

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

Pour obtenir le grade de

DOCTEUR D'UNIVERSITE

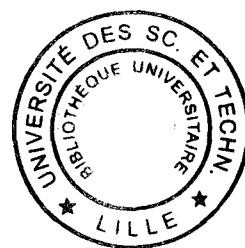
Spécialité : Automatique et Informatique Industrielle

par

Régis CANART

Ingénieur ICAM

Contribution à la Commande Adaptative et Prédictive.



Soutenu le 27 novembre 1997 devant le jury composé de :

Président : J-P. RICHARD (LAIL - EC Lille)

Rapporteurs : M. KSOURI (ENIT de Tunis)
A. RACHID (LSA - Université de Picardie)
F. ROTELLA (LGP - ENIT de Tarbes)

Examineurs : P. BORNE (LAIL - EC Lille)
A. EI KAMEL (LAIL - EC Lille)
D. MEIZEL (HEUDIASYC - UTC)
P. VANHEEGHE (ISEN - Lille)
C. VASSEUR (USTL / I3D - Lille I)

Travaux réalisés sous la direction du Professeur P. BORNE au Laboratoire d'Automatique et d'Informatique Industrielle de Lille (LAIL URA-CNRS D1440), Ecole Centrale de Lille, Villeneuve d'Ascq, France.

Avant-propos

Les travaux de recherche présentés dans ce mémoire ont été réalisés au Laboratoire d'Automatique et d'Informatique Industrielle de l'Ecole Centrale de Lille sous la direction du Professeur Pierre Borne.

Je lui en suis très reconnaissant et tiens à lui exprimer mes plus vifs remerciements pour m'avoir accueilli au LAIL dans de très bonnes conditions, et pour m'avoir ensuite conseillé tout en me laissant prendre de nombreuses initiatives dans mes travaux, et en m'accordant une liberté très appréciée. Je ne cache pas non plus mon admiration pour son dynamisme incontestable qui se retrouve au travers de ses innombrables activités scientifiques reconnues internationalement.

Je souhaite également remercier le Professeur J-P. Richard pour l'honneur qu'il me fait en acceptant de présider le jury de thèse, pour l'intérêt qu'il a manifesté envers mes recherches, et pour les conseils judicieux qu'il m'a prodigués avec un sens aigu de la rigueur mathématique et un souci permanent de clarté.

Je tiens aussi à remercier vivement le Professeur M. Ksouri d'avoir accepté le long et méticuleux travail de rapporteur. Ses connaissances et conseils sur la commande adaptative et prédictive constituent « des retours d'information » très riches qui ouvrent de nombreuses perspectives de recherche.

Mes très vifs remerciements s'adressent aussi au Professeur A. Rachid. Sa grande expérience dans le domaine des systèmes automatiques et son approbation à propos des travaux présentés mettent en valeur ce mémoire et les résultats obtenus.

Je souhaite également remercier le Professeur F. Rotella d'avoir accepté d'être rapporteur. Ses suggestions et remarques constructives se sont avérées très bénéfiques pour la rigueur et la clarté des développements théoriques et pour la qualité générale du mémoire.

Je remercie également avec gratitude Monsieur A. El Kamel pour sa participation au jury de thèse, et pour ses conseils avisés et fort utiles.

Je voudrais aussi remercier vivement le Professeur D. Meizel pour l'honneur qu'il me fait en acceptant de participer au jury de thèse, et pour l'intérêt qu'il témoigne envers mes travaux.

De même, je souhaite adresser de sincères remerciements à Monsieur P. Vanheeghe dont les rencontres et discussions occasionnelles ont donné lieu à des idées fructueuses.

Je suis également très honoré de la présence du Professeur C. Vasseur, et le remercie de participer au jury et de consacrer un temps précieux à l'étude de ce mémoire.

Pour son aide très appréciable lors de la mise au point des algorithmes de commande sur la machine « TALC », je suis très reconnaissant envers Monsieur J. Lasue. Je souhaite aussi adresser mes remerciements à Monsieur D. Lamy et Monsieur A. Saadane de l'ICAM avec qui j'ai pu garder de bons contacts, à l'occasion de multiples rencontres et grâce aux heures d'enseignement qu'ils m'ont confiées. Je voudrais également exprimer toute ma gratitude à l'ensemble du personnel du LAIL, de l'Ecole Centrale de Lille et du rectorat de l'académie de Lille, qui tous à leur manière contribuent au bon déroulement des activités de recherche et d'enseignement. Un grand merci en particulier à Max Vangreveninge, à qui j'ai très souvent demandé des travaux de reprographie à la dernière minute et de toute urgence.

Enfin ces dernières lignes de remerciements sont adressées à tous les chercheurs et thésards avec qui j'ai sympathisé pendant ces deux années de thèse. Je les remercie cordialement pour leur bonne humeur, leur amitié, leurs encouragements, leur aide et leurs conseils pour l'avenir...

Table des Matières

Introduction Générale.....	8
Chapitre I : Modélisation des Processus.....	11
1. Définition et rappels	11
2. Modélisation par réseaux de neurones ?.....	13
3. Modélisation par hyperplans.....	15
3.1. Modèle résultant des équations d'état.....	16
3.2. Modèle résultant des fonctions de transfert	19
3.3. Détermination de la période de régression	20
3.4. Modèle sous-échantillonné	23
3.5. Exemple	25
4. Généralisation à des systèmes non linéaires	27
4.1. Approximation linéaire.....	27
4.2. Définition d'un modèle local	28
4.3. Ecriture des modèles dans un repère local.....	29
5. Extension à certaines classes particulières de systèmes	30
5.1. Modélisation de systèmes à retards purs.....	30
5.2. Modélisation de systèmes bilinéaires	31
5.3. Modélisation de systèmes linéaires par rapport aux paramètres à adapter	32
6. Modélisation de zones inactives	33
6.1. Description et choix du modèle	33
6.2. Estimation de la zone inactive	36
6.3. Compensation	38
7. Validation du modèle.....	40
7.1. Influence des perturbations.....	40
7.2. Détection de dysfonctionnements.....	41
7.3. Estimation du degré de validité	42
8. Conclusion	44

Chapitre II : Adaptation des modèles	45
1. Introduction.....	45
2. Adaptation par rétro-propagation du gradient.....	45
3. Loi d'adaptation déduite d'une fonction de Lyapunov	46
4. Algorithme basé sur la méthode des moindres carrés.....	49
4.1. Choix de la fenêtre d'observation.....	49
4.2. Problème d'excitation permanente	50
4.3. Critère avec ajout de points déduits du modèle	51
4.4. Critère pour un modèle exprimé autour du point de fonctionnement.....	56
4.5. Critère pour le modèle avec la zone inactive.....	57
4.6. Critère pour le modèle déduit des fonctions de transfert	58
4.7. Rappels sur la résolution de systèmes linéaires	59
5. Expression des calculs en récursif	61
5.1. Principe et intérêts	61
5.2. Robustesse de l'algorithme.....	62
5.3. Inversion en récursif	67
6. Robustesse de la loi d'adaptation	68
6.1. Influence du bruit et filtrage des mesures	68
6.2. Vitesse de convergence.....	72
7. Conclusion	74

Chapitre III : Méthodes prédictives	75
1. Prédications de la consigne	75
1.1. Extrapolation de la consigne.....	75
1.2. Détection de discontinuité	78
2. Prédications des sorties du système.....	79
2.1. Prédications dans le repère global	79
2.2. Prédications dans le repère local	81
2.3. Sensibilité du calcul des prédictions.....	83
3. Calcul des erreurs prédites.....	84
4. Prédications avec des commandes en échelon ou en rampe.....	85
4.1. Contrainte sur l'allure de la commande.....	85
4.2. Formulation dans le repère global	86
4.3. Formulation dans le repère local.....	86
5. Conclusion	86

