qquad \text{(VI.20)}$$

ce qui implique $S(t) \to 0$ quand $t \to \infty$. Par conséquent, $\hat{e}(t)$ et tous ses dérivés convergent vers zéro. □

VI.5. Simulation et résultats

Pour illustrer la performance de l'approche proposée, nous considérons la commande d'un bras manipulateur à deux articulations représenté par la figure V.2, où son modèle dynamique non-linéaire est donné par les équations (V.25)-(V.32), décrites dans le chapitre V.
En utilisant la méthode d'Euler directe, nous obtenons la représentation d'état discrète suivante :

$$\begin{cases} x_1(k+1) = x_1(k) + \Delta t.\ x_2(k) + w_1(k) \\ x_2(k+1) = x_2(k) + \Delta t\left[\ f_1(X,k) + g_1(X,k)\,\tau_1(k) + d_1(k)\right] + w_2(k) \\ x_3(k+1) = x_3(k) + \Delta t.\ x_4(k) + w_3(k) \\ x_4(k+1) = x_4(k) + \Delta t\left[\ f_2(X,k) + g_2(X,k)\,\tau_2(k) + d_2(k)\right] + w_4(k) \\ Z_1(k) = x_1(k) + v_1(k) \\ Z_2(k) = x_3(k) + v_2(k) \end{cases} \qquad \text{(VI.21)}$$

où $w(k) = [w_1(k)\ w_2(k)\ w_3(k)\ w_4(k)]$ et $v(k) = [v_1(k)\ v_2(k)]$ sont les vecteurs des bruits d'états blancs Gaussiens de et de mesure avec des moyennes nulles. Δt est la période d'échantillonnage. k est l'instant discret.
EKF est implémenté comme représenté par les équations (III.57) à (III.59) où les matrices jacobéennes sont définies en annexe B.

EKF fournira le vecteur d'estimation d'état $\hat{X} = [\hat{\theta}_1, \dot{\hat{\theta}}_1, \hat{\theta}_2, \dot{\hat{\theta}}_2]^T = [\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4]^T$. Les conditions initiales d'état d'EKF sont choisies telles que $\hat{X}_{0/0} = [0,0,0,0]^T$.

La matrice de covariance d'erreur P est initialement définie comme une matrice unitaire de dimension 4×4, alors que les matrices de covariances des bruits Q et R ont des dimensions 4×4 et 2×2, respectivement, et sont supposées avoir la forme suivante:

$$Q = diag(q_{x_1}, q_{x_2}, q_{x_3}, q_{x_3}) = \begin{bmatrix} q_{x_1} & 0 & 0 & 0 \\ 0 & q_{x_2} & 0 & 0 \\ 0 & 0 & q_{x_3} & 0 \\ 0 & 0 & 0 & q_{x_4} \end{bmatrix} \quad \text{(VI.22)}$$

$$R = diag(r_1, r_2) = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \quad \text{(VI.23)}$$

Les valeurs de q_{x_i} et r_i dans les matrices Q et R sont généralement déterminées en utilisant une méthode d'essai-erreur. A des fins de comparaison, la performance de l'observateur EKF avec diverses compositions de Q et R est évaluée en utilisant l'erreur quadratique moyenne entre la position estimée et la position réelle qui est définie par l'équation (V.36).

Dans cette simulation, nous gardons les mêmes paramètres nominaux du robot que ceux définis précédemment (chapitre V). Un total de $N = 2000$ données de mesure sont simulées sur un intervalle de temps de 0 à 2 secondes avec un pas d'échantillonnage $\Delta t = 0.001\text{s}$.

Les trajectoires de référence désirées pour x_1 et x_3 sont choisies $x_{1d}(t) = 70^{\circ}$ et $x_{3d}(t) = 90^{\circ}$.

Les conditions initiales choisies sont : $x_1(0) = 0$, $x_2(0) = 0$, $x_3(0) = 0$ et $x_4(0) = 0$.

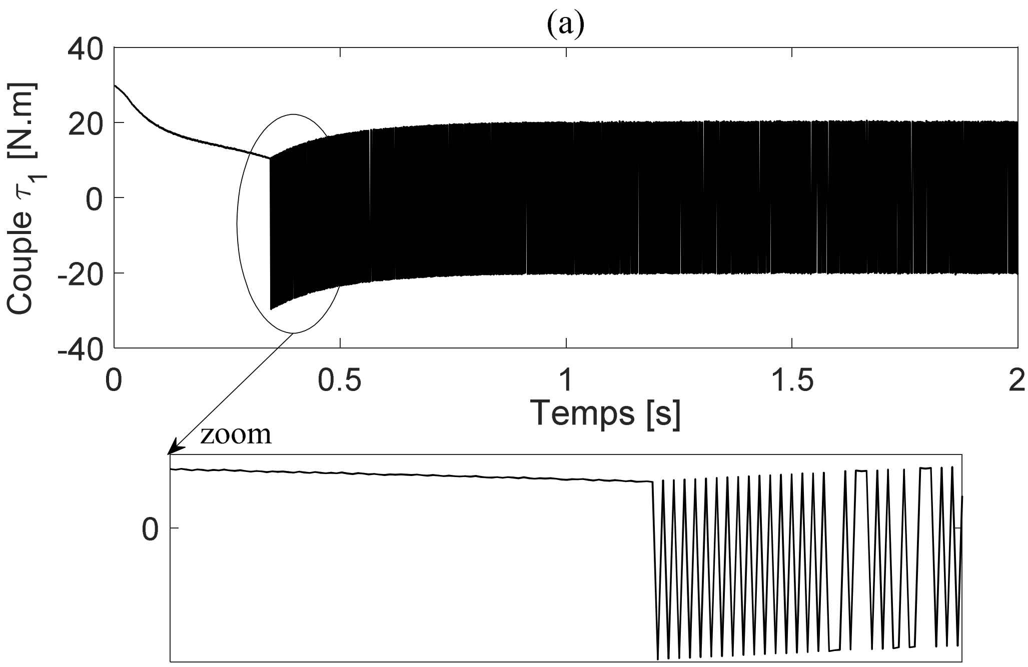
Trois types d'incertitudes sont injectés dans la structure pour vérifier la robustesse du contrôleur: (1) Les incertitudes des paramètres ($+10\%$ sur les valeurs des paramètres nominaux du modèle). (2) Les perturbations externes aléatoires choisies pour être réparties uniformément comme suit: $d_1 = 0.3 \times \text{rand}$ et $d_2 = 0.3 \times \text{rand} \times \sin(t)$, $D = 1$. Notez que $d = [d_1, d_2]^T$ est appliquée à $t > 1\text{s}$. (3) Bruits gaussiens aléatoires pour les états et les mesures avec des valeurs moyennes nulles et avec des covariances $q = 10^{-2}$ et $r = 10^{-4}$, respectivement.

Dans un premier temps, nous avons simulé le système avec la commande mode glissant seulement, afin de montrer son inconvénient. En appliquant la loi de commande (VI.9), et après quelques essais, nous avons choisi $K = 20 I_{2\times2}$ et $\lambda = 5 I_{2\times2}$ où $I_{i\times i}$ est une matrice d'identité de dimension $i \times i$.

Tableau VI.1. Performances d'EKF pour un robot à 2-articulations par l'estimation essai-erreur

Cas	**entrées *Q* et *R***						**MSE_EKF [rad]**	**Qualité d'estimation**
	q_{x_1}	q_{x_2}	q_{x_3}	q_{x_4}	r_1	r_2		
1	1	1	1	1	1	1	1.0881	Mauvaise
2	0.1	0.1	10^{-3}	0.1	10^{-4}	10^{-4}	1.9894×10^{-5}	Bonne
3	0.1	0.01	0.1	0.1	0.1	10^{-6}	1.4567×10^{-5}	Bonne
4	0.01	0.01	0.02	0.01	0.01	0.08	1.4287×10^{-5}	Très bonne

Le tableau VI.1 montre les performances typiques d'EKF avec leurs matrices de covariance correspondantes (avec les éléments q_{x_1}, q_{x_2}, q_{x_3}, q_{x_4}, r_1 et r_2) obtenues par la méthode essai-erreur. On constate que de bons résultats de performance d'estimation ont été obtenus lorsque *Q* et *R* sont égaux (cas 2 et 3 dans le tableau VI.1), mais une mauvaise sélection de (q_{x_1}, q_{x_2}, q_{x_3}, q_{x_4}, r_1 et r_2) peut produire une mauvaise performance d'estimation (cas 1). Notez que la meilleure performance d'estimation est obtenue avec les matrices *Q* et *R* ($q_{x_1} = q_{x_2} = q_{x_4} = 0.01$ $q_{x_3} = 0.02$, $q_{r_1} = 0.01$, $q_{r_2} = 0.08$) (cas 4) qui correspondent au plus petit MSE. Les couples de commande par rapport au meilleur cas (cas 4) sont montrés sur la figure VI.2(a) et (b).



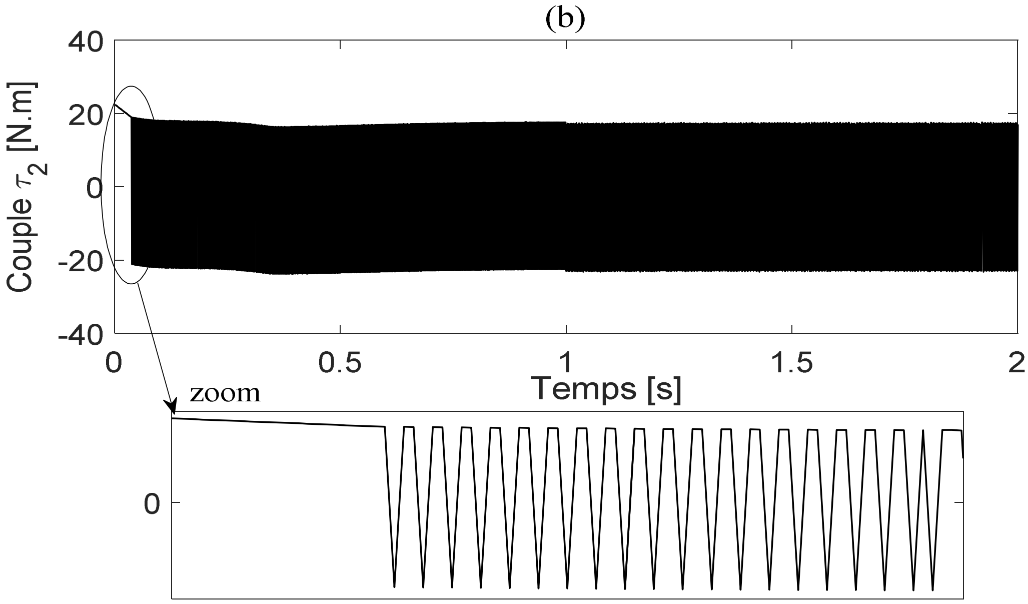


Fig. VI.2. Les couples de commande en utilisant l'SMC traditionnel avec EKF standard
(a) Articulation-1, (b) Articulation-2

La figure VI.2 représente l'allure des deux couples de commande. À partir des zooms de ces couples, on observe clairement, que les performances des couples de commande ne sont pas satisfaisantes en raison du phénomène de broutement provoqué par la sélection inappropriée du gain de commutation.

Afin de résoudre ce problème, la propriété de lissage de la logique floue type-2 intervalle est exploitée comme nous l'avons vu dans la section I.5 (chapitre I). En utilisant (VI.13), les fonctions d'appartenance de S sont choisies comme illustré sur la figure 3(a), dans laquelle les variables linguistiques suivantes ont été utilisées: **négatif (N)**, **zéro (Z)**, **positif (P)**. Les fonctions d'appartenance de K_{IT2FLS} sont choisies comme illustré sur la figure 3(b), dans laquelle les variables linguistiques suivantes ont été utilisées: **positif petit (PP)**, **positif grand (PG)**.

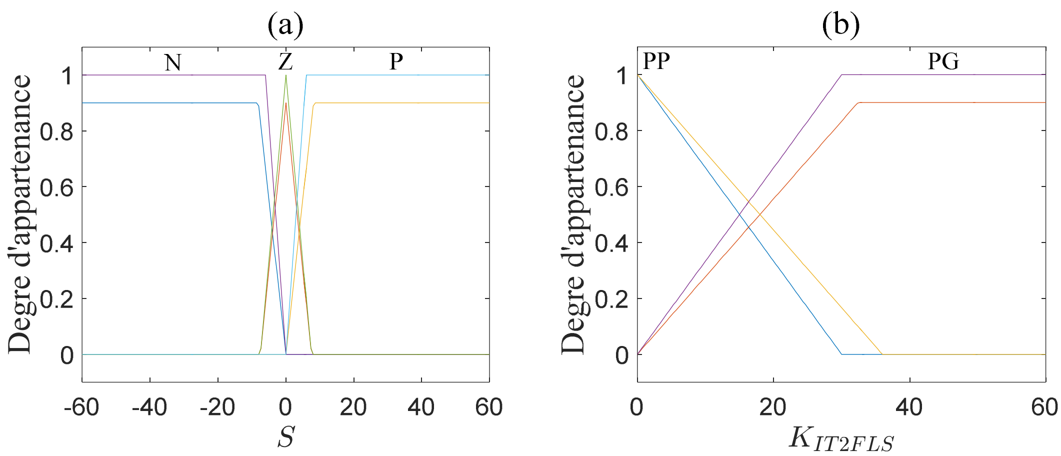


Fig. VI.3. (a) Fonctions d'appartenance d'entrée (b) Fonctions d'appartenance de sortie

L'ensemble de règles floues adoptées contient 3 règles définies comme suit:

Règle 1 : **Si** S est N , **Alors** K_{IT2FLS} est PG

Règle 2 : **Si** S est Z, **Alors** K_{IT2FLS} est PP

Règle 3 : **Si** S est P, **Alors** K_{IT2FLS} est PG

Ces règles régissent la relation entrée-sortie entre S et K_{IT2FLS} en adoptant le moteur d'inférence de type Mamdani, dans lequel la méthode du centre de gravité est utilisée pour la défuzzification comme dans (VI.12). En considérant l'hypothèse 4, dans laquelle $-30 \leq \tau(t) \leq +30$.

Les figures VI.4 à VI.7 exposent les résultats de simulation issus de cette application.

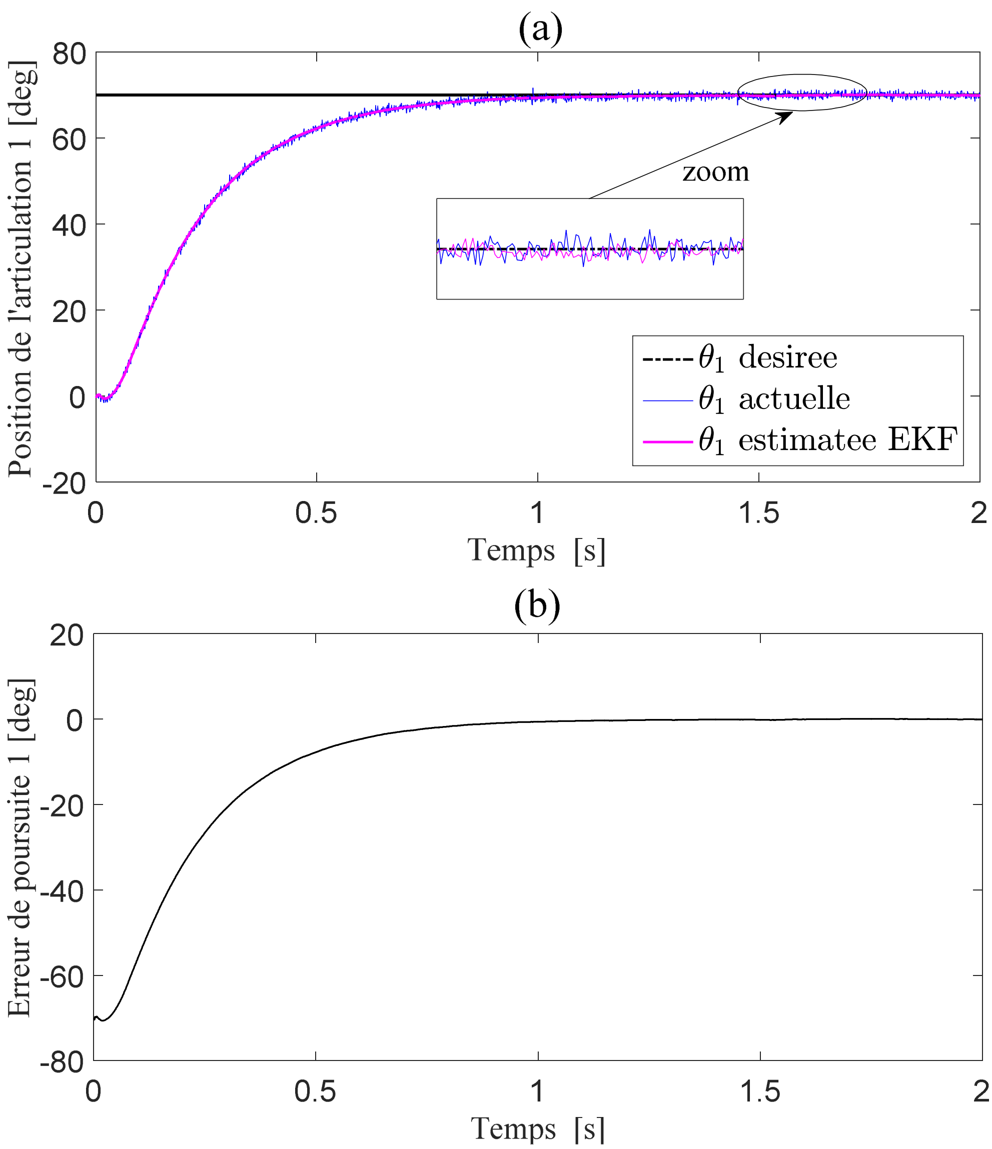


Fig. VI.4. (a) Position désirée, réelle et estimée de l'articulation-1 (b) Erreur de poursuite de position de l'articulation-1, en utilisant IT2FSMC avec EKF standard.