Chapitre IV : Détermination de la loi de commande.....	87
1. Introduction.....	87
2. Principes et règles de base de la commande	87
2.1. Détermination de la consigne et du modèle de référence	87
2.2. Dynamique d'annulation des erreurs	88
2.3. Erreur statique et détermination du point de fonctionnement	89
3. Critère avec limitations des dérivées successives des commandes	90
3.1. Définition du critère.....	90
3.2. Réglage des coefficients de pondération	91
3.3. Calcul de la loi de commande.....	92
3.4. Exemple d'application sur un robot PUMA 560	92
4. Commande minimisant les erreurs et les variations des erreurs.....	94
4.1. Définition du critère.....	94
4.2. Influence de l'allure des commandes futures.....	96
4.3. Commande d'un système multivariable	97
4.4. Commande d'un système multivariable instable	98
5. Commande avec réglage du taux d'annulation de l'erreur	100
5.1. Définition du critère.....	100
5.2. Critère avec pondération des erreurs prédites.....	102
5.3. Exemple d'un système comportant un retard pur	103
5.4. Etude d'un système d'ordre élevé.....	104
6. Commande avec filtrage des erreurs.....	106
6.1. Détermination des coefficients du filtre	106
6.2. Définition du critère et résolution.....	106
6.3. Formulation récursive des calculs	107
6.4. Exemple de commande d'un système bruité	108
6.5. Etude d'un système à non minimum de phase.....	110
7. Conclusion	113

Chapitre V : Mise en oeuvre sur la machine TALC	114
1. Description du processus	114
1.1. Modélisation de la partie optique	115
1.2. Modélisation de la partie mécanique	119
2. Etude du contrôleur et du processus en simulation	121
2.1. Définition du modèle et estimation des zones inactives	121
2.2. Définition du contrôleur et simulation sans perturbation	122
2.3. Influence des perturbations	123
3. Etude expérimentale	124
3.1. Essai sans perturbation	124
3.2. Influence des perturbations	126
4. Conclusion	127
Conclusion générale.....	128
Bibliographie	129

Introduction Générale

Commander un processus est une tâche ardue dont la réalisation doit s'effectuer en prenant en considération de multiples contraintes. Une commande de systèmes, que ce soit pour des robots ou pour tout autre processus doit répondre aux nombreuses exigences stipulées lors de la conception. Les caractéristiques imposées et les propriétés indispensables sont la robustesse, la stabilité et la précision de la loi de commande. Cette loi doit être établie de sorte que le processus réalise les opérations souhaitées avec des performances optimales dans des conditions de sécurité maximales. La détermination du système de contrôle est donc un point très important pendant la conception puis lors de l'utilisation des processus.

Les objectifs à atteindre sont multiples et difficiles à satisfaire simultanément. Une première catégorie de conditions à respecter vise à garantir la sécurité de fonctionnement du processus. La robustesse numérique des algorithmes, leur stabilité, la robustesse de la loi de commande face à des perturbations ou à des bruits sur les mesures sont des points qui méritent un examen attentif. Ces problèmes sont d'autant plus cruciaux, que les méthodes adaptatives ont tendance à amplifier et à aggraver les conséquences dues à la moindre défaillance d'un actionneur ou d'un capteur. Pour des raisons de sécurité, il est alors impératif d'élaborer un algorithme de supervision, ou au minimum de surveillance du processus et de la loi de commande. Un premier niveau d'analyse sur l'état du processus et du contrôleur s'avère alors nécessaire pour valider et tester le bon fonctionnement de l'ensemble.

Dans un second temps, les objectifs à atteindre ont rapport aux performances du contrôleur. On cherche à obtenir une bonne précision avec un temps de réponse le plus court possible. Le processus doit donc suivre la consigne imposée avec des erreurs les plus petites possibles en évitant les oscillations et les dépassements. La loi de commande doit aussi être définie de manière à préserver les actionneurs et le processus en limitant les efforts et les sollicitations auxquels ils sont soumis. On juge également les performances d'une loi de commande à sa faculté d'adaptation aux changements de conditions de fonctionnement. Une grande flexibilité et une grande robustesse par rapport à des variations de l'environnement sont souhaitables.

En troisième point, la mise en oeuvre sur calculateur de la loi de commande et les limites imposées par le matériel utilisé engendrent des restrictions et des contraintes supplémentaires. Il faut en effet tenir compte de certains problèmes annexes mais bien réels tels que les problèmes du temps de calcul requis et de la précision des calculs qui est parfois à l'origine d'erreurs importantes. D'autre

part, la précision des mesures effectuées par les capteurs, et les limites physiques des actionneurs sont d'autres contraintes matérielles non négligeables.

Pour parvenir à une loi de commande répondant aux spécifications souhaitées, et qui aboutit à un bon compromis permettant à la fois d'améliorer les performances et d'utiliser les processus dans de meilleures conditions de sécurité, nous nous basons sur le principe de la commande adaptative, indirecte et prédictive. L'approche retenue se schématise ainsi :

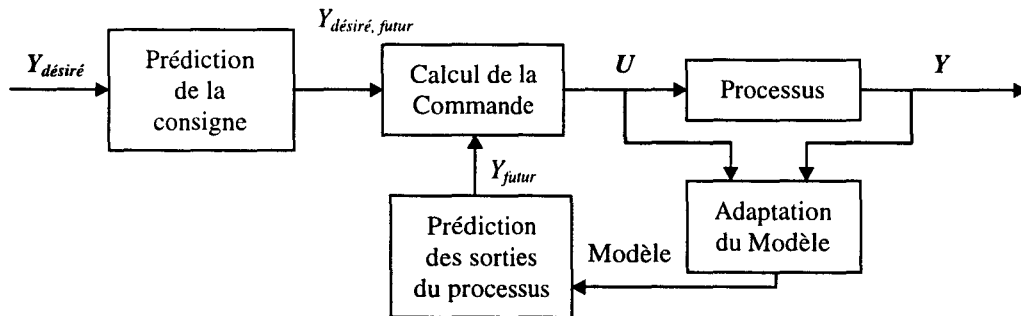


Figure 1 : Schéma de principe d'un contrôleur adaptatif indirect.

Ce schéma composé des blocs représentant le processus, le modèle à adapter, le calcul des prédictions, et la loi de commande fait ressortir clairement les thèmes clés autour desquels s'articule le plan de la thèse.

L'analyse de plusieurs modèles et de leur bien-fondé pour la commande adaptative fera l'objet d'un premier chapitre. Les rappels théoriques essentiels et les références s'y rattachant seront énoncés au fur et à mesure dans chaque chapitre aux moments opportuns. Dans le deuxième chapitre nous étudions plusieurs méthodes d'adaptation de ces modèles. Le temps de convergence, la stabilité de l'adaptation et la robustesse des algorithmes sont les principaux aspects abordés.

Le sujet du troisième chapitre porte sur l'utilisation des modèles précédemment définis pour déduire les valeurs futures de l'état du processus. Les intérêts de la commande prédictive et ses limites y sont développés et analysés. Les lois de commande établies dans le quatrième chapitre sont fondées sur les prédictions des sorties du système et sur les erreurs prédites par rapport à la consigne. En effet ce sont ces erreurs qui doivent être minimisées selon des critères assurant à la fois une bonne stabilité, une bonne précision et de bonnes performances dynamiques.

Les lois de commande proposées dans le quatrième chapitre reposent sur des critères permettant d'atteindre les objectifs fixés de différentes manières. L'étude de la sensibilité par rapport au bruit de mesure et aux erreurs de modélisation sera également entreprise. Des exemples en simulation illustreront les résultats présentés, et mettront en avant les propriétés de chacune des lois de commande. Le cinquième chapitre sera consacré à un exemple expérimental de commande de processus multivariable et non linéaire, qui permet de valider les méthodes de commande proposées et de tester en pratique la robustesse des algorithmes utilisés.

Chapitre I :

Modélisation des Processus

1. Définition et rappels

Dans le cas des méthodes de commandes adaptatives indirectes et prédictives, le modèle joue un rôle primordial, car c'est à partir de ce dernier que s'établissent les prédictions du processus et par conséquent la loi de commande. Nous allons donc dans cette partie rappeler quelques notions élémentaires à propos de la modélisation de systèmes dynamiques, et les conditions à respecter pour obtenir un modèle le plus fidèle possible par rapport au système réel.

La modélisation d'un processus découle soit de lois physiques, soit de l'analyse et du traitement de données expérimentales. Dans le premier cas, les lois physiques, électriques, électromagnétiques, thermodynamiques ou les lois telles que le principe fondamental de la dynamique, décrivent le fonctionnement du processus et en donnent un modèle. Ce dernier indique comment se comporte le système et quelle est l'influence des divers phénomènes mécaniques, électriques ou électromagnétiques...

Cette étape est indispensable lors de la conception d'un processus pour analyser, optimiser et simuler son évolution. Mais ce type d'approches atteint ses limites lorsqu'il faut contrôler un système dont la connaissance des valeurs physiques est imprécise voire très approximative. Certaines valeurs peuvent être parfois difficiles à déterminer théoriquement. On préfère alors opter pour un modèle obtenu expérimentalement. L'identification du processus s'effectue à l'aide des données et mesures prises lors d'essais réalisés spécifiquement pour identifier le processus [POV 75]. On peut alors citer à titre indicatif certaines méthodes fréquentielles et temporelles fréquemment employées. Par exemple, le traitement adéquat des données entrées-sorties du processus fournit le diagramme de Bode, le diagramme de Black ou encore le lieu de Nyquist qui sert ensuite à calculer les compensateurs ou autres contrôleurs à implanter. Pour des systèmes du premier ou second ordre, des méthodes basées sur l'étude de la réponse indicielle permettent de déduire une estimation de la fonction de transfert du processus, que l'on exploite pour aboutir à une loi de commande répondant aux spécifications désirées.

Chapitre I : Modélisation des Processus

Les approches théoriques et expérimentales peuvent être utilisées avantageusement ensemble car chacune apporte des informations nécessaires à l'autre. La modélisation physique du processus indique les propriétés structurelles, alors que l'expérimentation permet d'estimer les valeurs numériques des paramètres du modèle.

Néanmoins, en fonction des conditions d'utilisation, les caractéristiques et paramètres du processus peuvent varier dans un large intervalle et surtout de manière inconnue a priori. Dans des conditions aussi incertaines et évolutives, une loi de commande basée sur un modèle figé aura des performances qui se dégradent dès lors que les erreurs de modélisation par rapport au processus réel deviennent trop importantes. La stabilité de l'ensemble « processus et contrôleur » peut même devenir critique. On observe alors soit des temps de réponse très longs soit au contraire des phénomènes oscillatoires indésirables. Ces variations entraînent éventuellement des saturations des actionneurs, puis accidentellement une divergence de la loi de commande, et finalement une détérioration dangereuse du processus. Dans le cas de systèmes linéaires, la solution préconisée pour éviter ce genre d'incidents est d'ajouter des compensateurs et de modifier les réglages du contrôleur afin d'améliorer les marges de gain et de phase. L'obtention d'une plus grande robustesse s'effectue alors au détriment des performances dynamiques. Les temps de réponse se trouvent rallongés et les erreurs moyennes de position ou de vitesse augmentent.

Ces limitations et ce type de compromis sont inhérents aux méthodes non adaptatives. Pour repousser ces limites, il faut soit un mécanisme d'ajustement ou d'auto-réglage des paramètres du contrôleur [BOR 90] (méthodes adaptatives directes), soit un algorithme d'identification en temps réel (méthodes adaptatives indirectes) [BIS 92, MEY 93, ROV 95]. En fonction des techniques employées ou approches retenues, les termes utilisés sont loi d'adaptation ou loi d'apprentissage. Dans ce chapitre, ces deux termes désignent l'identification du processus et la mise à jour des paramètres du modèle.

Ces premières remarques d'ordre général ont pour objectif de souligner l'intérêt et la nécessité d'identifier le système en ligne, lorsque ses paramètres sont susceptibles d'évoluer dans des proportions importantes. L'identification en ligne consiste alors à obtenir à tout instant un modèle, dont le comportement dynamique est le plus proche possible du comportement réel du système. Contrairement à des approches de modélisation physique des processus, comme l'approche Bond Graph, notre but pour la commande est avant tout d'obtenir un modèle mathématique traduisant le comportement entrées-sorties du processus. Nous ne cherchons donc pas à mettre en évidence les phénomènes physiques intervenant dans le comportement du processus, ni à en dégager les

propriétés structurelles, mais seulement à connaître l'influence quantitative des valeurs d'entrées sur les variables de sorties. Le modèle est donc considéré comme une fonction particulière dépendant du temps, entre les entrées et les sorties du processus. Ce point de vue laisse une très grande liberté dans le choix du modèle. Toute fonction décrivant correctement le comportement du processus et l'influence des entrées sur les sorties peut a priori être utilisée pour déduire une loi de commande. La modélisation du processus par des réseaux de neurones réputés pour leur faculté à donner une approximation de n'importe quelle fonction est donc étudiée.

2. Modélisation par réseaux de neurones ?

La littérature scientifique sur les réseaux de neurones artificiels est abondante et a connu une forte croissance cette décennie traduisant l'intérêt de telles méthodes pour résoudre ou apporter des solutions innovantes à des problèmes dans des domaines très variés. La diversité des types d'applications possibles a conduit à une multitude de types de réseaux de neurones aux caractéristiques et propriétés fortement distinctes [PAT 95]. Sous le même terme est donc regroupé un large éventail d'architectures de réseaux. Nous évoquons dans ce chapitre uniquement l'utilisation de réseaux de neurones pour l'identification de processus, pour la commande et pour la prédiction de séries temporelles.

Le premier avantage est la grande diversité de solutions qu'offrent les réseaux de neurones artificiels pour traiter un problème de modélisation. Ils ont la capacité de donner une approximation de n'importe quelle fonction avec une précision arbitraire, dépendant uniquement de la taille du réseau et de la fonction à identifier. L'approche par les réseaux de neurones est donc très attrayante. Cette propriété d'approximateur universel [NAR 90] est une des raisons majeures du succès indéniable et du vif intérêt porté sur les réseaux de neurones. Pour l'identification, ils effectuent des traitements sur les signaux et les informations et mesures provenant des capteurs qui permettent de déduire des modèles très généraux et qui sous certaines conditions conviennent pour donner un modèle de n'importe quel processus non linéaire et multivariable [AUS 95], [KOS 95]. Dans le domaine des prédictions de série temporelle, des réseaux de neurones multicouches se sont révélés très performants pour déduire par exemple à partir d'un grand nombre de données un modèle de prédiction du taux d'échange d'une monnaie qu'aucune loi économique ne permet d'établir précisément. La validation sur d'autres échantillons montre l'efficacité de tels modèles pour la prédiction des valeurs sur les 10 voire 50 pas futurs. La contrepartie est la complexité du modèle obtenu et la difficulté pour l'exploiter judicieusement. Tout calcul d'optimisation des coefficients du modèle, ou d'optimisation d'une variable d'entrée par rapport à une sortie devient un problème

d'optimisation non convexe et nécessite alors l'usage d'algorithmes complexes dont la difficulté de mise en oeuvre et le temps de calcul requis sont les principales limites pour la commande en temps réel.