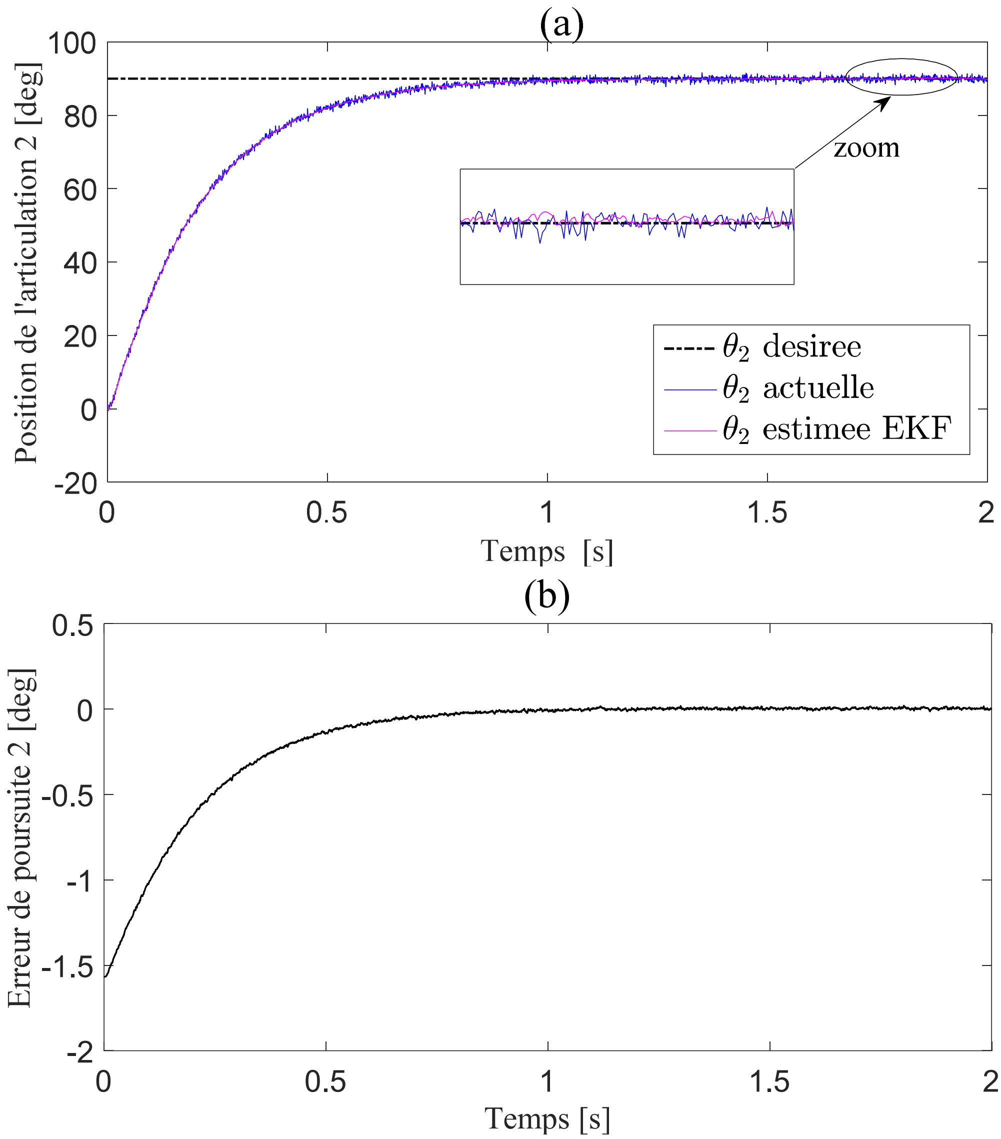
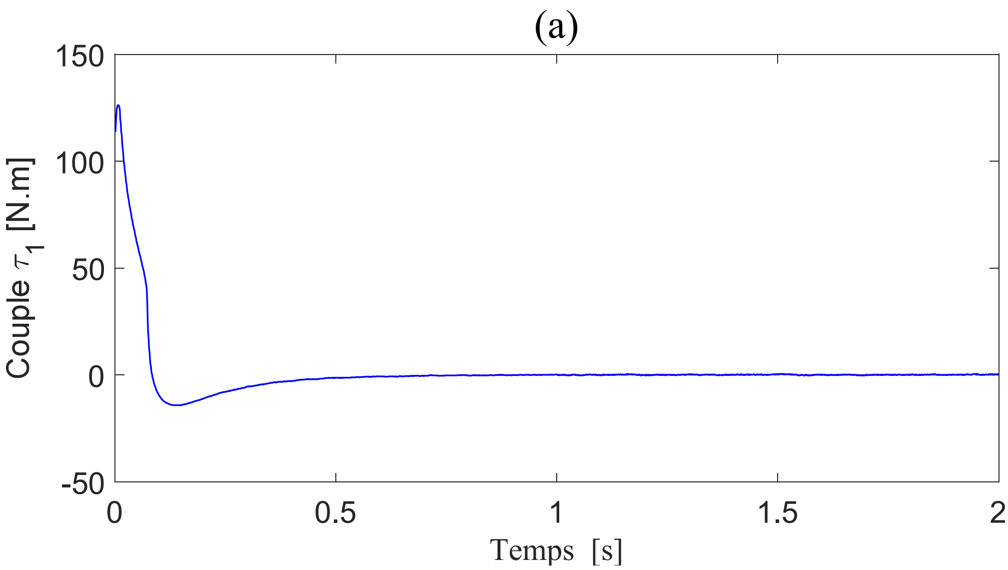


Fig. VI.5. (a) Position désirée, réelle et estimée de l'articulation-2 (b) Erreur de poursuite de position de l'articulation-2, en utilisant IT2FSMC avec EKF standard.



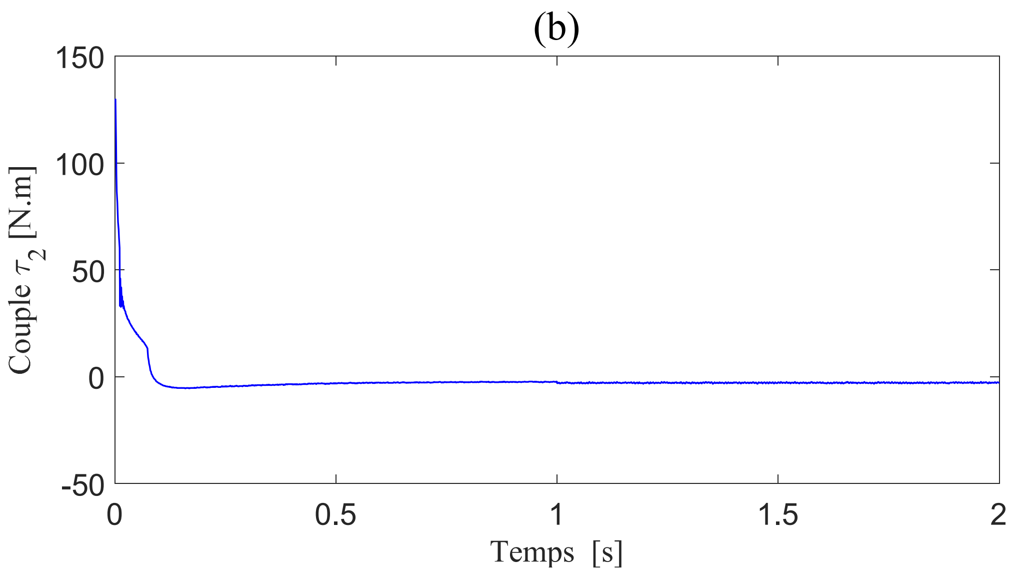


Fig. VI.6. Couples de commande en utilisant IT2FSMC avec EKF standard appliquées (a) Articulation-1, (b) Articulation-2.

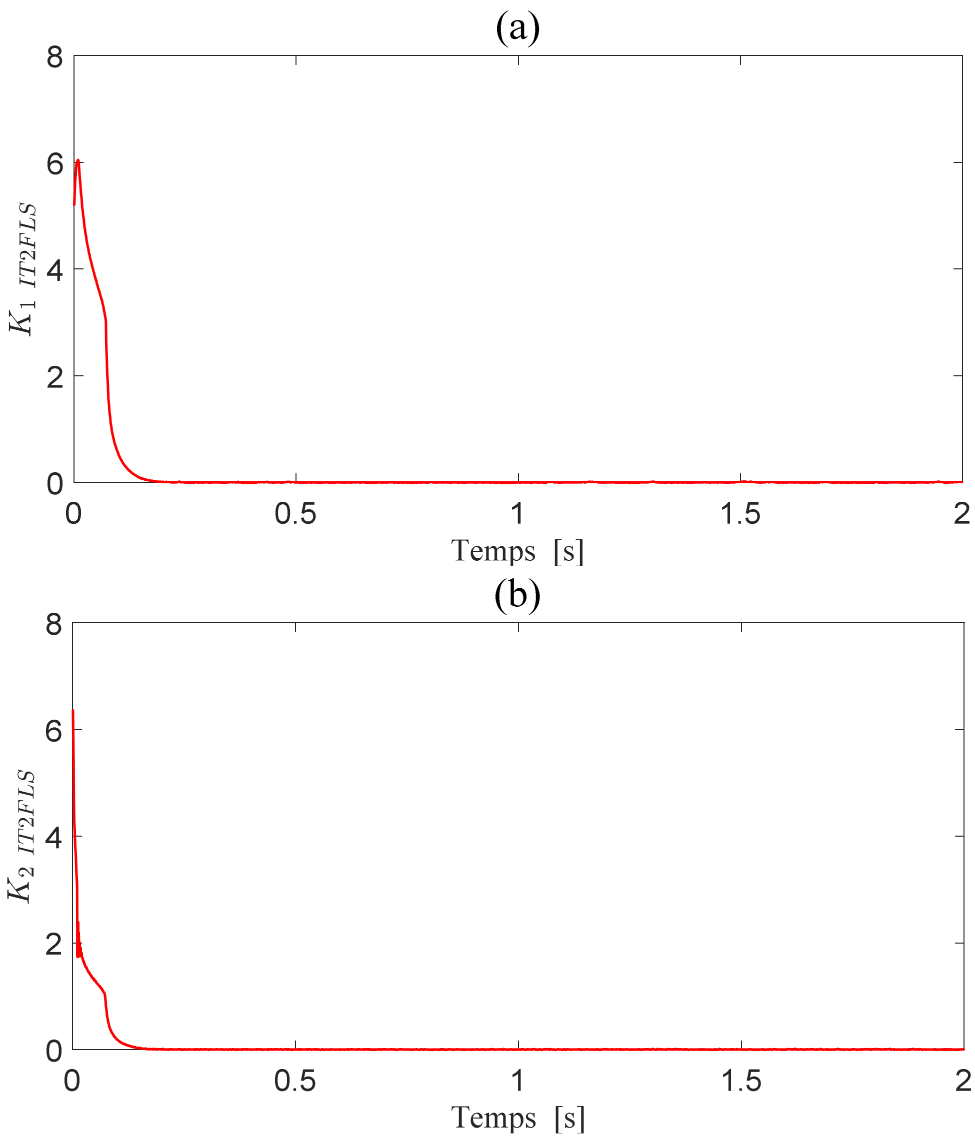


Fig. VI.7. (a) Evolution temporelle du gain flou $K_{1\ IT2FLS}$ (b) Evolution temporelle du gain flou $K_{2\ IT2FLS}$

La figure VI.4(a), présente la position désirée, réelle et estimée de l'articulation-1, et comme nous voyons la performance du système sous la présence des variations paramétriques et les perturbations externes est satisfaisante (voir aussi la figure VI.4(b)), qui représente l'erreur de poursuit en position de l'articulation-1.

La figure VI.5(a) montre la position désirée, réelle et estimée de l'articulation-2, et comme nous voyons la performance du système sous la présence des variations paramétriques et les perturbations externes est satisfaisante (voir aussi la figure VI.5(b)), qui représente l'erreur de poursuit en position de l'articulation-2.

La figure VI.6 donne les deux nouveaux couples de commande appliqués au système, en les comparants avec les deux anciens couples de commande (figure VI.2), on peut voir clairement que le phénomène de broutement est presque disparu. Les gains flous K_{IT2FLS} adaptés sont représentés sur la figure VI.7.

A partir de ces résultats de simulation, il est clair que le contrôleur IT2FSMC fournit une réponse désirée avec des signaux de commande lisses en un temps d'atteinte minimal et cela sous la présence d'incertitudes et de perturbations. Notez que la précision de prédiction d'EKF n'est pas tout à fait satisfaisante (voir Figures VI.4(a) et VI.5(a)) en raison du choix de la méthode essai-erreur pour les matrices de covariance.

Notez que dans toutes les simulations ci-dessus, les matrices de covariance EKF ont été ajustées en utilisant la méthode d'essai-erreur, qui est simple à réaliser, mais qui prend beaucoup de temps. En fait, il n'est pas possible de déduire facilement une relation entre les matrices de covariance et les meilleurs résultats d'estimation. Dans ce qui suit, nous proposons de résoudre ce problème par un nouvel algorithme évolutif basé sur la biogéographie, appelé « optimisation basée sur la biogéographie». Puis, on va comparer avec les résultats qui seront obtenus avec l'algorithme PSO.

VI.5.1. Méthode BBO-EKF

Dans cette partie, nous proposons d'optimiser EKF par l'algorithme BBO. Cette dernière permet de fournir une solution optimale pour trouver les matrices de covariance optimales *Q* et *R* d'EKF. BBO-EKF est un algorithme d'optimisation combinant l'algorithme BBO avec l'observateur optimale EKF. Nous suggérons de rechercher la combinaison optimale de six variances q_{x_1}, q_{x_2}, q_{x_3}, q_{x_4}, r_1 et r_2 simultanément, en utilisant l'algorithme II.3 (voir chapitre II) pour trouver les matrices de covariance optimales *Q* et *R*, ce qui nous permettra d'obtenir de meilleures estimations avec une précision plus grande que la méthode d'essai-erreur.

Dans cette partie, nous avons exécuté le BBO-EKF d'une manière hors-ligne afin de trouver efficacement les matrices de covariance optimale *Q* et *R*. Le critère (évaluateur de la qualité de prédiction) qui a été utilisé est l'erreur quadratique moyenne (V.36) entre la sortie réelle et la sortie estimée en fonction d'un certain nombre d'itérations à effectuer pour chaque étape d'estimation.

Après avoir exécuté le BBO-EKF, les matrices de covariance optimisées *Q* et *R* et leurs performances MSE ont été illustrées dans le tableau VI.2. Il convient de noter que la convergence de la méthode BBO vers la solution optimale dépend des paramètres suivants :

Nombre d'habitats (taille de la population) $NP = 20$, *Nombre maximal de générations* égal à 100, *Nombre de variables de décision* (SIV) égal à 6, *Taux d'immigration* $\lambda = 1$, *Taux d'émigration* $E = 1$, *Coefficient d'absorption* $\alpha = 0.9$, et *Mutation de probabilité* $m_{\max} = 0.1$.