Il faut aussi avoir conscience des limitations des réseaux de neurones dues à la lenteur d'adaptation des poids et à la durée d'apprentissage qui rend vaine toute tentative d'identification de processus variant rapidement dans le temps. En effet dans ce cas, à cause de la dynamique d'adaptation qui est plus lente que les variations du processus, le modèle obtenu ne peut pas donner une bonne approximation de la réalité. L'usage de réseaux de neurones pour des processus dont les conditions de fonctionnement évoluent brusquement et de manière inconnue semble donc illusoire. Seuls les processus stationnaires ou variant très lentement peuvent être identifiés par ce type de réseaux. La lenteur d'adaptation a pour autre conséquence de rendre indispensable une phase d'initialisation des poids du réseau. Durant cette étape souvent assez longue, le réseau n'est pas autonome, et n'est pas en mesure de commander le processus. Il doit uniquement s'adapter en fonction des données qui lui sont fournies. Cette étape préliminaire s'effectue généralement hors-ligne et peut nécessiter de nombreux essais, ce qui est plutôt contraignant et rallonge le temps de mise en oeuvre.

L'architecture du réseau choisi est un facteur très important. Certains types de réseau génèrent des fonctions ou des familles de fonctions dont l'allure ne reflète pas le comportement du processus. En effet la structure du réseau peut être très éloignée de la structure adéquate pour modéliser un processus. L'obtention d'un modèle satisfaisant n'est pas garantie et devient même très improbable. Une mauvaise adéquation de la structure du modèle par rapport au système lui-même conduit obligatoirement à des résultats d'un faible niveau. Même si les réseaux de neurones peuvent a priori identifier n'importe quelle fonction, il faut veiller à choisir une structure bien adaptée. L'utilisation par exemple des réseaux à fonctions radiales de base [SAN 92] ne convient pas forcément pour identifier et commander un système linéaire. En effet dans ce cas, cette méthode revient à essayer de trouver une approximation d'un plan par un ensemble de fonctions gaussiennes. Cette approximation donne des résultats satisfaisants que si le nombre et les paramètres de ces fonctions sont définis judicieusement a priori [SAD 95]. La détermination d'une structure correcte s'obtient principalement grâce à l'analyse du processus à commander.

En plus des nombreuses difficultés existantes pour identifier le système rapidement et pour parvenir à une structure correcte, les réseaux sont pénalisés par leur relative complexité de mise en oeuvre et par leur taille qui croît démesurément dès lors que l'on souhaite obtenir une meilleure précision et

que l'on s'intéresse à des processus multivariés. Des algorithmes ont été développés pour réduire ou augmenter le nombre de neurones en fonction de la précision obtenue et désirée. Ce test permet d'optimiser la taille du réseau par rapport au problème à résoudre. Cependant dans certains cas, le nombre de neurones et donc le nombre de poids à adapter augmente trop fortement et mène à des réseaux de trop grande taille difficiles à adapter correctement. Pour des raisons identiques, il est illusoire de vouloir identifier le système dans le domaine complet d'évolution des variables d'état, qui peut être très étendu dans le cas de systèmes de grande dimension comme par exemple un robot à 6 axes. Un modèle global valable dans tout l'espace d'état accessible est d'autant plus difficile à obtenir que les données disponibles à un instant donné sont en un point de l'espace, ou tout au plus dans une zone réduite sur une courte période. Obtenir un modèle complet et précis d'un processus complexe avec des réseaux de neurones est donc en pratique une opération peu réaliste.

Les méthodes basées sur les réseaux de neurones comportent donc des défauts limitant fortement leur usage pour la modélisation et la commande de processus. Les reproches sont généralement formulés à juste titre. D'une part les résultats ne sont pas toujours convaincants à cause de la lenteur d'adaptation, et d'autre part la notion physique y est absente.

3. Modélisation par hyperplans

L'étude sur les réseaux de neurones artificiels a mis en évidence les limites de cette approche mais surtout indique les voies à suivre pour aboutir à des méthodes adaptatives plus robustes. Des conditions à respecter apparaissent clairement pour réaliser une loi de commande efficace. La première contrainte concerne le temps et la dynamique d'adaptation qui doivent être plus rapides que les variations propres du processus. La seconde règle à respecter porte sur la structure du modèle qui doit être choisie de manière appropriée par rapport au processus. La troisième contrainte provient de la complexité des modèles. Ces derniers doivent en effet servir à calculer les prédictions et à établir la loi de commande. L'utilisation des modèles doit donc être suffisamment aisée pour parvenir aux objectifs désirés.

Notre choix se dirige alors vers les modèles linéaires discrets, qui peuvent être considérés comme des cas particuliers de réseaux de neurones ne comportant qu'une seule couche avec des fonctions d'activation linéaire en sortie des neurones. Cette simplification extrême rend possible une exploitation efficace du modèle pour le calcul des prédictions, et permet d'adapter les paramètres du modèle avec des critères du type moindres carrés bien plus performants que les méthodes utilisées pour l'apprentissage et l'adaptation des réseaux de neurones.

En se basant sur les méthodes d'identification paramétrique [BOR 92], et avec les extensions proposées pour s'affranchir des restrictions rencontrées habituellement, les algorithmes développés à partir d'un modèle linéaire peuvent s'appliquer à des processus non linéaires variant dans le temps. Des rappels préalables complétés par une analyse justifiant le choix d'un modèle linéaire discret sont nécessaires.

3.1. Modèle résultant des équations d'état

Un système linéaire continu multivariable mis sous forme d'équation d'état peut s'écrire :

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (1.1)$$

avec x l'état du système, u le vecteur de commande et y le vecteur de sortie.

Nous considérons pour la suite des développements la matrice D nulle. Après discrétisation et sous réserve que la période de discrétisation soit suffisamment petite, le modèle peut s'écrire sous la forme discrète:

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u_k \\ y_k &= C x_k \end{aligned} \quad (1.2)$$

Avec les matrices A_d et B_d obtenues par une approximation de la dérivée, ou en utilisant la transformée bilinéaire, ou encore en passant par la résolution des équations différentielles et par le calcul des exponentielles de matrices [GOL 89]. L'approche la plus simple consiste à effectuer l'approximation suivante :

$$\begin{aligned} A_d &= I + hA \\ B_d &= hB \end{aligned} \quad (1.3)$$

avec h la période d'échantillonnage.

Ne disposant pas du vecteur d'état x complet mais seulement des mesures y , le modèle discret que l'on souhaite identifier ne doit s'exprimer qu'en fonction des sorties et des commandes. Le changement de base peut s'obtenir entre autres en écrivant le modèle sur n instants et en procédant à une succession de substitutions dont l'obtention et l'écriture sont simplifiées en multipliant chaque équation par la puissance adéquate de la matrice A_d .