Tableau VI.2. Performances d'EKF optimisé à l'aide du BBO

Taille de l'essaim	Itérations	éléments de *Q* et *R*						MSE_BBO [rad]
		q_{x_1}	q_{x_2}	q_{x_3}	q_{x_4}	r_1	r_2	
20	5	0.0054	0.0484	0.0054	0.0035	0.0797	0.0666	5.7355×10^{-6}
	10	0.0029	0.0696	0.0063	0.0015	0.0823	0.0960	4.1097×10^{-6}
	50	0.0002	0.0312	0.0302	0.0295	0.0941	0.0643	3.7545×10^{-6}
	100	0.0001	0.0121	0.0564	0.0188	0.0842	0.0791	2.7394×10^{-6}

Dans le tableau VI.2, la meilleure solution est un habitat à faible valeur d'MSE. On observe, que le MSE est diminué à 2.7394×10^{-6} après 100 itérations. On notera que ce MSE est très petit par rapport à l' MSE obtenu par la méthode d'essai-erreur ($MSE_{essai\text{-}erreur} = 1.4287 \times 10^{-5}$), ce qui confirme l'efficacité de la méthode proposée.

VI.5.2. Méthode PSO-EKF

Afin de comparer les performances du processus BBO-EKF avec d'autres algorithmes, nous donnons dans le tableau VI.3, les matrices de covariance optimisées *Q* et *R* et les performances de l'estimation en utilisant la méthode PSO-EKF, où l'on voit bien que le MSE est diminué à 5.1932×10^{-6} après 100 itérations.

Notez que nous avons utilisé l'algorithme PSO avec les paramètres suivants: *Coefficient d'auto-reconnaissance* $c_1 = 1.49$, *Coefficient social* $c_2 = 1.49$, *Poids d'inertie* $w = 0.73$. Puisque nous avons six paramètres à optimiser, la dimension de la simulation sera de 6. Notez également que l'essaim simulé avec une taille de 20 avec un nombre maximal de générations égal à 100.

Tableau VI.3. Performances d'EKF optimisé à l'aide du PSO

Taille de l'essaim	**Itérations**	**éléments de *Q* et *R***						**MSE_PSO [rad]**
		q_{x_1}	q_{x_2}	q_{x_3}	q_{x_4}	r_1	r_2	
20	5	0.001	0.0214	0.01	0.0632	0.0793	0.0638	9.8340×10^{-6}
	10	0.0001	0.0335	10^{-5}	0.0326	0.0500	0.0832	9.7784×10^{-6}
	50	10^{-6}	0.0772	10^{-4}	0.0313	0.0673	0.0800	6.9638×10^{-6}
	100	10^{-5}	0.0675	10^{-5}	0.0157	0.0893	0.0923	5.1932×10^{-6}

A partir des résultats présentés dans les tableaux VI.3 et VI.4, si on s'intéresse uniquement à la fonction objectif qui est la minimisation de l'erreur quadratique moyenne, la comparaison des approches BBO-EKF et PSO-EKF montre que, tous les deux sont capables de trouver les matrices de covariance optimales *Q* et *R*. On peut constater que l'MSE obtenu en utilisant l'approche BBO-EKF est plus petit que l'MSE obtenu en utilise la méthode PSO-EKF, lorsque le nombre d'itérations (génération) augmente. Il est évident que l'algorithme proposée BBO-EKF permet d'avoir des résultats nettement supérieurs à ceux de la méthode PSO-EKF en terme de MSE. Par conséquent, il peut être confirmé que IT2FSMC combiné avec la technique BBO-EKF est plus performant que IT2FSMC combiné avec la technique PSO-EKF. Notez que la comparaison a été faite dans les mêmes conditions (nombre de génération, taille de la population, population initiale).

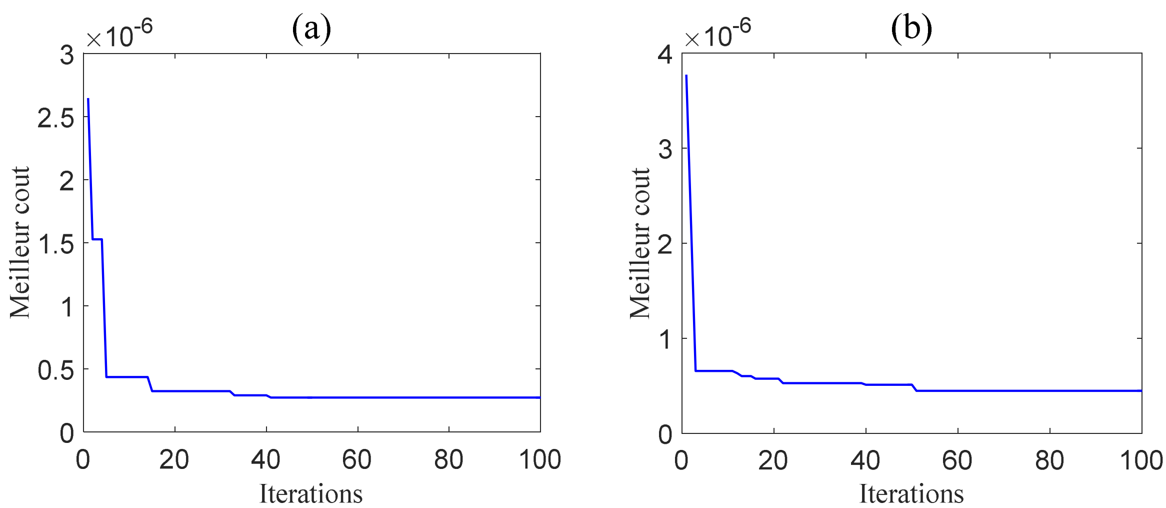


Fig. VI.8. Evolution de la fonction objectif / fitness versus 100 itérations.
(a) BBO par rapport au Tab. VI.2, (b) PSO par rapport au Tab. VI.3

La figure VI.8 montre l'évolution de la fonction objectif pour les meilleures solutions obtenues par les algorithmes BBO et PSO pendant100 itérations, où l'on remarque que la convergence du BBO est plus rapide que la convergence du PSO.

Dans ce qui suit, nous présentons les résultats de simulation finaux relatifs à la commande par mode glissant floue type-2 intervalle, avec les meilleurs paramètres optimaux d'EKF obtenu en utilisant l'algorithme BBO.

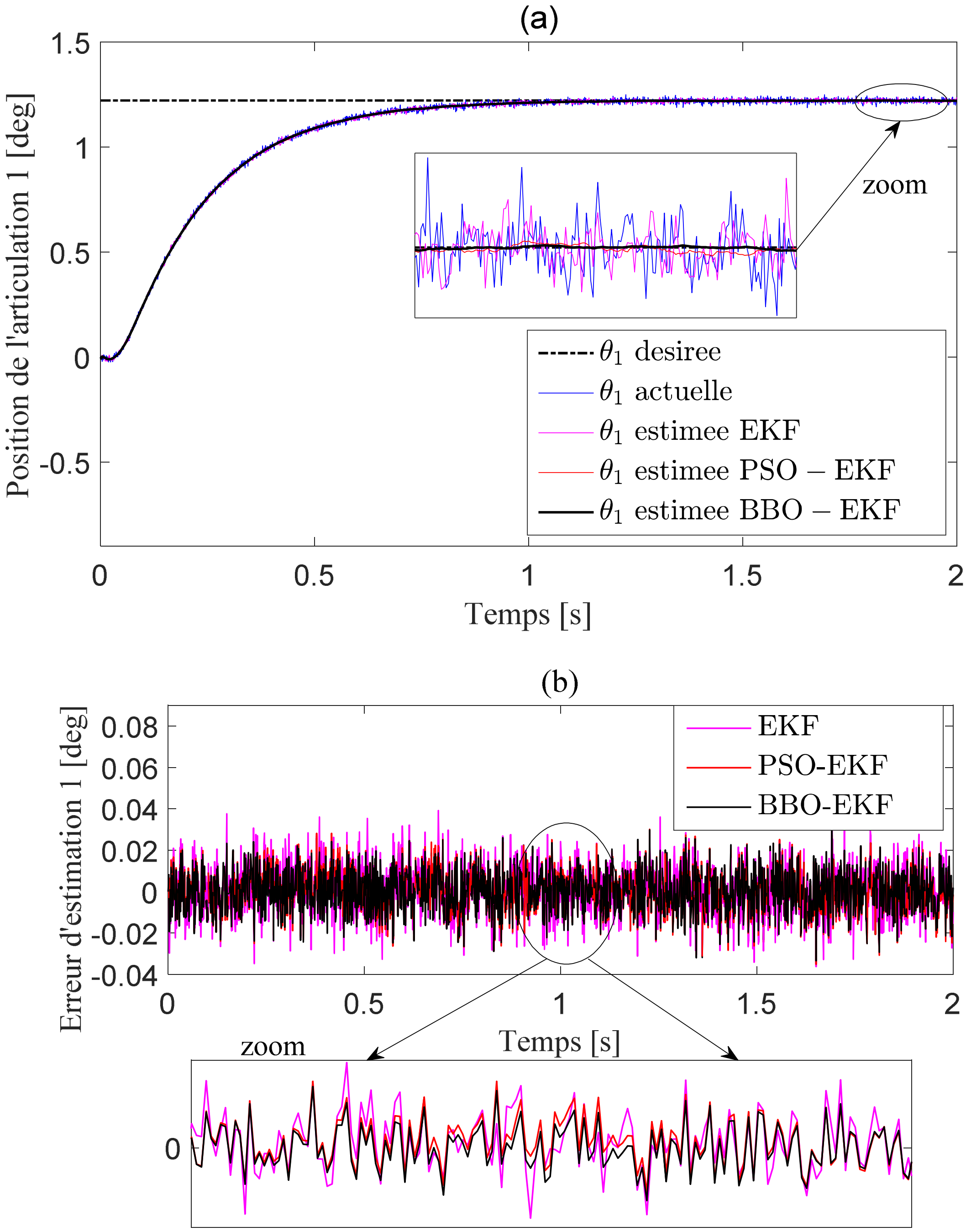


Fig. VI.9. (a) Positions désirée, réelle et estimée de l'articulation-1 (b) Erreurs d'estimation en position de l'articulation-1, en utilisant IT2FLS SMC pour différents algorithmes d'optimisation du EKF.

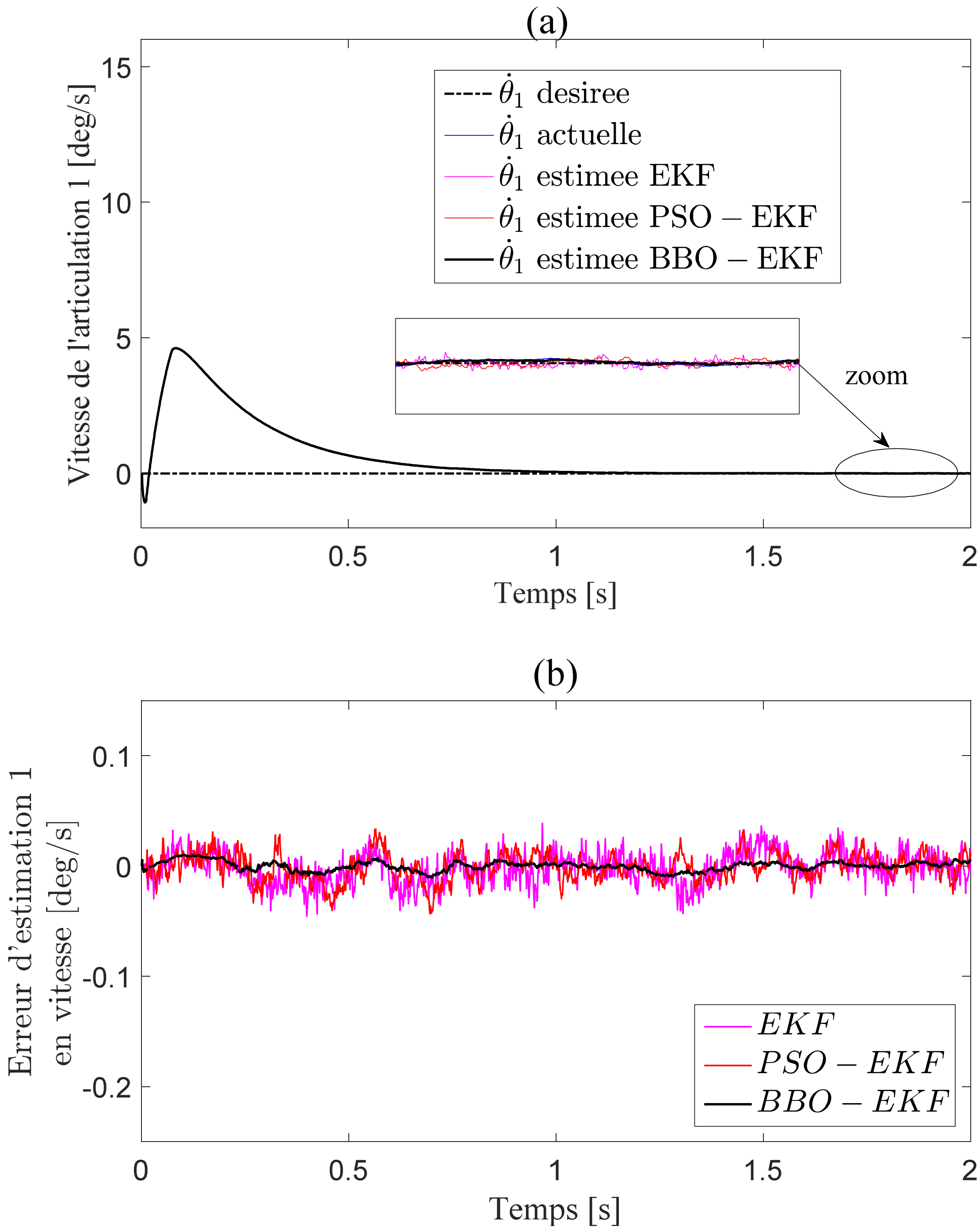


Fig. VI.10. (a) Vitesses désirée, réelle et estimée de l'articulation-1 (b) Erreurs d'estimation en vitesse de l'articulation-1, en utilisant IT2FLS SMC pour différents algorithmes d'optimisation du EKF.

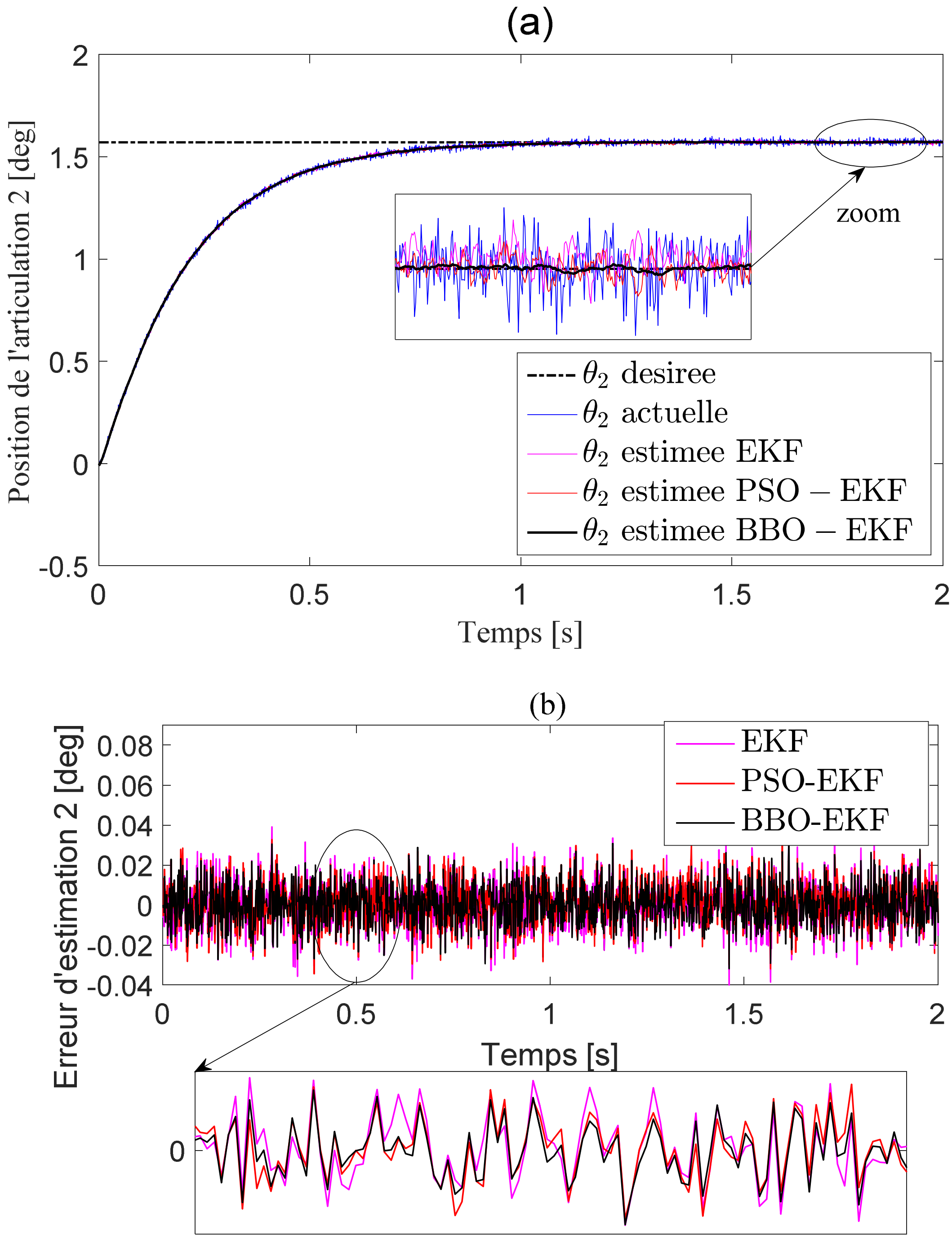


Fig. VI.11. (a) Positions désirée, réelle et estimée de l’articulation-2 (b) Erreurs d'estimation en position de l’articulation-2, en utilisant IT2FLS SMC pour différents algorithmes d'optimisation du EKF.

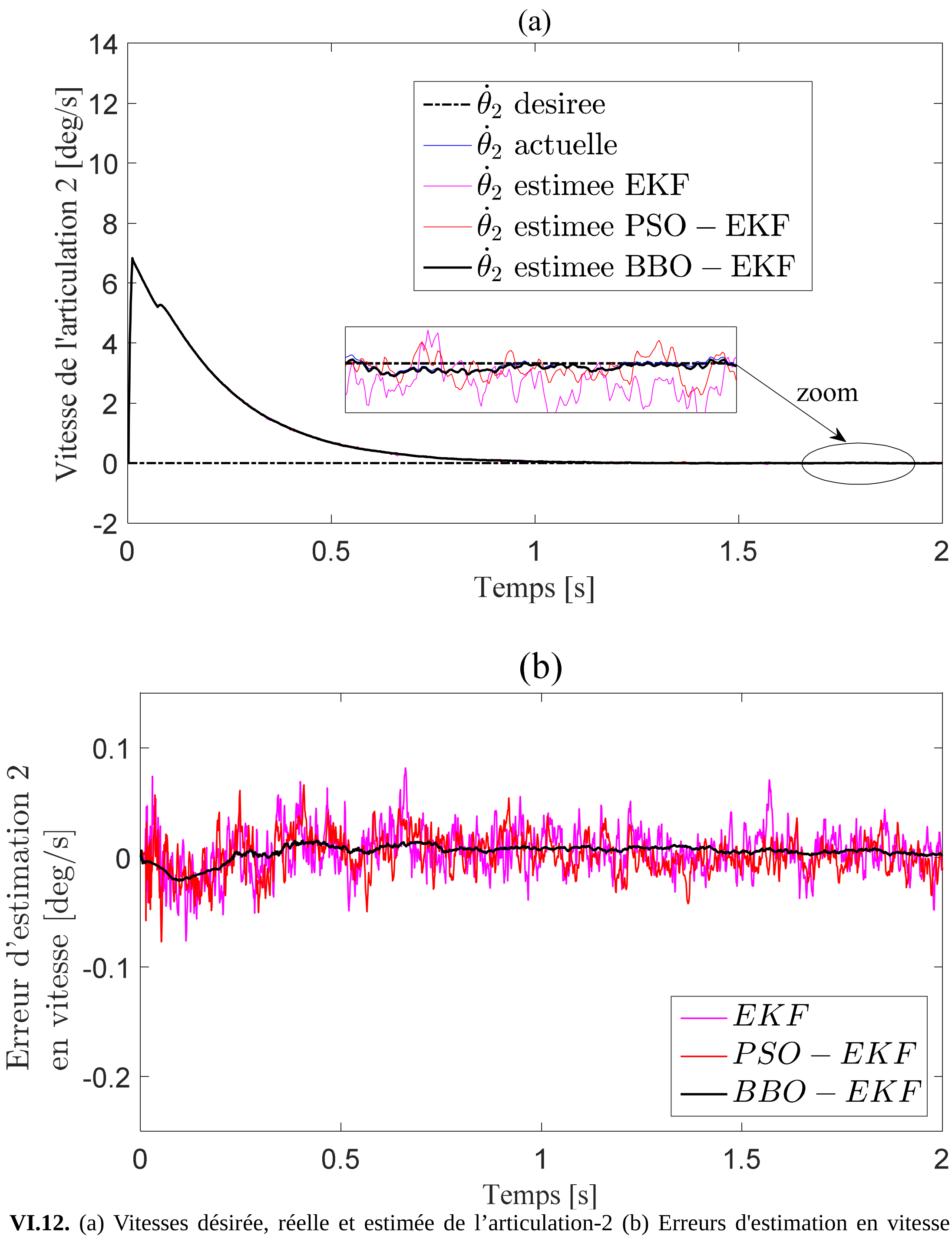


Fig. VI.12. (a) Vitesses désirée, réelle et estimée de l'articulation-2 (b) Erreurs d'estimation en vitesse de l'articulation-2, en utilisant IT2FLS SMC pour différents algorithmes d'optimisation du EKF.

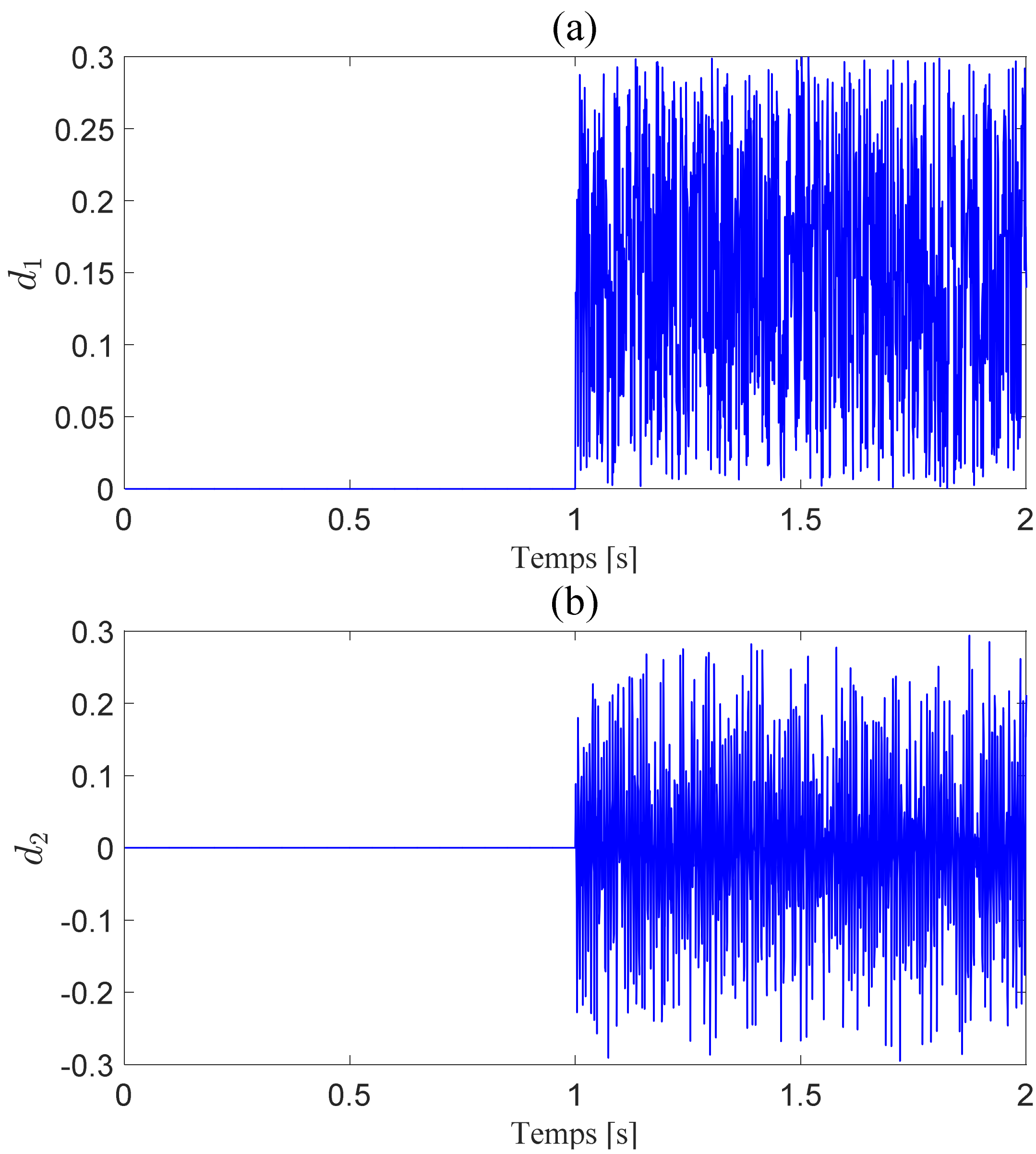


Fig. VI.13. Perturbations externes appliquées à (a) Articulation-1, (b) Articulation-2.

La figure VI.9(a), présente les réponses des positions désirées, réelles et estimées de l'articulation-1, avec les solutions optimales des matrices de covariance Q et R, représentées sur les tableaux VI.1, VI.2 et VI.3, obtenues en utilisant les méthodes : essai-erreur, BBO et PSO, respectivement. Les erreurs d'estimation correspondantes en position, sont présentées sur la figure VI.9(b). Les vitesses désirées, réelles et estimées correspondantes et leurs erreurs d'estimation sont présentées sur les figures VI.10(a) et (b), respectivement. Dans la figure VI.11(a), nous présentons les réponses de position désirées, réelles et estimées de l'articulation-2 avec les valeurs optimales des matrices de covariance Q et R, représentées sur les tableaux VI.1, VI.2 et VI.3, obtenues en utilisant les méthodes : essai-erreur, BBO et PSO, respectivement. Les erreurs d'estimation correspondantes en position, sont présentées sur la figure VI.11(b). Les vitesses désirées, réelles et estimées correspondantes et leurs erreurs d'estimation sont présentées sur les figures VI.12(a) et (b), respectivement.

À partir de ces figures, on observe qu'il y a de bonnes performances de poursuite de trajectoire en position et en vitesse des deux articulations, et d'estimation d'états du système. Les sorties du système convergent rapidement vers leurs références.

Les perturbations externes appliquées aux articulations sont représentées sur les Figures VI.13(a) et (b), respectivement. Dans toutes ces figures, Il est évident que la commande floue type-2 intervalle par mode glissant combiné avec le processus BBO-EKF, est plus performante que la commande floue type-2 intervalle par mode glissant combiné avec le processus PSO-EKF, pour un bras manipulateur à deux articulations.

VI.6. Conclusion

Dans ce chapitre, en considérant les incertitudes des paramètres et les perturbations externes, nous avons proposé une nouvelle application de l'algorithme d'optimisation basée sur la biogéographie, pour optimiser le filtre de Kalman étendu. La commande par mode glissant a été combinée avec un système flou type-2 intervalle, pour assurer une bonne robustesse. La stabilité du système en boucle fermée a été garantie en utilisant le critère de stabilité de Lyapunov. La performance d'EKF a été améliorée en ajustant les paramètres des matrices de covariance Q et R, dans lesquelles l'algorithme BBO a été utilisé, et il a été comparé à la technique PSO. La méthode d'optimisation proposée a permis de sélectionner les matrices de covariances optimales Q et R. Une comparaison entre la commande combinée avec BBO-EKF et celle avec PSO-EKF a été faite, en présence de bruits stochastiques. Cette comparaison a confirmé que la performance de la commande floue type-2 intervalle par mode glissant combinée avec le processus BBO-EKF est meilleure que celle du PSO-EKF. Les résultats de simulation ont montré une amélioration significative des performances en utilisant la méthode d'optimisation proposée sur un robot manipulateur à deux articulations.

Conclusion générale

La vie, c'est comme une bicyclette, il faut avancer pour ne pas perdre l'équilibre.
Albert Einstein

Les travaux présentés dans cette thèse de Doctorat, apportent des contributions aux problèmes de commande non-linéaire et d'estimation d'état pour les systèmes robotiques, tout en utilisant les techniques de l'intelligence artificielle.

Dans ce travail, nous sommes intéressés particulièrement aux systèmes robotiques, non-linéaires. Premièrement, il s'agit, de montrer que les logiques floues peuvent être utilisées pour approximer des lois de commande pour les systèmes non-linéaires. Dans les techniques de commande proposées, la propriété d'approximation universelle de la logique floue est exploitée pour approximer le gain de commutation de la commande discontinue. Deuxièmement, il s'agit de proposer des nouvelles alternatives pour la résolution du problème d'estimation du filtre de Kalman étendu. Dans les méthodes d'optimisation proposées, les paramètres optimisés sont des matrices de covariance du bruit d'état et du bruit de mesure Q et R, respectivement. Dans ce qui suit, nous donnons un résumé des résultats obtenus et des perspectives à notre travail.

Dans le chapitre I, nous avons abordé principalement deux approximations intelligentes: logiques floues type-1 et type-2. Dans un premier temps, nous avons présenté les notions de base pour un système flou type-1, nécessaires pour la compréhension de notre travail qui a été abordé dans les chapitres IV et V. Dans un deuxième temps, nous avons décrit les notions de base pour un système flou type-2, nécessaires pour la compréhension de notre travail qui a été abordé dans le chapitre VI.

Dans le chapitre II, nous avons décrit brièvement le principe des algorithmes génétiques, algorithme d'optimisation par essaim de particules et algorithme basé sur la biogéographie, qui permettent d'explorer de façon très efficace l'espace des solutions possibles d'un problème. Nous avons vu dans les chapitres V et VI de cette thèse, comment on peut utiliser les algorithmes, PSO et BBO pour ajuster les paramètres du filtre de Kalman étendu afin d'améliorer la performance d'estimation des états du système à commander.

Nous avons consacré le chapitre III d'une part aux quelques rappels sur les propriétés générales des systèmes non-linéaires, nous avons mis le point sur quelques concepts relatifs à la stabilité des systèmes non-linéaires (stabilité au sens de Lyapunov). Puis, nous avons exposé un rappel sur la modélisation, représentation entrée-sortie des systèmes non-linéaires. Enfin, nous avons présenté les différentes méthodes les plus utilisées pour la commande des systèmes non-linéaires telle que : la commande linéarisante, mode glissant, et backstepping. D'autre part, nous avons présenté deux approches très étudiées dans la littérature, concernant l'estimation d'état des systèmes non-linéaires, telles que l'observateur à mode glissant et le filtre de Kalman étendu.

Dans le chapitre IV, après une brève présentation du contexte historique et opérationnel, nous avons exposé l'état de l'art sur les différentes stratégies de commande mises en œuvre pour remplir les différentes missions du quadrirotor. Ensuite, nous avons proposé une loi de commande robuste pour la commande d'altitude et d'attitude d'un quadrirotor via backstepping-mode-glissant floue type-1. Cette structure de commande est capable d'assurer de bonnes performances de poursuite et de garantir la stabilité du système en boucle fermée. L'approche proposée combine trois termes, le premier est le mode glissant qui sert à stabiliser la dynamique du quadrirotor, le deuxième utilise la technique backstepping pour synthétiser la stabilité de la commande par mode glissant, le troisième est un système flou type-1 pour adapter le gain de commutation pour réduire le phénomène de broutement. La robustesse vis-à-vis des perturbations externes élevées est assurée par la commande backstepping-mode-glissant. Les résultats de simulation obtenus montrent de bonnes performances de poursuite et de robustesse, et mettent en évidence la capacité et l'efficacité du système flou développé.