Chapitre I : Modélisation des Processus

$$\begin{aligned}
 x_{k+n} &= A_d x_{k+n-1} + B_d u_{k+n-1} & x_{k+n} &= A_d x_{k+n-1} + B_d u_{k+n-1} \\
 x_{k+n-1} &= A_d x_{k+n-2} + B_d u_{k+n-2} & A_d x_{k+n-1} &= A_d^2 x_{k+n-2} + A_d B_d u_{k+n-2} \\
 \vdots & & \text{d'où :} & \\
 x_{k+2} &= A_d x_{k+1} + B_d u_{k+1} & A_d^{n-1} x_{k+2} &= A_d^n x_{k+1} + A_d^{n-1} B_d u_{k+1} \\
 x_{k+1} &= A_d x_k + B_d u_k & A_d^n x_{k+1} &= A_d^{n+1} x_k + A_d^n B_d u_k
 \end{aligned} \tag{1.4}$$

L'utilisation des deux systèmes et la substitution des dernières équations vers les premières permet de déduire les relations suivantes :

$$\begin{aligned}
 x_{k+n} &= A_d^n x_k + \sum_{i=0}^{n-1} A_d^{n-1-i} B_d u_{k+i} \\
 x_{k+n-1} &= A_d^{n-1} x_k + \sum_{i=0}^{n-2} A_d^{n-2-i} B_d u_{k+i} \\
 \vdots & \\
 x_{k+2} &= A_d^2 x_k + A_d B_d u_k + B_d u_{k+1} \\
 x_{k+1} &= A_d x_k + B_d u_k \\
 x_k &= x_k
 \end{aligned} \tag{1.5}$$

La multiplication par la matrice d'observation C des équations aux instants $k, k+1, \dots, k+n-1$ fournit n nouvelles équations, dans lesquelles apparaissent isolément le vecteur d'état x_k et le vecteur des sorties $y_{k+n-1}, y_{k+n-2}, \dots, y_k$.

$$\begin{aligned}
 y_{k+n-1} &= CA_d^{n-1} x_k + \sum_{i=0}^{n-2} CA_d^{n-2-i} B_d u_{k+i} \\
 \vdots & \\
 y_{k+2} &= CA_d^2 x_k + CA_d B_d u_k + CB_d u_{k+1} \\
 y_{k+1} &= CA_d x_k + CB_d u_k \\
 y_k &= Cx_k
 \end{aligned} \tag{1.6}$$

d'où le système :

$$\begin{bmatrix} y_{k+n-1} \\ \vdots \\ y_{k+2} \\ y_{k+1} \\ y_k \end{bmatrix} = \begin{bmatrix} CA_d^{n-1} \\ \vdots \\ CA_d^2 \\ CA_d \\ C \end{bmatrix} x_k + \begin{bmatrix} CB_d & CA_d B_d & \dots & CA_d^{n-2} B_d \\ 0 & \ddots & & \vdots \\ \vdots & 0 & CB_d & CA_d B_d \\ 0 & & 0 & CB_d \\ 0 & & \dots & 0 \end{bmatrix} \begin{bmatrix} u_{k+n-2} \\ \vdots \\ u_{k+2} \\ u_{k+1} \\ u_k \end{bmatrix} \tag{1.7}$$

Nous retrouvons dans ce système la matrice d'observabilité du système discrétisé que l'on suppose de rang maximum [BOR 90], qui est donc inversible ou pseudo-inversible dans le cas multivariable. L'obtention d'une matrice singulière nous indiquerait que le système n'est pas observable. La conception du processus, le placement des capteurs ou l'agencement des actionneurs seraient alors à revoir. Nous excluons ce cas qui est supposé résolu lors de la conception du processus. Le vecteur x_k peut donc être isolé et s'écrit :

$$x_k = \begin{bmatrix} CA_d^{n-1} \\ \vdots \\ CA_d^2 \\ CA_d \\ C \end{bmatrix}^{-1} \begin{bmatrix} y_{k+n-1} \\ \vdots \\ y_{k+2} \\ y_{k+1} \\ y_k \end{bmatrix} - \begin{bmatrix} CA_d^{n-1} \\ \vdots \\ CA_d^2 \\ CA_d \\ C \end{bmatrix}^{-1} \begin{bmatrix} 0 & CB_d & \dots & CA_d^{n-2} B_d \\ \vdots & \vdots & \ddots & \vdots \\ 0 & CB_d & CA_d B_d & \vdots \\ \vdots & 0 & CB_d & \vdots \\ 0 & \dots & 0 & \vdots \end{bmatrix} \begin{bmatrix} u_{k+n-1} \\ \vdots \\ u_{k+2} \\ u_{k+1} \\ u_k \end{bmatrix} \quad (1.8)$$

En remplaçant x_k dans la première équation du système (1.5) et après multiplication par la matrice C , on déduit y_{k+n} :

$$y_{k+n} = CA_d^n \begin{bmatrix} CA_d^{n-1} \\ \vdots \\ CA_d^2 \\ CA_d \\ C \end{bmatrix}^{-1} \begin{bmatrix} y_{k+n-1} \\ \vdots \\ y_{k+2} \\ y_{k+1} \\ y_k \end{bmatrix} + \begin{bmatrix} CB_d & CA_d B_d & \dots & CA_d^{n-2} B_d & CA_d^{n-1} B_d \end{bmatrix} - CA_d^n \begin{bmatrix} CA_d^{n-1} \\ \vdots \\ CA_d^2 \\ CA_d \\ C \end{bmatrix}^{-1} \begin{bmatrix} 0 & CB_d & \dots & CA_d^{n-2} B_d \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & CB_d & CA_d B_d & \vdots \\ \vdots & 0 & CB_d & \vdots \\ 0 & \dots & 0 & \vdots \end{bmatrix} \begin{bmatrix} u_{k+n-1} \\ \vdots \\ u_{k+2} \\ u_{k+1} \\ u_k \end{bmatrix} \quad (1.9)$$

Le vecteur de sortie à l'instant $k+n$ se déduit à partir des valeurs des sorties et des commandes aux instants k à $k+n-1$. La modélisation du processus par un système linéaire discret exprimé dans l'espace de régression des sorties est donc possible. En décalant les indices de $-n+1$ pas, on aboutit à une expression entre le vecteur de sortie y_{k+1} et les vecteurs $y_k, y_{k-1}, \dots, y_{k-n+1}$, et $u_k, u_{k-1}, \dots, u_{k-n+1}$. L'écriture et les notations retenues pour la modélisation d'un processus avec un modèle multivariable linéaire discret sont :

$$Y_{k+1} = J_x X_k + J_u U_k \quad (1.10)$$

$$\text{avec } Y_{k+1} = y_{k+1}, X_k = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-n+1} \end{bmatrix} \text{ et } U_k = \begin{bmatrix} u_k \\ u_{k-1} \\ \vdots \\ u_{k-n+1} \end{bmatrix}$$

et avec J_x la matrice pondérant le vecteur X_k correspondant aux régressions du vecteur y , et J_u la matrice portant sur le vecteur régression U_k du vecteur des commandes u .

Avec ce modèle nous ne disposons pas de l'état x du processus dans l'espace qui peut être obtenu avec une modélisation suivant les lois physiques, mais nous avons son état reconstruit dans une autre base correspondant à l'espace des régressions des sorties. Le choix de cette base est immédiat

car le modèle ainsi obtenu découle directement des variables du processus accessibles à la mesure, qui correspondent généralement aux valeurs à commander.

3.2. Modèle résultant des fonctions de transfert

Les équations d'état et les fonctions de transfert correspondent à deux représentations différentes d'un même système, qui permettent chacune de mettre en avant certaines propriétés plus ou moins facilement. En effet, le formalisme avec les fonctions de transfert conduit à des résultats exprimés plus simplement pour aboutir au modèle en fonction des régressions des sorties du système. La conversion entre les équations d'état (1.1) et une fonction de transfert est donnée par la relation:

$$H(s) = C(sI - A)^{-1}B + D \quad (1.11)$$

ou encore

$$H(s) = C(sI - A)^{-1}B \text{ en considérant la matrice } D \text{ nulle.} \quad (1.12)$$

En monovarié, la fonction de transfert $H(s)$ entre la sortie y et l'entrée u peut s'écrire :

$$D(s)Y = N(s)U \quad (1.13)$$

avec $D(s)$ le déterminant de $(sI-A)^{-1}$ et $N(s)$ le numérateur de $H(s)$, le degré du numérateur étant inférieur au degré du dénominateur qui est supposé égal à n .

En multivarié, les fonctions de transfert obtenues ont le même dénominateur, mais possèdent des numérateurs distincts pour chacune des nu entrées et des ny sorties. Le modèle sous une forme non canonique peut donc s'écrire :

$$D(s) \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{ny} \end{bmatrix} = \begin{bmatrix} N_{1,1}(s) & N_{1,2}(s) & & N_{1,nu}(s) \\ N_{2,1}(s) & N_{2,2}(s) & & N_{2,nu}(s) \\ \vdots & & \ddots & \vdots \\ N_{ny,1}(s) & N_{ny,2}(s) & \cdots & N_{ny,nu}(s) \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{nu} \end{bmatrix} \quad (1.14)$$

Les polynômes sont discrétisés en prenant comme approximation de la dérivée s , l'opérateur discret $\frac{q-1}{h}$, où q désigne l'opérateur d'avance et h est la période de discrétisation. La conversion des numérateurs et dénominateurs en discret (N_d et D_d) se réalise alors par substitution des puissances de s par son approximation discrète, ce qui permet d'obtenir le modèle en fonction des régressions des sorties y aux instants $k+n$ à k , notées $y_{i,k+n,\dots,k}$, et des entrées u aux instants $k+n-1$ à k , notées $u_{i,k+n-1,\dots,k}$, avec n le degré de $D(s)$, et $n-1$ le degré maximal des numérateurs.

$$\begin{bmatrix} D_d & & & & \\ & D_d & & & \\ & & \ddots & & \\ & & & D_d & \end{bmatrix} \begin{bmatrix} y_{1,k+n,\dots,k} \\ y_{2,k+n,\dots,k} \\ \vdots \\ y_{ny,k+n,\dots,k} \end{bmatrix} = \begin{bmatrix} N_{d,1,1} & N_{d,1,2} & \cdots & N_{d,1,nu} \\ N_{d,2,1} & N_{d,2,2} & & \\ \vdots & & \ddots & \\ N_{d,ny,1} & & \cdots & N_{d,ny,nu} \end{bmatrix} \begin{bmatrix} u_{1,k+n-1,\dots,k} \\ u_{2,k+n-1,\dots,k} \\ \vdots \\ u_{nu,k+n-1,\dots,k} \end{bmatrix} \quad (1.15)$$

Le décalage des variables de $n-1$ pas, la réorganisation des matrices et le changement de l'ordre des variables dans les régressions donnent alors un modèle qui peut se mettre sous la forme :

$$\begin{bmatrix} y_{1,k+1} \\ y_{2,k+1} \\ \vdots \\ y_{ny,k+1} \end{bmatrix} = \begin{bmatrix} J_x & & & \\ & J_x & & \\ & & \ddots & \\ & & & J_x \end{bmatrix} \begin{bmatrix} y_{1,k,\dots,k-n+1} \\ y_{2,k,\dots,k-n+1} \\ \vdots \\ y_{ny,k,\dots,k-n+1} \end{bmatrix} + J_u \begin{bmatrix} u_k \\ u_{k-1} \\ \vdots \\ u_{k-n+1} \end{bmatrix} \quad (1.16)$$

avec J_x une matrice ligne affectant les vecteurs régressions de chaque variable y_1, y_2, \dots, y_{ny} , et avec J_u les coefficients pondérant le vecteur régression des commandes u .

En posant,

$$Y_{k+1} = \begin{bmatrix} y_{1,k+1} \\ y_{2,k+1} \\ \vdots \\ y_{ny,k+1} \end{bmatrix}, X_k = \begin{bmatrix} y_{1,k,\dots,k-n+1} \\ y_{2,k,\dots,k-n+1} \\ \vdots \\ y_{ny,k,\dots,k-n+1} \end{bmatrix} \text{ et } U_k = \begin{bmatrix} u_k \\ u_{k-1} \\ \vdots \\ u_{k-n+1} \end{bmatrix} \quad (1.17)$$

le modèle s'écrit :

$$Y_{k+1} = \begin{bmatrix} J_x & & & \\ & J_x & & \\ & & \ddots & \\ & & & J_x \end{bmatrix} X_k + J_u U_k \quad (1.18)$$

Le modèle non canonique déduit des fonctions de transfert simplifie le calcul des prédictions et de la commande, mais présente l'inconvénient d'être plus difficile à identifier à cause du nombre de paramètres de la matrice J_u qui est plus important qu'avec le modèle déduit à partir des équations d'état.

3.3. Détermination de la période de régression

La précision des modèles discrets obtenus après conversion des équations différentielles dépend de la période choisie. Les modèles continus et discrets peuvent être qualifiés d'équivalents s'ils ont le même comportement, c'est-à-dire si les réponses temporelles à n'importe quelle entrée sont

similaires. Nous rappelons ci-dessous les règles à respecter pour obtenir un modèle discret le plus proche possible du modèle continu.

Nous considérons le cas où le pas de discrétisation est unique et constant. L'utilisation d'un modèle avec un pas différent entre les régressions, ou avec un pas variable ou adaptatif permettrait de représenter plus finement le processus à commander. A priori le modèle ainsi déduit refléterait mieux le processus réel, et améliorerait les résultats. Mais d'un point de vue algorithmique, une telle modélisation causerait des difficultés supplémentaires. La robustesse et la stabilité ne seraient alors pas systématiquement renforcées. La commande risquerait au contraire d'être moins stable et plus sensible aux perturbations. Par souci d'efficacité et de simplicité, les modèles discrets utilisés par la suite sont donc tous à pas fixe.

Le choix de la période de discrétisation doit s'effectuer en tenant compte des dynamiques du processus, des bruits de mesure et de la précision à laquelle s'effectuent les calculs. Le théorème de Shannon [LON 95] nous donne une première condition à respecter mais celle-ci n'est pas suffisante, car l'influence du bruit et de la précision des mesures n'est pas prise en compte explicitement. Le théorème de Shannon indique seulement que la fréquence d'échantillonnage doit être supérieure au double de la fréquence maximale des signaux à étudier. Pour respecter cette condition, la période d'échantillonnage doit donc être suffisamment petite. Toutefois les dynamiques du système les plus rapides peuvent être négligées dans certaines circonstances. Il est alors non nécessaire de choisir un pas d'échantillonnage très petit.

La précision des calculs est aussi un paramètre à prendre en compte, car le nombre de chiffres significatifs nécessaires pour obtenir un modèle discret satisfaisant augmente lorsque la période décroît. Pour une précision des calculs donnée, la période doit être alors supérieure à un seuil. D'autre part, les mesures renvoyées par les capteurs sont erronées à cause du bruit et de la quantification des mesures. Or les variations de ces mesures servent à l'estimation des paramètres du modèle qui est donc d'autant plus faussée que les bruits sont importants par rapport aux variations réelles du processus. Afin de réduire la sensibilité des paramètres du modèle par rapport aux bruits, la période entre les régressions doit être suffisamment grande.