Dans le chapitre V, nous avons proposé une loi de commande pour une classe de systèmes non-linéaires. La loi de commande proposée comprend une loi de commande linéarisante classique, augmentée d'une composante de robustification par mode de glissement. La composante de robustification a été introduite pour assurer la robustesse de la structure de commande, par rapport aux effets des variations paramétriques et les perturbations externes. La combinaison de la commande linéarisante avec la composante de robustification, montre une meilleure performance. La composante de robustification a été renforcée également par un système flou type-1 pour réduire le phénomène de broutement, le principe du système flou type-1 est explicité. La structure de base d'un système flou type-1 a été constituée essentiellement d'un fuzzificateur, d'une base de règles, d'un moteur d'inférence floue et d’un defuzzificateur. L’observateur EKF a été introduit pour estimer les états du système. La performance d’estimation obtenue avec l'observateur EKF est également comparée à celle obtenue par l’observateur SMO. L’ajustement des paramètres de l’estimateur EKF a été fait par l'ajustement des matrices de covariance Q et R via l’algorithme d’optimisation PSO, la performance obtenue a été comparée avec la technique GA. On conclut que la performance de la commande proposée combinée avec le processus PSO-EKF donne des résultats plus précis que ceux obtenus par l’approche GAs. Des simulations sont alors présentées pour illustrer d’une part la robustesse de la commande proposée et d’autre part les performances de l’observateur optimisé, pour les systèmes non-linéaires en présence de variations paramétriques et les perturbations externes simultanément.

Dans le chapitre VI, en considérant les incertitudes des paramètres et de perturbations externes, nous avons proposé une nouvelle application de l’algorithme d'optimisation basée sur la biogéographie, pour optimiser le filtre de Kalman étendu. La commande par mode glissant a été combinée avec un système flou type-2 intervalle, pour assurer une bonne robustesse. La performance d’EKF a été améliorée en ajustant les paramètres des matrices de covariance Q et R, dans lesquelles l'algorithme BBO est utilisé, et il a été comparé à la technique PSO. La méthode d'optimisation proposée permet de sélectionner les matrices de covariance optimale Q et R. Les résultats de simulation confirment que les performances de la commande floue type-2 intervalle par mode glissant combinée avec le processus BBO-EKF étaient meilleures que celle où elle a été combinée avec le processus PSO-EKF.

Perspectives

Le travail réalisé au cours de cette thèse ouvre un certain nombre de perspectives. Parmi les problèmes qui n’ont pas été abordés ici d’une façon détaillée et qui peuvent faire l’objet d’une recherche future, on peut citer principalement deux directions. Une première voie de recherche possible est l'amélioration de la loi de commande. Une seconde, consiste à améliorer la performance de l’observateur.

Amélioration de la loi de commande

Plusieurs étapes de la loi de commande synthétisée décrite dans les trois derniers chapitres, peuvent certainement être améliorées. Par exemple, lors de la définition du système flou, de nombreuses caractéristiques doivent être choisies de façon intuitive : nombre de termes linguistiques qualifiant les entrées ou les sorties, nombre de des règles floues, etc. Nous proposons qu'il faudrait optimiser simultanément, tous les paramètres réglant le fonctionnement du système flou (fonctions d'appartenance, règles floues, défuzzification).

- La mise en œuvre pratique de ces résultats serait une autre direction intéressante pour de futures recherches. Cela pourrait aider à vérifier les résultats de la simulation.
- Nous envisageons la validation expérimentale des approches proposées.

Enfin, les approches proposées pourraient être explorées pour d'autres systèmes. Cela pourrait montrer l'efficacité des approches dans différents domaines et démontrer la flexibilité des approches.

Amélioration de la performance de l'observation

Hybridation des techniques de l'intelligence artificielle avec les processus d'optimisation PSO-EKF et BBO-EKF pour pouvoir améliorer la performance de l'observateur EKF pour l'estimation d'état.

Bibliographie

Bibliographies

[1] A. Isidori, *Nonlinear Control Systems - An Introduction*, Berlin: 2nd ed. Springer-Verlag, 1989.

[2] H. K. Khalil, *Nonlinear Systems*. Macmillan, New York, 2002.

[3] J.-J. E. Slotine, W. Li, *Applied Nonlinear Control*. London: Prentice-Hall, Inc., 1991. ISBN: 0130408905.

[4] V. I. Utkin, and H. Lee, "Chattering problem in Sliding Modes Control Systems," IFAC Proceedings Volumes, vol. 39, iss. 5, pp. 1, 2006.

[5] J.-J. E. Slotine, "Sliding controller design for nonlinear systems," *Int. J. of Control*, vol. 40, no. 2, pp. 421-434, 1984.

[6] A. G. Bondarev, S. A. Bondarev, N. E. Kosteleva, and V. I. Utkin, "Sliding Modes in Systems with Asymptotic State Observers," *Automation and remote control*, vol. 46, no. 6, pp. 49–64, 1985.

[7] S. V. Emel'yanov, S. V. Korovin, and L. V. Levantovsky, "Higher Order Sliding Modes in the Binary Control System," *Soviet Physics*, vol. 31, no. 04, pp. 291–293, 1986.

[8] V. Bregeault, "Quelques contributions a la theorie de la commande par modes glissants," PhD thesis, Ecole Centrale de Nantes, 2010.

[9] Y. Tang, "Terminal sliding mode control for rigid robots," *Automatica*, vol. 34, no. 1, pp. 51–56, 1998.

[10] A. Hamzaoui, N. Essounbouli, and J. Zaytoon, "Fuzzy sliding mode control for uncertain siso systems,"In IFAC Conf. on Intelligent Control Systems and Signal Processing, Faro, Portugal, April 2003. Elsevier-Oxford, pp. 233–238.

[11a] L. A. Zadeh, "Fuzzy logic, neural networks and soft computing," *Comm. A CM*, vol. 3, pp. 77-84, 1994.

[11b] L.A. Zadeh. "Soft computing and fuzzy logic," *IEEE Software*, vol. 11, no. 6, pp. 48-56, 1994.

[12] C. H. Chen, *Fuzzy logic and neural network handbook*, McGraw-Hill, 1996.

[13] J. Liu. *Monte Carlo Strategies in Scientic Computing*. Springer, New York, 2001.

[14] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 535–550, 2000.

[15] J.M. Mendel, *Uncertain rule-based fuzzy logic systems: introduction and new direction*, Prentice-Hall, Upper Saddle River, NJ, 2001.

[16] J.M. Mendel, "A quantitative comparison of interval type-2 and type-1 fuzzy logic systems: First results," In IEEE Int. Conf. Fuzzy Systems, 2010, pp. 1–8.

[17] H. F. Ho, Y. K. Wong, and A. B. Rad, "Adaptive Fuzzy Sliding Mode Control with Chattering Elimination for Nonlinear SISO Systems," Simulation Modeling Practice and Theory, vol. 17, pp. 1199-1210, 2009.

[18] N. Noroozi, M. Roopaei, and M. Z. Jahromi, " Adaptive fuzzy sliding mode control scheme for uncertain systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 11, pp. 3978–3992, 2009.

[19] S. Zeghlache, K. Kara, and D. Saigaa, "Fault Tolerant Control Based on Interval Type-2

Fuzzy Sliding Mode Controller for Coaxial Trirotor Aircraft," *ISA Trans*., vol. 51, no. 11, pp. 215-231, 2015.

[20] L.-T. Ren, S.-S. Xie, Z.-G. Miao, H.-S. Tian and J.-B. Peng. "Fuzzy robust sliding mode control of a class of uncertain systems," *Springer, Journal of Central South University*, vol. 23, iss. 9, pp. 2296-2304, 2016.

[21] P. Mercader, K. Soltesz, and A. Baños, "Robust PID design by chance-constrained optimization," *ISA Transactions*, vol. 354, no. 18, pp. 8217–8231, 2017.

[22] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

[23] R.R. Yager, S. Ovchinnikov, R.M. Tong, et H.T. Nouyen, "Fuzzy Sets and Application," Selected Paper by L.A. Zadeh. J. Wiley, New York, 1987.

[24] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and stability analysis*, NJ: Prentice-Hall Englewood Cliffs, 1994.

[25] E. H. Mamdani, and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man Mach. Studies*, vol. 7, no. 1, pp. 1-13, 1975.

[26] T. Tamakawa, "High speed fuzzy controller hardware system," Proc. 2nd Fuzzy System Symp., 1986, pp. 122-130.

[27] J. H. Park, S. Kim, and C. Moon, "Adaptive fuzzy controller for the nonlinear systems with unknown sign of the input gain," *International Journal of Control Automation and System*s, vol. 4, pp. 178-186. 2006.

[28] M. C. Tanaka, J. M. M. Fernandes, and W. M. Bessa, "Feedback linearization with fuzzy compensation for uncertain nonlinear systems," *Int. J. Comput.Commun*. vol. 8, no. 5, pp. 736–743, 2013. ISSN 1841-9836.

[29] M. H. Khooban, "Design an intelligent proportional derivative pd feedback linearization control for nonholonomic-wheeled mobile robot," *J. Intelligent & Fuzzy Systems*, vol. 26, no. 4, pp.1833–1843, 2014.

[30] B. Robyns, F. Berthereau, J.-P. Hautier et H. Buyse, "A fuzzy-logic-based multimodel field orientation in an indirect foc of an induction motor," *IEEE Trans. on Industrial Electronics*, vol. 47, pp. 380–388, 2000.

[31] L. Leclercq, B. Robyns et J. Grave, "Control based on fuzzy logic of a flywheel energy storage system associated with wind and diesel generators," *Mathematics and Computers in Simulation archive – Special issue: Modelling and simulation of electrical machines, converters and systems*, vol. 63, pp. 271–280, 2003.

[32] H. F. Ho, Y. K. Wong, and A. B. Rad, "Adaptive Fuzzy Sliding Mode Control with Chattering Elimination for Nonlinear SISO Systems," *Simulation Modeling Practice and Theory*, vol. 17, pp. 1199-1210, 2009.

[33] F. Piltan, et al., "Design novel nonlinear controller applied to robot manipulator: Design new feedback linearization fuzzy controller with minimum rule base tuning method," *Int. J. Rob. Auto. (IJRA)*, vol. 3, pp. 1–12, 2012.

[34] N. Wang, S. Tong, and Y. Li, "Observer-based adaptive fuzzy dynamic surface control of non-linear non-strict feedback system," *IET Control Theory Applications*, vol. 11, no. 17, pp. 3115–3121, 2017.

[35] A. Sarabakha, C. Fu, E. Kayacan and T. Kumbasar," Type-2 Fuzzy Logic Controllers Made Even Simpler: From Design to Deployment for UAVs," *IEEE Trans. Ind. Electron.*, vol. 56, iss. 6, pp. 5069–5077, 2018.

[36] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-I," Information Sciences, vol. 8, no. 3, pp. 199–249, 1975.

[37] N. N. Karnik and J. M. Mendel, "introduction to type-2 fuzzy logic systems," In: Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence, pp. 915–920, vol. 2, 1998.

[38] J. M. Mendel, "Fuzzy systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, no. 3, pp. 347-377, Mar. 1995.

[39] K.M. Passino, and S. Yurkovich, *Fuzzy control*. Addison-Wesley Longman Inc, 1998.

[40] B. Bouchon-Meunier, *La logique floue et ses applications*. Addison Wesley, Paris, 1995.

[41] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its applications to modelling and control," *IEEE trans. syst., man, cybern.*, vol. SMC-15, no. 1, 1985.

[42] J. Godjevac. Idées nettes sur la logique floue. Presses polytechniques et universitaires

romandes, Lausannel, 1999.
[43] A. Idri, and A. Abran. La logique floue appliquée aux modèles d'estimation d'effort de développement de logiciels-cas du modèle cocomo'81. Presses polytechniques et universitaires romandes, Lausannel, 2004.
[44] H. Buhler, *Le réglage par logique floue*. Presses Polytechniques Romandes, 1994.
[45] D. Driankov, H. Hellendorn, and R. Palm. *Model based fuzzy control*, In Springer-Verlag, New York, 1996.
[46] S. Labiod, "Commande adaptative par systèmes flous: application aux robots manipulateurs," Magister thesis, Université de Jijel, Algérie, 1998.
[47] O. Castillo and P. Melin, *Type-2 fuzzy logic : theory and applications*. Springer Verlag, 2008.
[48] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2002.
[49] D. Wu and J.M. Mendel, "Enhanced karnik–mendel algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.
[50] W. Tfaili, "Conception d'un algorithme de colonie de fourmis pour l'optimisation continue dynamique," Ph.D. thesis, Université Paris 12 Val de Marne UFR de Sciences et Technologie, 2007.
[51] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
[52] K. Iba, "Reactive power optimization by genetic algorithms," *IEEE Trans on power systems*, vol.9, no.2, pp. 685-692, 1994.
[53] R. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory," in Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS '95, Oct. 4–6, 1995, Nagoya, Japan, pp. 39–43, 1995.
[54] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *In Computer Graphics*, vol. 21, pp. 25–34, 1987.
[55] F. Heppner and U. Grenander, "A stochastic nonlinear model for coordinated bird flocks," In E. Krasner, editor, The ubiquity of chaos, pp. 233–238. AAAS Publications, 1990.
[56] D. Simon, "Biogeography-Based Optimization," *IEEE Trans. Evol. Comput*., vol. 12 *no.* 6, pp. 702–713, 2008.
[57] R. MacArthur and E. Wilson, *The Theory of Biogeography*. Princeton University Press, Princeton, NJ, 1967.
[58] R. Ricklefs, and G. Miller, *Ecologie*. éd. de Boeck, 2005. ISBN 2-7445-0145-X.
[59] H. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization," *Information Sciences*, vol. 180, no. 18, pp. 3444–3464, 2010.
[60] T. Bäck, *Evolutionary algorithms in theory and practice - evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, 1996. ISBN 978-0-19-509971-3.
[61] P. Ioannou, and J. Sun, *Robust Adaptive Control*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1996.
[62] A. Isidori, *Nonlinear control systems*, Springer Verlag, 1995.
[63] V. I. Utkin, *Sliding modes and their applications in variable structure systems*, MIR Publ, Moscow, 1978.
[64] A. Levant, "Sliding order and sliding accuracy in sliding mode control," Int. J. Control, vol. 58, pp. 1247–1263, 1993.
[65] M.C.S. Alaoui, "Commande et observateur par modes glissants d'un système de pompage et d'un bras manipulateur," Ph.D. thesis, Université de Sidi Mohammed ben Abdellah, Maroc, 2009.
[66] J.-J. E. Slotine, and S.S. Sastry, "Tracking Control of Nonlinear System using Sliding Surface, with Application to Robotic Manipulators," *Int. J. Control*, vol. 38, pp. 465–492, 1983.
[67] V. I. Utkin, *Sliding Modes in Control Optimization*, Springer-Verlag, 1992.
[68] I. Kanellakopoulos, P. V. Kokotovic, A. S. Morse, "systematic design of adaptive controllers for feedback linearizable systems," *IEEE Transactions on Automatic*

Control, vol. 36, no. 11, pp.1241-1253, 1991.

[69] A. Feurer , and A. S. Mors, “Adaptive control of single-input, single-output linear systems,” *IEEE Transactions on Automatic Control*, vol. 23, no. 4, pp. 557-569, 1978.

[70] J. Tsinias “Sufficient lyapunov-like conditions for stabilization,” Math. Contr. Signal. Syst. Vol. 2, pp. 343-357, 1989.

[71] P. V. Kokotovic, H. J. Sussmann, “A positive real condition for global stabilization of nonlinear systems,” *Systems and Control Letters*, vol. 13, pp. 125-133, 1989.

[72] B. L. Walcott, M. J. Corless, and S. H. Żak, “Comparative study of non-linear state-observation techniques. *Int. J. Contr*., vol. 45, pp. 2109–2132, 1987.

[73] E. A. Misawa and J. K. Hedrick, “Nonlinear observers : a state-of-the-art survey,” *Trans. of the ASME, J. Dyn. Syst. , Meas. & Cont*., vol. 111, no. 3, pp. 344–352, 1989.

[74] T. O. Kowalska, “Application of extended luenberger observer for flux and rotor time-constant estimation in induction motor drives. *IEE Proc. Control Theory Appl*., vol. 36, pp. 324–330, 1989.

[75] C. C. Witt and J. J. E. Slotine, “Sliding observers for robot manipulator,” *Automatica*, vol. 27, no. 5, pp. 859–864, 1991.

[76] B. Karanayil, M. F. Rahman, and C. Grantham, “Online stator and rotor resistance estimation scheme using artificial neural networks for vector controlled speed sensorless induction motor drive,” *IEEE Trans. Indus. Electron.* vol. 54, pp. 167–176, 2007.

[77] X-J. Ma, Z. Sun, and Y-Y. He, “Analysis and design of fuzzy controller and fuzzy observer,” IEEE 12th symposium on computational intelligence and informatics (CINTI), Budapest, Hungary, pp. 45–50, 2011.