Pour résoudre le problème de la détermination de la période, il est plus pertinent de s'intéresser non plus directement aux fréquences des signaux mais à la précision du modèle désiré par rapport à la précision des mesures et à l'amplitude des bruits. L'objectif est de déterminer la plus grande valeur possible de la période permettant d'obtenir un modèle discret avec une erreur inférieure à un seuil fixé. L'approximation des matrices A_d et B_d par un développement du premier et du second ordre donne respectivement :

$$\begin{aligned} A_{d,1} &= I + hA & A_{d,2} &= I + hA + \frac{1}{2}h^2 AA \\ B_{d,1} &= hB & B_{d,2} &= hB + \frac{1}{2}h^2 AB \end{aligned} \quad \text{et} \quad (1.19)$$

avec h la période de régression et A et B les matrices du système (1.1).

Les conditions pour obtenir une bonne approximation du premier ordre peuvent s'exprimer de la manière suivante :

$$\begin{aligned} \frac{1}{2}h^2 \|AA\| &\ll h\|A\| \\ \frac{1}{2}h^2 \|AB\| &\ll h\|B\| \end{aligned} \quad (1.20)$$

avec $\|\cdot\|$ désignant une norme de matrice.

Pour les applications numériques, on choisit la norme de Frobenius [GOL 89] qui est la plus fréquemment utilisée et définie ainsi :

$$\text{Soit } X \in \mathbb{R}^{n \times m}, \quad \|X\|_F = \left(\sum_{i=1}^n \sum_{j=1}^m x_{ij}^2 \right)^{\frac{1}{2}}$$

La période h est choisie telle que les erreurs commises en négligeant les termes du second ordre représentent au maximum p pour-cent des termes du premier ordre, d'où les relations :

$$\begin{aligned} h\|AA\| &= 2p\|A\| \\ h\|AB\| &= 2p\|B\| \end{aligned} \quad (1.21)$$

Le paramètre p est choisi en fonction de la précision désirée par rapport à la précision des mesures et par rapport à l'amplitude des bruits. Pour des signaux fortement bruités, la valeur de p à choisir est de l'ordre de 0.1, le modèle n'est pas précis mais la présence de bruit ne permet pas d'obtenir de meilleurs résultats. Pour des valeurs de p de l'ordre de 0.01 le modèle obtenu est plus précis.

Une estimation de la valeur de h s'effectue alors en minimisant les erreurs e_A et e_B selon le critère quadratique G défini ci-dessous :

$$\begin{aligned} e_A &= h\|AA\| - 2p\|A\| \\ e_B &= h\|AB\| - 2p\|B\| \end{aligned} \quad (1.22)$$

et

$$G = \frac{1}{2}(e_A^2 + e_B^2) \quad (1.23)$$

La minimisation de ce critère donne :

$$h = 2p \frac{\|A\|\|AA\| + \|B\|\|AB\|}{\|AA\|\|AA\| + \|AB\|\|AB\|} \quad (1.24)$$

La matrice C du système a aussi une influence sur le calcul de la période h . Sa prise en compte s'effectue en analysant l'impact des erreurs d'approximation sur le vecteur de sortie y_{k+1} avec :

$$\begin{aligned}x_{k+1} &= A_d x_k + B_d u_k \\ y_{k+1} &= C x_{k+1}\end{aligned}$$

d'où

$$y_{k+1} = CA_d x_k + CB_d u_k$$

Pour éviter de commettre des erreurs d'approximation trop importantes lors de la conversion entre le modèle continu et discret, les conditions suivantes doivent être respectées :

$$\begin{aligned}\frac{1}{2}h^2 \|CAA\| &<< h \|CA\| \\ \frac{1}{2}h^2 \|CAB\| &<< h \|CB\|\end{aligned}\tag{1.25}$$

La même démarche que précédemment appliquée avec ces nouvelles contraintes indique la période h à choisir :

$$h = 2p \frac{\|CA\| \|CAA\| + \|CB\| \|CAB\|}{\|CAA\| \|CAA\| + \|CAB\| \|CAB\|}\tag{1.26}$$

Cette approche repose sur l'étude du compromis entre la précision et la sensibilité du modèle discret. Le choix d'une période convenable pour que l'identification et la commande donnent des résultats satisfaisants, s'obtient alors à partir de la connaissance des dynamiques du système, de la précision désirée et de la précision des mesures.

3.4. Modèle sous-échantillonné

Pour des systèmes à dynamiques lentes et à caractère instable, la période h d'acquisition des mesures et de calcul de la commande doit être assez courte pour éviter que le système ne s'écarte trop de la consigne entre deux instants d'échantillonnage, et aussi afin d'obtenir une mise à jour du modèle rapide. Cependant un bon compromis entre la précision du modèle et sa sensibilité aux bruits est obtenu avec une période de régression plus importante. Ce dilemme peut être résolu en définissant un facteur de sous échantillonnage nbs . Le modèle utilisé est alors défini de la façon suivante :

$$\frac{1}{nbs}(y_{k+nbs} + y_{k+nbs-1} + \dots + y_{k+1}) = \frac{1}{nbs} J_x \begin{bmatrix} y_k + y_{k-1} + \dots + y_{k-nbs+1} \\ y_{k-nbs} + y_{k-nbs-1} + \dots + y_{k-2nbs+1} \\ \vdots \\ y_{k-(nbry-1)nbs} + y_{k-(nbry-1)nbs-1} + \dots + y_{k-nbry*nbs+1} \end{bmatrix} \quad (1.27)$$

$$+ \frac{1}{nbs} J_u \begin{bmatrix} u_k + u_{k-1} + \dots + u_{k-nbs+1} \\ u_{k-nbs} + u_{k-nbs-1} + \dots + u_{k-2nbs+1} \\ \vdots \\ u_{k-(nbru-1)nbs} + u_{k-(nbru-1)nbs-1} + \dots + u_{k-nbru*nbs+1} \end{bmatrix}$$

avec $nbry$ et $nbru$ le nombre de régressions utilisées pour les vecteurs de sorties y et de commandes u , et nbs le facteur de sous échantillonnage du modèle.

La moyenne des mesures y sur nbs instants à partir d'un instant k est notée $\bar{y}_{nbs,k}$. Avec cette convention d'écriture, le modèle devient :

$$\bar{y}_{nbs,k+1} = J_x \begin{bmatrix} \bar{y}_{nbs,k-nbs+1} \\ \bar{y}_{nbs,k-2nbs+1} \\ \vdots \\ \bar{y}_{nbs,k-nbry*nbs+1} \end{bmatrix} + J_u \begin{bmatrix} \bar{u}_{nbs,k-nbs+1} \\ \bar{u}_{nbs,k-2nbs+1} \\ \vdots \\ \bar{u}_{nbs,k-nbru*nbs+1} \end{bmatrix} \quad (1.28)$$

$$(1.29)$$

que l'on note plus simplement :

$$\bar{Y}_{k+1} = J_x \bar{X}_k + J_u \bar{U}_k \quad (1.30)$$

Le passage du modèle (1.10) vers le modèle sous échantillonné s'effectue donc par un calcul de la moyenne des mesures sur l'intervalle de temps adéquat. Les calculs relatifs à l'identification et à la commande sont cadencés à une fréquence $1/h$ mais la période de régression du modèle vaut $nbs h$. La présence de dynamiques rapides négligées peut biaiser les résultats et constitue un risque d'instabilité et une limite quant à l'utilisation de ce modèle. En pratique, le facteur nbs ne doit pas être trop important ($nbs < 10$).