[78] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *ASME Trans.- Part D, J. Basic Engineering*, vol. 82, pp. 34–45, 1960.

[79] R. E. Kalman and R. S. Bucy, “New results in linear filtering and prediction theory,” *ASME Trans. - Part D, J. Basic Engineering*, vol. 83, pp. 95–108, 1961.

[80] D. G. Luenberger, “Observing the state of a linear system,” *IEEE Transaction on Military Electronics*, vol. 8, pp.74–80, 1964.

[81] Y. Xiong, and M. Saif, “Sliding mode observer for nonlinear uncertain systems,” *IEEE Trans. on Automatic Control*, vol. 46, no. 12, pp. 2012-2017, 2001. DOI: 10.1109/9.975511

[82] S. V. Drakunov, “Sliding mode observer based on equivalent control method,” In IEEE Conf. Decis. Control, 1992, pp. 2368–2369.

[83] S. V. Drakunov and V. Utkin, “Sliding mode observer. Tutorial,” IEEE Conf. Decis. Control, 1995, pp. 3376–3379.

[84] Y. Song and J. W. Grizzle, “The extended Kalman filter as a local asymptotic observer for nonlinear discrete-time systems,” In American Control Conference, IEEE, 1992, pp. 3365–3369.

[85] C. Manes, F. Parasiliti, and M. Tursin, “A comparative study of rotor flux estimation in induction motors with a nonlinear observer and the extended Kalman filter,” Proc. IEEE IECON’94, 1994, pp. 2149–2154.

[86] M. M. Olama, S. M. Djouadi, I. G. Papageorgiou, and C. D. Charalambous, “Position and velocity tracking in mobile networks using particle and Kalman filtering with comparison,” *IEEE Trans. on Veh. Technol*,, vol. 57, no. 02, pp. 1001–1010, 2008.

[87] Y. Laamari, K. Chafaa, B. Athamena, “Particle swarm optimization of an extended Kalman filter for speed and rotor flux estimation of an induction motor drive,” *Electrical Engineering, Archivfür Elektrotechnik, Springer-Verlag Berlin Heidelberg*, vol. 97, pp. 129-138, 2014.

[88] S. Bouabdallah, and R. Siegwart, “Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor,” in Proc. IEEE Int. Conf. on Robot. and Autom., Barcelona, Spain, 2005, pp. 2247-2252.

[89] A. Tayebi, and S. Mc. Gilvray, “Attitude stabilization of a VTOL Quadrotor Aircraft,” *IEEE Trans. on Control Syst. Technol.*, vol. 14, no. 3, pp. 562-571, 2006.

[90] S. Bouabdallah, “Design and control of quadrotors with application to autonomous flying,” Ph.D. dissertation, EPFL, 2007.

[91] R. R. Murphy, E. Steimle, C. Gri-n, C. Cullins, M. Hall, and K. Pratt, “Cooperative use

of unmanned sea surface and micro aerial vehicles at Hurricane Wilma," *Journal of Field Robotics*, vol. 25, no. 3, pp. 164-180, March 2008.

[92] L. Wallace, A. Lucieer, C. Watson, and D. Turner, "Development of a UAV-LiDAR System with Application to Forest Inventory," *Remote Sensing*, vol. 4, no. 12, pp. 1519-1543, May 2012.

[93] F. I. Petrescu, *Near the Flying Time*, Lulu.com, 2011.

[94] A. Rango, A. Laliberte, J. E. Herrick, C. Winters, K. Havstad, C. Steele, D. Browning, "Unmanned aerial vehicle-based remote sensing for rangeland assessment, monitoring, and management," *Journal of Applied Remote Sensing*, 3: 033542–033515. 2009.

[95] Convervation Drones.org., 2017. http://conservationdrones.org/.

[96] K. Anderson, K. J. Gaston, "Lightweight unmanned aerial vehicles will revolutionize spatial ecology," *Frontiers in Ecology and the Environment*, vol. 11, pp. 138–146. 2013.

[97] J. L. Morgan, S. E. Gergel, N. C. Coops, "Aerial photography: A rapidly evolving tool for ecological management," *BioScience*, vol. 60, pp. 47–59. 2010.

[98] CRASAR, Center for Robot Assisted Search and Rescue, Texas A&M University, 2013. http://crasar.org/disasters/.

[99] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, "Modeling and control of a flying robot for contact inspection," In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2012, pp. 3532-3537.

[100] M. Wojciech, "Robotized Inspection and Diagnostics – Basic Issues," in Advanced Solutions in Diagnostics and Fault Tolerant Control, 2017, pp. 131–142.

[101] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, "Unmanned Aircraft Systems in Remote Sensing and Scientific Research: Classification and Considerations of Use," *Remote Sensing*, vol. 4, no. 12, pp.1671-1692, June 2012.

[102] M. Brutto, A. Borruso, and A. D'Argenio, "UAV Systems for Photogrammetric Data Acquisition of Archaeological Sites," *International Journal of Heritage in the Digital Era*, vol. 1, no. 0, pp. 7-14, January 2012.

[103] C. Zhang, and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: a review," *Precision Agriculture*, vol. 13, no. 6, pp. 693-712, July 2012.

[104] DroneJournalisme.org, 2017. http://www.dronejournalism.org/.

[105] A. S. Sniderman, and M. Hanis, *Drones for Human Rights*. New York Times, 2012.

[106] Matternet Web Site, 2017. http://matternet.us/.

[107] K. Alexis, G. Nikolakopoulos, A. Tzes, "Model predictive quadrotor control: attitude, altitude and position experimental studies," *IET Control Theory Appl.*, vol. 6, no. 12, pp. 1812–1827, 2012.

[108] P. Castillo, R. Lozano, and A. Dzul, "Stability of a mini rotorcraft with four rotors," *IEEE Control Syst. Magazines*, vol. 25, no. 6, pp. 45–55, 2005.

[109] A. Das, K. Subbarao, and F. Lewis, "Dynamic inversion with zero-dynamics stabilisation for quadrotor control," *IET Control Theory and Appl.*, vol. 3, no. 3, pp. 303–314, 2008.

[110] J.-J. Xiong, E.-H. Zheng, "Position and attitude tracking control for a quadrotor UAV," *ISA Trans.*, vol. 53, no. 3, pp. 725-31, 2014.

[111] L. Chen, and G. Wang, "Attitude Stabilization for a Quadrotor Helicopter Using a PD Controller," 3rd IFAC Int. Conf. on Intell. Control and Autom. Sci., Chengdu, China, 2013, pp. 236–239.

[112] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor," in Proc. IEEE Int. Conf. on Intell. Robots and Syst. (IROS), Sendai, Japan, 2004, pp. 2451 – 2456.

[113] T. Nuchkrua, and M. Parnichkun, "Identification and optimal control of Quadrotor," *Thammasat Int. J. Sci. Technol.*, vol. 17, no. 4, pp. 36-54, 2012.

[114] M. Hassan, *et al*., "Stabilized controller design for attitude and altitude controlling of Quad-rotor under disturbance and noisy conditions," *American Journal of Applied Sci.*, vol. 10, no. 8, pp. 819-831, 2013.

[115] A. Mokhtari, A. Benallegue, and Y. Orlov, "Exact linearization and sliding-mode

observer for a quadrotor unmanned aerial vehicle," *Int. J. Robot. Autom.*, vol. 21, no. 1, pp. 39–49, 2006.

[116] M.Y. Amir, and V. Abbas, "Modeling and Neural Control of Quadrotor Helicopter," *Yanbu J. Eng. and Sci.*, vol. 2, pp. 35-49, 2011.

[117] C. Coza, and C. J. B. Macnab, "A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization," NAFIPS 2006, IEEE Xplore, 2006, pp. 454-458.

[118] G. Limnaios, and N. Tsourveloudis, "Fuzzy Logic Controller for a Mini Coaxial Indoor Helicopter," *Springer, Journal of Intell. & Robotic Syst.*, vol. 65, no. 1, pp. 187–201, 2012.

[119] CH. Hong, KC. Choi et BS. Kim, "Applications of adaptive neural network control to an unmanned airship," *Int. J. Control, Autom. Syst.*, vol. 7, iss. 6, pp. 911, 2009.

[120] M. Lower, W. Tarnawski, "Quadrotor Navigation Using the PID and Neural Network Controller," In: W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, J. Kacprzyk (eds) Theory and Engineering of Complex Systems and Dependability. *Advances in Intelligent Systems and Computing*, vol 365. Springer, Cham, 2015, pp. 265–274.

[121] I. Fantoni, and R. Lozano, *Non-linear control for underactuated mechanical systems*, Springer, 2002.

[122a] T. Madani, and A. Benallegue, "Backstepping Control for a Quadrotor Helicopter," in Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst., Beijing, China, 2006, pp. 3255–3260.

[122b] T. Madani, and A. Benallegue, "Control of a quadrotor mini-helicopter via full state backstepping technique," Proceedings of the 45th IEEE Conference on Decision and Control, 2006, pp. 1515-1520.

[123] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," *IET Control* Theory *Appl.*, vol. 4, no. 11, pp. 2343–2355, 2010.

[124] D.V. Rao, and T.H. Go, "Automatic landing system design using sliding mode control," *Aerosp. Sci. Technol*., vol. 32, no. 1, pp. 180–187, 2014.

[125] M. Reinoso, and L. I. Minchala, "Trajectory tracking of a quadrotor using sliding mode control," *IEEE Latin America Trans.*, vol. 14, no. 5, pp. 2157-2166, 2016.

[126] R. Xu, and Ü. Özgüner, "Sliding Mode Control of a Quadrotor Helicopter," *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, 2006, pp. 4957-4962.

[127] R. Xu, Ü. Özgüner, "Sliding mode control of a class of underactuated systems," *In Automatica*, vol. 44, Iss. 1, pp. 233-241, 2008.

[128] T. Li, Y. Zhang, and B. W. Gordon, "Nonlinear Fault-Tolerant Control of a Quadrotor UAV Based on Sliding Mode Control Technique," *In IFAC Proceedings Volumes*, vol. 45, iss. 20, pp.1317-1322, 2012.

[129] M. J. Reinoso, L. I. Minchala, P. Ortiz, D. F. Astudillo, and D. Verdugo, "Trajectory tracking of a quadrotor using sliding mode control," in *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2157-2166, 2016.

[130] H. Boudjedi, F. Yacef, O. Bouhali, and N. Rizoug, "Dual neural network for adaptive sliding mode control of quadrotor helicopter stabilization," *Int. J. Inf. Sci. and Techn.*, vol. 2, no. 4, pp. 1–14, 2012.

[131] H.-S. Kang, *et al*., "Design of sliding-mode control based on fuzzy disturbance observer for minimization of switching gain and chattering," *Soft Computing*, vol. 19, no. 4, pp. 851–858, 2015.

[132] E.-H. Zheng, J.-J. Xiong, and J.-L. Luo, "Second order sliding mode control for a quadrotor UAV," *ISA Trans.*, vol. 53, pp. 1350–1356, 2014.

[133] B. Abci, G. Zheng, D. Efimov, M.E.B. El Najjar, "Integral backstepping sliding mode control for quadrotor helicopter under external uncertain disturbances," *Aerospace Science and Technology*, vol. 68, pp. 299–307, 2009.

[134] A. Medjghou, N. Slimane and K. Chafaa. "Fuzzy sliding mode control based on backstepping synthesis for unmanned quadrotors," *Advances in Electrical and Electronic Engineering*, vol. 16, iss. 2, pp. 135–146, 2018.

[135] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," *IEEE Trans. Control Syst. Technol.*, vol. 12, iss. 4, pp. 510–516, 2004.

[136] L. Derafa, T. Madani, and A. Benallegue, “Dynamic modelling and experimental identification of four rotor helicopter parameters,” in *IEEE-ICIT*, Mumbai, India, 2006, pp. 1834–1839.

[137] L. Derafa, A. Ouldali, T. Madani, and A. Benallegue, “Four Rotors Helicopter Yaw and Altitude Stabilization,” Proceedings of the World Congress on Engineering 2007, vol I. London, UK, 2007. ISBN:978-988-98671-5-7.

[138] H. Khebbache, “Robust Control Algorithm Considering the Actuator Faults for Attitude Tracking of an UAV Quadrotor Aircraft,” *Int. J. Control Autom.*, vol. 5, iss. 4, pp. 55–66, 2012.

[139] A. Medjghou, A. Soltani, L. Abdou, “Design of a Sliding Mode Control for a Mini-drone Quadrotor,” International Conference of Modeling and Simulation (ICMS’2014), Blida, Algeria, 2014, pp. 25-36.

[140] A. L. D. Franco, et al., “Robust nonlinear control associating robust feedback linearization and H_∞ control,” *IEEE Transactions on Automatic Control***,** vol. 51, no. 7, pp. 1200–1207, 2006.

[141] C-J. Huang, J. Ohri, “*Fuzzy feedback linearization control for mimo nonlinear system and its application to full-vehicle suspension system,” Springer, J. Circ. Syst. Sig. Proc.*; vol. 28, pp. 959–991, 2009.

[142] L. Wang, “The feedback linearization based on backstepping technique,” IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009, pp. 282–286.

[143] J. O. Pedro, and O. A. Dahunsi, “Neural based feedback linearization control of a servo-hydraulic vehicle suspension system,” *Int. J. Appl. Math. Comput. Sci.*, vol. 21, no. 1, pp. 137–147, 2011.

[144] C-C. Fuh, H-H. Tsai, and W-H. Yao, “Combining a feedback linearization controller with a disturbance observer to control a chaotic system under external excitation,” Elsevier, Communications in Nonlinear Science and Numerical Simulation, pp. 1423–1429, 2012.

[145] S. Şahin, “Learning feedback linearization using artificial neural networks,” *Int. J. Neural Process. Lett., vol. 44, no. 3,pp.* 625–637, 2016.

[146] K. D. Do, and J. Pan, “Global tracking control of underactuated ships with nonzero off-diagonal terms in their system matrices,” *Automatica*, vol. 41, no. 1, pp. 87–95, January 2005.

[147] J. W. Helton, and M. R. James, “Extending H_∞ Control to Nonlinear Systems,” *Bulletin of the American Mathematical Society*, vol. 38, no. 1, pp. 93–96, October, 2000.

[148] W. Chang, “Adaptive fuzzy-based tracking control for nonlinear SISO systems via VSS and H_∞ approaches,” *IEEE Trans. Fuzzy Syst.*, vol. 9, pp. 278-292. 2001.

[149] S. Azizi, “Sufficient LMI conditions and Lyapunov redesign for the robust stability of a class of feedback linearized dynamical systems,” *ISA Transactions*, vol. 68, pp. 90–98, 2017.

[150] T. Du, *et al.*, “Application of Kalman Filters and Extended Luenberger Observers to Induction Motor Drives,” Proc. Intelligent Motion conference, pp. 369–386, 1994.

[151] K. L. Shi, et al., “Speed estimation of an induction motor drive using an optimized extended Kalman filter,” *IEEE Trans. Ind. Electron.*, vol. 49, pp. 124–133, 2002.

[152] M. Barut, S. Bogosyan, and M. Gokasan, “Speed-sensorless estimation for induction motors using extended Kalman filters,” *IEEE Trans. Ind. Electron.*, vol. 54, pp. 272-280, 2007.

[153] C. Biçer; L. Özbek; H. Erbay, “Performance and stochastic stability of the adaptive fading extended Kalman filter with the matrix forgetting factor,” vol. 14, no. 1, pp. 934-945, 2016.

[154] Y. Agrebi, et al., “Rotor speed estimation for indirect stator flux oriented induction motor drive based on MRAS scheme,” *J. Electr. Syst.* vol. 3, pp. 131–143, 2007.

[155] R. Shahnazi, Mohammad-R. Akbarzadeh-T, “PI Adaptive Fuzzy Control With Large and Fast Disturbance Rejection for a Class of Uncertain Nonlinear Systems,” *IEEE Trans. on Fuzzy Systems*, vol. 16, no. 1, pp. 187-197, 2008.

[156] P. Vas, “Ensorless Vector and Direct Torque Control,” Oxford University Press.

London: ISBN 978-0198564652, 0198564651, 1998.

[157] S. Bolognani, R. Oboe, and M. Zigliotto, “Sensorless full-digital PMSM drive with EKF estimation of speed and rotor position,” *IEEE Trans. Ind. Electron.* vol. 46, pp. 184–191, 1999.

[158] Z. Liang, Y. Yang, and Y. Zeng, “Eliciting compact T-S fuzzy models using subtractive clustering and coevolutionary particle swarm optimization,” *Neuro-computing*, vol. 72, no. 10-12, pp.2569–2575. 2009.

[159] S. S. Sastry, *Nonlinear systems: analysis, stability and control*, Springer Verlag, 1999. ISBN: 0-387-98513-1, New York, USA

[160] H. Sira-Ramirez, “Structure at infinity, zero dynamics and normal forms of systems undergoing sliding motion,” *International Journal of Systems Science*, vol. 21, no. 4, pp. 665–674, 1990.

[161] G. Bartolini, P. Pydynowski, “Asymptotic linearization of uncertain nonlinear systems by means of continuous control,” *International Journal of Robust and Nonlinear*, vol. 3, no. 2, pp. 87–103, 1993.

[162] G. Bartolini, A. Ferrara, E. Usani, “Chattering avoidance by second-order sliding mode control,” *IEEE Transactions on Automatic Control*, vol. 43, no. 2, pp. 241–246, 1998.

[163] F. Cupertino, D. Naso, E. Mininno, and B. Turchiano, “Sliding-Mode Control With Double Boundary Layer for Robust Compensation of Payload Mass and Friction in Linear Motors,” *IEEE Trans. Ind. Appl.*, vol. 45, no. 5, pp. 1688–1696, 2009.

[164] G. Herrmann, S. K. Spurgeon, C. Edwards, “On Robust, Multi-Input Sliding-Mode Based Control with a State-Dependent Boundary Layer,” *Journal of Optimization Theory and Applications*, vol. 129, no. 1, pp. 89–107, 2016.

[165] A. Tayebi, “Adaptive iterative learning control for robot manipulators,” *Elsevier, Automatica*, vol. 40, no. 7, pp. 1195–1203, 2004.

[166] N. Kapoor and J. Ohri, “Fuzzy Sliding Mode Controller (FSMC) with Global Stabilization and Saturation Function for Tracking Control of a Robotic Manipulator,” *J. Control Syst. Eng.*, vol. 1, pp. 50–56, 2013.

[167] M. R. Soltanpour *et al.* , “Fuzzy Sliding Mode Control Design for a Class of Nonlinear Systems with Structured and Unstructured Uncertainties,” *Int. J. Innov. Comput. Inform. Control*, vol. 9, no. 7, pp. 2713–2726, 2013.

[168] F. Baklouti, S. Aloui, and A. Chaari, “Fuzzy sliding mode fault tolerant control for a class of perturbed nonlinear systems,” *Int. J. Model. Identif.* Control, vol. 25, no. 4, pp. 313–322, 2016.

[169] R. Naoual *et al.*, “Fuzzy sliding mode control for the two-link robot,” *Int. J. Syst. Control Commun.*, vol. 6, no. 1, pp. 84–96, 2014.

[170] L. Chen, *et al.*, “Sliding mode adaptive control for DC motors using function approximation form,” *Int. J. Model. Identif. Control*, vol. 26, no. 3, pp. 238–252, 2016.

[171] A. Medjghou, M. Ghanai, N. Bounouara, and K. Chafaa, “Fuzzy switch-gain sliding mode control for robotic manipulators based on an extended Kalman filter,” in Proc. 8th Int. Conf. on Modelling, Identification and Control, IEEE Xplore, Algers, Algeria, 2016, pp. 198–206.

[172] N. N. Karnik, J. M. Mendel, and Q. Liang, “Type-2 fuzzy logic systems,” *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, 643–658, 1999.

[173] M. Biglarbegian, W. Melek, and J. M. Mendel, “On the robustness of Type-1 and Interval Type-2 fuzzy logic systems in modeling,” *Expert Systems with Applications, Elsevir*, vol. 73, pp. 161–17, 2017.

[174] G. Hasanifard, A. A. Gharaveisi, and M. A. Vali, “Fuzzy modeling for chaotic systems via interval type-2 T-S fuzzy model with parametric uncertainty,” *J. Theoret. Appl. Phys.*, vol. 8, no. 1, pp. 115, 2014.

[175] O. Castillo, *Type-2 fuzzy logic in intelligent control applications*. Springer-Verlag Berlin Heidelberg, 2012. Doi: 10.1007/978-3-642-24663-0.

[176] G. Zong, Y. Wu, and L. Zhang, “Finite time tracking control for rigid robotic manipulators with friction and external disturbances,” *J. Syst. Sci. Syst. Eng.*, vol. 14, no. 1, pp. 115–125, 2005.

[177] V. Utkin and H. Lee, “Chattering Analysis. Advances in Variable Structure and Sliding

Mode Control," *Springer Verlag: Berlin*, vol. 334, pp. 107–123, 2006.

[178] A. Msaddek, A. Gaaloul, and F. M'sahli, "Adaptive fuzzy supervision of the gain of the higher order sliding mode control," *Int. J. Autom. Control*, vol.9, no. 3, pp. 228–246, 2015.

[179] Y. Koubaà, M. Boukattaya, and T. Damak, "Path tracking control of non-holonomic wheeled mobile robot with skidding and slipping," *Int. J. of Modelling, Identification and Control*, vol. 26, no. 3, pp. 218–226, 2016.

[180] A. Medjghou, M. Ghanai and K. Chafaa. "A Robust Feedback Linearization Control Framework Using an Optimized Extended Kalman Filter," *Journal of Engineering Science and Technology Review*, vol. 10, no. 5, pp. 1–16, 2017.

[181] V. I. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electro-Mechanical Systems*, 2nd ed., CRC Press, Taylor&Francis, 2009. ISBN: 978-1-4200-6560-2.

[182] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[183] J. M. Mendel, R. I. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Trans. Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, 2006.

Annexes

Annexe A.

Paramètres du quadrirotor

Le modèle du quadrotor présenté dans le chapitre IV, représente les différents paramètres utilisés qui sont donnés dans le tableau suivant :

Tableau A.1. Paramètres nominaux du quadrirotor

K_{fa}= diag(5.5670; 5.5670; 6.3540)×10^{-4} N/rad/s		
K_{fd}= diag(0.032; 0.032; 0.048) N/m/s		
J= diag(3.8278; 3.8278; 7.1345)×10^{-3} N.m/rad/s2		
C_p= 2.9842×10^{-5} N/rad/s	C_d= 3.2320×10^{-7} N.m/rad/s	
J_r = 2.8385×10^{-5} N.m/rad/s^2	m = 400 g	l = 20.5 cm
g_a= 9.81	a_1 = -1	a_2 = -0.1454
a_3 = -0.0074	a_4 = 1	a_5 = -0.1454
a_6 = 0.0074	a_7 = -1.3061×10^{-4}	a_8 = -0.0830
a_9 = -0.0011	a_{10} = -0.001	a_{11} = -0.0013
b_1 = 65.3117	b_2 = 65.2946	b_3 = 130.6063

Annexe B.

Matrices Jacobiennes du bras manipulateur

Dans cet annexe nous présentons les matrices Jacobiennes du bras manipulateur utilisées dans le chapitre V

$$F_k = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{bmatrix},\ W_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ et } V_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

où $f_{11} = 1$, $f_{12} = \Delta t$, $f_{13} = 0$, $f_{14} = 0$

$$\begin{aligned} f_{21} = \Delta t\, (((m_2 l_{c2}{}^2 + I_2)\ (g\, S_1 (l_1 m_2 + l_{c1} m_1) + g\, l_{c2} m_2 S_{12}))\, /\, (I_1 I_2 + l_1^2 l_{c2}{}^2\, m_2{}^2 + I_2 l_1^2\, m_2 + I_2 l_{c1}{}^2 m_1 + \\ I_1 l_{c2}{}^2\, m_2 + l_{c1}{}^2 l_{c2}{}^2 m_1 m_2 - l_1^2\, l_{c2}{}^2 m_2{}^2 C_2{}^2) - (g\ l_{c2}\, m_2 S_{12}\, (m_2 l_{c2}{}^2 + l_1 m_2 C_2 l_{c2} + I_2)\ )\, /\ (I_1 I_2 + l_1^2 \times \\ l_{c2}{}^2 m_2{}^2 + I_2 l_1^2 m_2 + I_2 l_{c1}{}^2\, m_1 + I_1 l_{c2}{}^2\, m_2 + l_{c1}{}^2 l_{c2}{}^2 m_1\, m_2 - l_1^2 l_{c2}{}^2 m_2{}^2 C_2{}^2)). \end{aligned}$$

$$\begin{aligned} f_{22} = \Delta t\, \left[(2\, l_1\ l_{c2}\, m_2\, \dot{q}_2\, S_2 (m_2\, l_{c2}{}^2 + I_2))\, /\, (I_1 I_2 + l_1^2\, l_{c2}{}^2\, m_2{}^2 + I_2 l_1^2\, m_2 + I_2 l_{c1}{}^2 m_1 + I_1 l_{c2}{}^2\, m_2 + l_{c1}{}^2 \times \right. \\ l_{c2}{}^2 m_1 m_2 - l_1^2 l_{c2}{}^2 m_2{}^2 C_2{}^2) + (2\, l_1\, l_{c2} m_2 \dot{q}_1 S_2 (m_2 l_{c2}{}^2 + l_1 m_2 C_2 l_{c2} + I_2))\, /\, (I_1 I_2 + l_1^2 l_{c2}{}^2 m_2{}^2 + I_2 l_1^2 \times \\ \left. m_2 + I_2 l_{c1}{}^2 m_1 + I_1 l_{c2}{}^2 m_2 + l_{c1}{}^2 l_{c2}{}^2 m_1 m_2 - l_1^2 l_{c2}{}^2 m_2{}^2 C_2{}^2\,)\right] + 1. \end{aligned}$$

$$\begin{aligned} f_{23} = (\Delta t\ l_1\, l_{c2}\, m_2\, \tau_2\, S_2)\ /\ (I_1 I_2 + l_1^2\, l_{c2}{}^2\, m_2{}^2 + I_2\, l_1^2\, m_2 + I_2 l_{c1}{}^2\, m_1 + I_1\, l_{c2}{}^2\, m_2 + l_{c1}{}^2\, l_{c2}{}^2 m_1\, m_2 - l_1^2 \times \\ l_{c2}{}^2\, m_2{}^2\, C_2{}^2) - \Delta t\ (\ (\ (-l_1\, l_{c2}\, m_2 C_2\ \dot{q}_1{}^2 + g\ \ l_{c2}\, m_2\, S_{12})\ \ (m_2\ \ l_{c2}{}^2 + l_1\, m_2\, C_2\, l_{c2} + I_2)\ \ )\ \ /\ \ (I_1\ \times \\ I_2 + l_1^2\ \ l_{c2}{}^2\, m_2{}^2 +\ I_2\, l_1^2 m_2 + I_2\, l_{c1}{}^2\, m_1 + I_1\, l_{c2}{}^2\ \ m_2\ + l_{c1}{}^2\, l_{c2}{}^2\, m_1\, m_2 - l_1^2\, l_{c2}{}^2\, m_2{}^2\, C_2{}^2) - (\ \ (m_2\ \times \\ l_{c2}{}^2 + I_2)\ (\dot{q}_2\, (\dot{q}_1\, l_1\, l_{c2}\, m_2\, C_2 + \dot{q}_2\, l_1\, l_{c2}\, m_2\, C_2) + g\, l_{c2}\, m_2\, S_{12} + \dot{q}_1 \dot{q}_2\, l_1\, l_{c2}\, m_2 C_2))\, /\, (I_1 I_2 + l_1^2\, l_{c2}{}^2 \times \\ m_2{}^2 + I_2 l_1^2 m_2 + I_2 l_{c1}{}^2 m_1 + I_1 l_{c2}{}^2\, m_2 + l_{c1}{}^2\, l_{c2}{}^2\, m_1\, m_2 - l_1^2\, l_{c2}{}^2\, m_2{}^2 C_2{}^2) +\ \ (l_1\, l_{c2}\, m_2\, S_2\, (l_1\, l_{c2}\, m_2 \times \\ S_2\, x_2{}^2 + g\ \ l_{c2}\, m_2\, C_{12})\ )\ /\ (I_1\, I_2 + l_1^2\, l_{c2}{}^2\, m_2{}^2 + I_2\, l_1^2\, m_2 + I_2\, l_{c1}{}^2 m_1 + I_1\, l_{c2}{}^2 m_2 + l_{c1}{}^2\, l_{c2}{}^2\, m_1\, m_2 - \\ l_1^2\, l_{c2}{}^2 m_2{}^2\, C_2{}^2) + (2\ l_1^2 l_{c2}{}^2 m_2{}^2 C_2\, S_2\, (l_1\, l_{c2}\, m_2\, S_2\, \dot{q}_1{}^2 + g l_{c2} m_2 C_{12})\ (m_2 l_{c2}{}^2 + l_1\, m_2\, C_2 l_{c2} + I_2)\ )\ / \\ (-l_1^2\, l_{c2}{}^2\, m_2{}^2\, C_2{}^2 + l_1^2\, l_{c2}{}^2\, m_2{}^2 + I_2 l_1^2\, m_2 + m_1\, l_{c1}{}^2\, l_{c2}{}^2\, m_2 + I_2\, m_1\, l_{c1}{}^2 + I_1 l_{c2}{}^2\, m_2 + I_1\, I_2)^2 + (2 \times \\ l_1^2\, l_{c2}{}^2\, m_2{}^2\, C_2 S_2\ \ (m_2\, l_{c2}{}^2 I_2)\ (\dot{q}_2\, (\dot{q}_1\, l_1\, l_{c2}\, m_2\, S_2 + \dot{q}_2\, l_1\, l_{c2}\, m_2\, S_2) - g\, C_1 (l_1\, m_2\, l_{c1} + m_1) - g\, l_{c2}\, m_2 \times \\ C_{12} + \dot{q}_1\, \dot{q}_2\, l_1 l_{c2}\, m_2\, S_2)\ \ )\, /\, (-l_1^2\, l_{c2}{}^2\, m_2{}^2 C_2{}^2 + l_1^2 l_{c2}{}^2\ \ m_2{}^2 + I_2\, l_1^2\, m_2 + m_1\, l_{c1}{}^2\, l_{c2}{}^2\, m_2 + I_2\, m_1\, l_{c1}{}^2 + \\ I_1\, l_{c2}{}^2\, m_2 + I_1 I_2)^2\,) - (2\ \Delta t\ \ l_1^2 l_{c2}{}^2\, m_2{}^2\, \tau_1\, C_2\, S_2 (m_2 l_{c2}{}^2 + I_2)\ )\, /\, (-l_1^2\, l_{c2}{}^2\, m_2{}^2\, C_2{}^2 + l_1^2\, l_{c2}{}^2\, m_2{}^2 + \\ I_2 l_1^2 m_2 + m_1 l_{c1}{}^2\, l_{c2}{}^2\, m_2 + I_2 m_1 l_{c1}{}^2 + I_1 l_{c2}{}^2\, m_2 + I_1 I_2)^2 + (2\ \Delta t\ \ l_1^2\, l_{c2}{}^2\, m_2{}^2\, \tau_2\, C_2\, S_2\, (m_2\, l_{c2}{}^2 + l_1 \times \\ m_2 C_2\, l_{c2} + I_2))\, /\, (-l_1^2\, l_{c2}{}^2 m_2{}^2 C_2{}^2 + l_1^2 l_{c2}{}^2 m_2{}^2 + I_2\, l_1^2 m_2 + m_1 l_{c1}{}^2\, l_{c2}{}^2\, m_2 + I_2 m_1\, l_{c1}{}^2 + I_1 l_{c2}{}^2\, m_2\ + \\ I_1 I_2)^2. \end{aligned}$$

$$f_{24}=(\Delta t\ \ (m_2\,l_{c2}{}^2+I_2)\,(2\,l_1\,l_{c2}\,m_2\,\dot{q}_1\,S_2+2\,l_1l_{c2}\,m_2\,\dot{q}_2\,S_2)\,)\,/\,(I_1I_2+l_1^{\,2}\,l_{c2}^2m_2{}^2+I_2\,l_1{}^2m_2+I_2\,l_{c1}^2\times m_1+I_1\,l_{c2}{}^2\,m_2+l_{c1}{}^2l_{c2}{}^2\,m_1\,m_2-l_1{}^2l_{c2}{}^2m_2{}^2C_2{}^2).$$

$f_{31}=0$, $f_{32}=0$, $f_{33}=1$, $f_{34}=\Delta t$

$$f_{41}=-\Delta t\,(((g\,S_1(l_1m_2+l_{c1}m_1)+g\,l_{c2}\,m_2\,S_{12})(m_2l_{c2}{}^2+l_1\,m_2\,C_2l_{c2}+I_2)\,)\,/\,(I_1I_2+l_1{}^2l_{c2}{}^2\,m_2{}^2+I_2\times l_1{}^2m_2+I_2l_{c1}{}^2\,m_1+I_1l_{c2}{}^2m_2+l_{c1}{}^2l_{c2}{}^2m_1m_2-l_1{}^2\,l_{c2}{}^2m_2{}^2\,C_2{}^2)-(g\,l_{c2}\,m_2S_{12}(m_2l_1^2+2\,m_2C_2\,l_1\,l_{c2}+m_1\,l_{c1}{}^2+m_2\,l_{c2}{}^2+I_1+I_2))\,/\,(I_1I_2+l_1^{\,2}\,l_{c2}{}^2m_2^{\,2}+I_2l_1^{\,2}\,m_2+I_2\,l_{c1}{}^2\,m_1+I_1\,l_{c2}{}^2m_2+l_{c1}{}^2l_{c2}{}^2m_1m_2-l_1{}^2l_{c2}{}^2m_2{}^2C_2{}^2)).$$

$$f_{42}=-\Delta t\ \ (\,(2\ \dot{q}_2\,l_1\,l_{c2}\,m_2\,S_2(m_2\,l_{c2}{}^2+l_1\,m_2\,C_2\,l_{c2}+I_2)\ \,)/(I_1\,I_2+l_1^{\,2}\,l_{c2}{}^2\ \,m_2{}^2+I_2\,l_1{}^2\,m_2+I_2\,l_{c1}{}^2\times m_1+I_1l_{c2}{}^2\,m_2+l_{c1}{}^2l_{c2}{}^2\,m_1m_2-l_1{}^2l_{c2}^2m_2{}^2\,C_2{}^2)+(2\,\dot{q}_1\,l_1l_{c2}\,m_2\,S_2\ \,(m_2l_1^2+2\,m_2\,C_2\,l_1l_{c2}+m_1l_{c1}^2+m_2\,l_{c2}{}^2+I_1+I_2\,))\,/\,(I_1I_2+l_1^{\,2}\,l_{c2}{}^2\,m_2{}^2+I_2l_1{}^2\,m_2+I_2l_{c1}{}^2\,m_1+I_1l_{c2}{}^2m_2+l_{c1}{}^2l_{c2}{}^2m_1m_2-l_1{}^2l_{c2}{}^2\times m_2{}^2C_2{}^2)).$$

$$f_{43}=\Delta t(\,(\,(g\,l_{c2}m_2S_{12}-l_1l_{c2}\,m_2\,C_2\,\dot{q}_1{}^2)\,(m_2l_1{}^2+2m_2\,C_2\,l_1l_{c2}+m_1l_{c1}{}^2+m_2l_{c2}{}^2+I_1+I_2)\,)\,/\,(I_1I_2+l_1{}^2l_{c2}{}^2m_2{}^2+I_2l_1{}^2m_2+I_2l_{c1}{}^2m_1+I_1l_{c2}{}^2\,m_2+l_{c1}{}^2\,c_2{}^2\,m_1m_2-l_1{}^2l_{c2}^2\,m_2{}^2\,C_2{}^2)-((\dot{q}_2\,(\dot{q}_1\,l_1\,l_{c2}m_2\,C_2+\dot{q}_2\,l_1\,l_{c2}m_2C_2)+g\,l_{c2}\,m_2\,S_{12}+\dot{q}_1\,\dot{q}_2\,l_1l_{c2}\,m_2C_2)\,(m_2l_{c2}{}^2+l_1\,m_2C_2\,l_{c2}+I_2)\,)/(I_1I_2+l_1{}^2l_{c2}{}^2m_2{}^2+I_2l_1{}^2m_2+I_2l_{c1}{}^2m_1+I_1l_{c2}{}^2m_2\ \ +l_{c1}{}^2l_{c2}{}^2m_1m_2-l_1{}^2\,l_{c2}{}^2m_2{}^2C_2{}^2)+(l_1l_{c2}\,m_2\,S_2\,(\dot{q}_2\ \,(\dot{q}_1\,l_1\,l_{c2}\,m_2\,S_2+\dot{q}_2\,l_1l_{c2}\,m_2\,S_2)-g\,C_1\,(l_1m_2+l_{c1}m_1)-g\ \,l_{c2}\,m_2C_{12}+\dot{q}_1\dot{q}_2\ \,l_1l_{c2}\,m_2\,S_2)\,)/(I_1I_2+l_1{}^2l_{c2}{}^2\,m_2{}^2+I_2\times l_1{}^2m_2\,I_2\,l_{c1}{}^2\,m_1+I_1\,l_{c2}{}^2m_2+l_{c1}{}^2\,l_{c2}{}^2m_1\,m_2-l_1{}^2l_{c2}{}^2\,m_2{}^2\,C_2{}^2)+(2\,l_1l_{c2}\,m_2\,S_2\,(l_1l_{c2}\,m_2\,S_2\,\dot{q}_1{}^2+g\times l_{c2}\,m_2\,C_{12}))\,/\,(I_1I_2+l_1^{\,2}\,l_{c2}{}^2\,m_2{}^2I_2l_1{}^2\,m_2+I_2l_{c1}{}^2m_1+I_1l_{c2}{}^2\,m_2+l_{c1}{}^2\,l_{c2}{}^2\,m_1m_2-l_1{}^2l_{c2}{}^2m_2{}^2C_2{}^2)+(2\,l_1{}^2l_{c2}{}^2\,m_2{}^2C_2S_2\,(m_2l_{c2}{}^2+l_1\,m_2C_2l_{c2}+I_2)(\dot{q}_2\,(\dot{q}_1l_1l_{c2}\,m_2\,S_2+\dot{q}_2\,l_1l_{c2}\,m_2S_2)-gC_1(l_1m_2+l_{c1}\times m_1)-g\,l_{c2}m_2C_{12}+\dot{q}_1\dot{q}_2\,l_1l_{c2}\,m_2S_2))\,/\,(I_1I_2+l_1{}^2l_{c2}{}^2\,m_2{}^2+I_2l_1{}^2\,m_2+I_2l_{c1}{}^2\,m_1+I_1l_{c2}{}^2m_2+l_{c1}{}^2\times l_{c2}{}^2m_1m_2-l_1{}^2l_{c2}{}^2m_2{}^2\,C_2{}^2\,)^2+(2\,l_1{}^2l_{c2}{}^2m_2{}^2C_2S_2\,(l_1l_{c2}\,m_2S_2\,\dot{q}_1{}^2+gl_{c2}m_2C_{12})(m_2l_1{}^2+2\,m_2C_2l_1\times l_{c2}+m_1l_{c1}{}^2+m_2l_{c2}{}^2+I_1+I_2)\,)\,/\,(I_1I_2+l_1{}^2l_{c2}{}^2m_2{}^2+I_2l_1{}^2\,m_2\ \ +I_2l_{c1}{}^2\,m_1+I_1l_{c2}{}^2\,m_2+l_{c1}{}^2\,l_{c2}{}^2\times m_1m_2-l_1{}^2l_{c2}{}^2m_2{}^2C_2{}^2\,)^2)+(\Delta t\ \,l_1\,l_{c2}\,m_2\,\tau_1\,S_2)\,/\,(I_1I_2+l_1{}^2l_{c2}{}^2\,m_2{}^2+I_2l_1{}^2m_2+I_2l_{c1}{}^2\,m_1+I_1l_{c2}{}^2\times m_2+\,l_{c1}{}^2l_{c2}{}^2m_1m_2-l_1{}^2l_{c2}{}^2m_2{}^2\,C_2{}^2)-(2\Delta t\,l_1l_{c2}\,m_2\,\tau_2S_2)/(I_1I_2+l_1{}^2l_{c2}^2m_2{}^2+I_2l_1{}^2m_2+I_2l_{c1}{}^2m_1+I_1l_2^2\,m_2+l_{c1}^2\,l_2^2\,m_1\,m_2-l_1^{\,2}\,l_{c2}^2\,m_2{}^2\,C_2^2)-(2\,\Delta t\,\ l_1^{\,2}\,l_{c2}^2\,m_2{}^2\,\tau_2C_2\,S_2(m_2\,l_1{}^2+2\,m_2C_2\,l_1\,l_{c2}+m_1\,l_{c1}^2+m_2l_{c2}{}^2+I_1+I_2))/(I_1I_2+l_1{}^2l_{c2}{}^2m_2{}^2+I_2l_1{}^2m_2+_2\,l_{c1}{}^2m_1+I_1l_{c2}{}^2\,m_2+l_{c1}{}^2l_{c2}{}^2\,m_1m_2-l_1{}^2l_{c2}{}^2\,m_2{}^2\times C_2{}^2)^2+(2\,\Delta t\ \,l_1^{\,2}\,l_{c2}{}^2m_2{}^2\tau_1C_2S_2\,(m_2l_{c2}{}^2+l_1m_2C_2l_{c2}+I_2))/(I_1I_2+l_1{}^2l_{c2}{}^2m_2{}^2+I_2l_1{}^2m_2+I_2l_{c1}{}^2\times m_1+I_1l_{c2}{}^2m_2+l_{c1}{}^2l_{c2}{}^2m_1m_2-l_1{}^2l_{c2}{}^2m_2{}^2C_2{}^2)^2.$$

$$f_{44}=1-(\,\Delta t\ \,(2\ \dot{q}_1\,l_1l_{c2}\,m_2S_2+2\ \dot{q}_2\,l_1l_{c2}\,m_2S_2)\,(m_2\,l_{c2}{}^2+l_1\,m_2C_2l_{c2}+I_2))\,/\,(I_1I_2+l_1{}^2l_{c2}{}^2m_2{}^2+I_2\times l_1{}^2m_2+I_2l_{c1}{}^2m_1+I_1l_{c2}{}^2m_2+l_{c1}{}^2l_{c2}{}^2m_1m_2-l_1{}^2l_{c2}{}^2m_2{}^2C_2{}^2)$$

$S_{12}=sin(q_1+q_2)$, $S_1=sin(q_1)$, $S_2=sin(q_2)$, $C_{12}=cos(q_1+q_2)$, $C_1=cos(q_1)$, $C_2=cos(q_2)$

Annexe C.

Rappels sur quelques notions de géométrie différentielle et l'observabilité des systèmes non-linéaires

Cet annexe contient quelques notions de géométrie différentielle et quelques définitions d'observabilité des systèmes non-linéaires, nécessaires à la compréhension du contenue de cette thèse.

C.1. Quelques notions de géométrie différentielle l'algèbre [1][62]

Définition C.1 (***Difféomorphisme***) **:** *Soit p un point de $\mathbb{R}^n$ et ϕ une application d'un ouvert $U \subset \mathbb{R}^n$ dans un ouvert $V \subset \mathbb{R}^n$, on dit que ϕ est un difféomorphisme local dans un voisinage U de p si ϕ est inversible de U dans un voisinage V du point $\phi(p)$ de V , et si ϕ^{-1} et ϕ est continue.*

Définition C.2 ***(Champ de vecteurs)*****:** *un champ de vecteur sur $\mathbb{R}^n$ est une fonction dérivable*

$$f : \mathbb{R}^n \to \mathbb{R}^n : \; f(X) = \begin{bmatrix} f_1(x) & f_2(x) & \dots & f_n(x) \end{bmatrix}^T \tag{C.9}$$

Définition C.3 ***(Dérivée de Lie)*** *Considérons h une fonction C^∞ de $\mathbb{R}^n$ dans $\mathbb{R}$. On définit la dérivée de Lie de h dans la direction de f, notée $L_f h$, la dérivée de h le long de la courbe intégrale de f en $t = 0$:*

$$L_f h(x) = \sum_{i=1}^{n} f_i(x) \frac{\partial h}{\partial x_i(x)} \tag{C.10}$$

Par définition, on écrit : $L_f^0 h = h$ et $L_f^k h = L_f \left(L_f^{k-1} h \right)$

Définition C.4 ***(Crochet de Lie)*** *On appelle crochet de Lie de deux champs de vecteurs f et g, le produit $[f, g]$ dont l'expression, dans les nouvelles coordonnées locales, est donnée par les n vecteurs suivants :*

$$[f, g](x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} - \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} g_1 \\ \vdots \\ g_n \end{bmatrix} \tag{C.11}$$

On peut répéter le crochet de Lie autant de fois qu'on veut, d'où l'opérateur suivant :

$$\left[f\left[f, \ldots, [f, g]\right]\right] = ad_f^k\, g(x) = \left[f, ad_f^{k-1}\, g\right](x) \text{ pour } k \geq 1 .$$

pour $k = 0$, *on a :* $ad_f^0\, g(x) = g(x)$.

C.2. Observabilité des systèmes non-linéaires

C.2.1. Cas continu

Considérons le système non-linéaire donné par la forme suivante :

$$\begin{cases} \dot{x}(t) = f(x) \\ y = h(x) \end{cases} \tag{C.1}$$

où $x \in M \subseteq \mathbb{R}^n$, et $y_k \in \mathbb{R}^p$ représente respectivement, l'état et la sortie du système. Les fonctions f, h sont des vecteurs de fonctions analytiques de dimensions appropriées. Notons que pour tout $x^0 \in M$, il existe une solution pour $\dot{x}(t) = f(x(t))$ telle que $x(0) = x^0$ et $x(t) \in M$ pour tout $t \in \mathbb{R}$. On note U un sous ensemble ouvert de M .

C.2.1.1. Condition du rang d'observabilité en temps continu

Définition C.5

Considérons le système dynamique de la forme (C.1). On dit que la paire $(f; h)$ *est observable au sens du rang si la condition donnée par (C.2) est satisfaite.*

$$rang(O) = rang \begin{pmatrix} dh \\ dL_f h \\ \vdots \\ dL_f^{n-1} h \end{pmatrix} = n \tag{C.2}$$

où

$$O = \begin{pmatrix} dh & dL_f h & \ldots & dL_f^{n-1} h \end{pmatrix}^T \tag{C.3}$$

avec $L_f h$ *est une fonction, s'appelle la dérivée de Lie de* h *dans la direction de* f .

$dL_f^{n-1} h$ *est la différentielle de* $L_f h$, *donné par:*

$$dL_f^n h = \left(\frac{\partial L_f^n h}{\partial x_1}, \frac{\partial L_f^n h}{\partial x_2}, \, \dots \, , \frac{\partial L_f^n h}{\partial x_n} \right) \tag{C.4}$$

C.2.2. Cas discret

Considérons le système non-linéaire à temps-discret donné par la forme suivante :

$$\begin{cases} x_{k+1} = f\left(x_k, u_k\right) \\ y_k = h\left(x_k\right) \end{cases} \tag{C.5}$$

où $x_k \in \mathbb{R}^n$, $u_k = \left(u_{1k}, \dots, u_{mk}\right) \in \mathbb{R}^m$ et $y_k \in \mathbb{R}^p$. Pour toute entrée $u_k \in \mathbb{R}^m$ constante, $f_{u,k}\left(x_k\right) = f\left(x_k, u_{m,k}\right)$ est un champ de vecteur C^∞ sur $\mathbb{R}^n$ et les h_i pour $i = 1, 2, \dots, p$, les composantes de h qui sont des fonctions définies de C^∞ de $\mathbb{R}^n$ sur $\mathbb{R}$.

De même que pour le cas continu, le problème d'observabilité consiste à pouvoir reconstruire tous les états du système à partir de la sortie et de ses itérées.

C.2.2.1. Condition de rang d'observabilité en temps discret

L'observabilité du système (C.5) peut être aussi vérifiée par la condition de rang d'observabilité.

Définition C.6 ***(observabilité au sens du rang)*** *Le système (C.5) est dit observable au sens du rang en* $x_0 \in \mathbb{R}^n$ *si :*

$$dim\left(dOh(x_0)\right) = n \tag{C.6}$$

où l'espace d'observation $O\left(h\right)\left(x_k\right)$ *est défini par :*

$$O\left(h\right)(x_k) = span\left(h_i\left(x_k\right), h_i \circ f_{u_{1,k}}\left(x_k\right), \dots, h_i \circ f_{u_{1,k}} \circ \dots \circ f_{u_{l,k}}\left(x_k\right)\right) \tag{C.7}$$

tel que : $1 \le i \le p$, $u_{1,k}, \dots, u_{l,k} \in \mathbb{R}^n$ *et* $x_k \in \mathbb{R}^n$.

Cette définition peut être reformulée comme suit :

Définition C.7 ***(observabilité au sens du rang)*** *Le système (C.5) est dit observable au sens du rang en* $x_0 \in \mathbb{R}^n$ *si :*

$$span\left\{dh,\ d(f \circ h), \dots, d(f^{n-1} \circ h)\right\} \tag{C.8}$$

est de rang n.