Les avantages de cette approche sont premièrement l'obtention d'un modèle avec une période appropriée par rapport aux dynamiques dominantes du processus, deuxièmement le filtrage des mesures réalisé lors du calcul de la moyenne qui réduit la sensibilité du modèle vis à vis des bruits. Enfin, le temps total de mise à jour du modèle est raccourci grâce à la fréquence d'acquisition qui est plus élevée par rapport à celle du modèle.

3.5. Exemple

L'exemple traité dans cette section a pour rôle d'illustrer le principe de conversion du système continu vers le modèle discret, de comparer les méthodes et de vérifier la validité des modèles. Le système continu considéré est composé de quatre variables d'état, de deux sorties y_1, y_2 et de deux commandes couplées u_1, u_2 .

$$\begin{cases} A = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -2 & -5 & -2 & 0 \\ -1 & 0 & 0 & 1 \\ -1 & -1 & -12 & -2 \end{bmatrix}, B = \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 2 \end{bmatrix} \\ C = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{cases} \quad (1.31)$$

La détermination de la période d'échantillonnage s'effectue en appliquant la formule (1.26) avec un seuil d'erreur p égal à 0.05. Avec ces paramètres, le résultat h vaut 0.0169 s. La période de régression est alors choisie égale à $h = 0.02$ s.

Les matrices discrétisées A_d et B_d du système continu A, B, C avec un pas $h=0.02$ valent :

$$A_d = \begin{bmatrix} 1 & 0.02 & 0 & -0.02 \\ -0.04 & 0.9 & -0.04 & 0 \\ -0.02 & 0 & 1 & 0.02 \\ -0.02 & -0.02 & -0.24 & 0.96 \end{bmatrix}, B_d = \begin{bmatrix} -0.02 & 0.02 \\ 0.02 & 0 \\ 0 & 0.02 \\ -0.02 & 0.04 \end{bmatrix} \quad (1.32)$$

La formule de conversion (1.9) appliquée à cet exemple donne le modèle :

$$\begin{bmatrix} y_{1,k+1} \\ y_{2,k+1} \end{bmatrix} = CA_d^2 \begin{bmatrix} CA_d \\ C \end{bmatrix}^{-1} \begin{bmatrix} y_{1,k} \\ y_{2,k} \\ y_{1,k-1} \\ y_{2,k-1} \end{bmatrix} + \left(C \begin{bmatrix} B_d & A_d B_d \end{bmatrix} - CA_d^2 \begin{bmatrix} CA_d \\ C \end{bmatrix}^{-1} \begin{bmatrix} 0 & CB_d \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} u_{1,k} \\ u_{2,k} \\ u_{1,k-1} \\ u_{2,k-1} \end{bmatrix} \quad (1.33)$$

dont l'application numérique est :

$$\begin{bmatrix} y_{1,k+1} \\ y_{2,k+1} \end{bmatrix} = \begin{bmatrix} 1.88 & 0.02 & -0.8828 & -0.0180 \\ -0.04 & 1.98 & 0.0384 & -0.9832 \end{bmatrix} \begin{bmatrix} y_{1,k} \\ y_{2,k} \\ y_{1,k-1} \\ y_{2,k-1} \end{bmatrix} + \begin{bmatrix} -0.02 & 0.04 & 0.0184 & -0.0360 \\ 0 & 0.02 & -0.0008 & -0.0176 \end{bmatrix} \begin{bmatrix} u_{1,k} \\ u_{2,k} \\ u_{1,k-1} \\ u_{2,k-1} \end{bmatrix} \quad (1.34)$$

L'utilisation des fonctions de transfert permet d'aboutir aux modèles continu et discret ci-dessous :

$$(s^4 + 7s^3 + 23s^2 + 69s + 76) \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} -s^3 - 5s^2 - 14s - 42 & 2s^3 + 13s^2 + 37s + 110 \\ 2 & s^3 + 8s^2 + 15s + 2 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \quad (1.35)$$

et

$$\begin{bmatrix} y_{1,k+1} \\ y_{2,k+1} \end{bmatrix} = \begin{bmatrix} 3.86 & -5.5892 & 3.5978 & -0.86866 \\ & & 0 & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} y_{1,k} \\ y_{1,k-1} \\ y_{1,k-2} \\ y_{1,k-3} \end{bmatrix} + \begin{bmatrix} 0 \\ 3.86 & -5.5892 & 3.5978 & -0.86866 \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} y_{2,k} \\ y_{2,k-1} \\ y_{2,k-2} \\ y_{2,k-3} \end{bmatrix} + \begin{bmatrix} u_{1,k} \\ u_{2,k} \\ u_{1,k-1} \\ u_{2,k-1} \\ u_{1,k-2} \\ u_{2,k-2} \\ u_{1,k-3} \\ u_{2,k-3} \end{bmatrix} \quad (1.36)$$

L validation des modèles s'effectue en comparant les réponses des modèles continus et discrets en appliquant des entrées quelconques sur les systèmes :

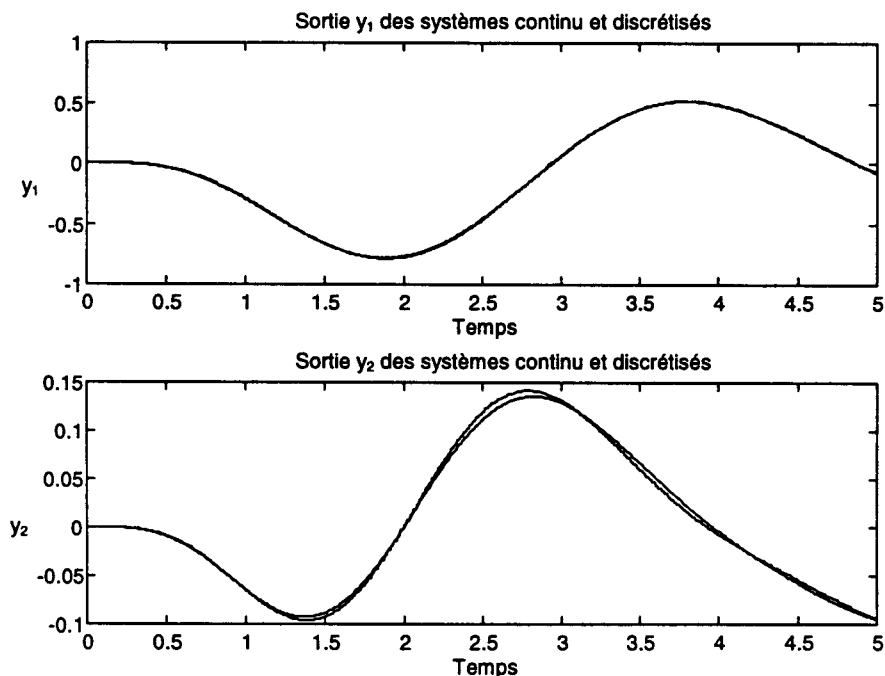


Figure 1.1 : Réponses des modèles continu et discrets.

L'approximation d'un système continu linéaire et invariant (1.1) par un modèle linéaire discret dans l'espace des sorties est structurellement possible. Si l'adaptation des coefficients s'effectue de manière adéquate, et si le choix du nombre de régressions et de la période est pertinent alors le modèle obtenu est une approximation satisfaisante du comportement du processus, et convient pour le calcul de la commande. En identifiant le processus en ligne, on réalise un observateur du processus et de son état. L'avantage que l'on tire de cette approche est donc double : nous disposons

après adaptation non seulement d'un modèle du processus mais aussi d'une reconstruction de son état dans une base obtenue directement à partir des régressions des mesures. L'observateur ainsi défini se déduit à partir des variables accessibles à la mesure et s'avère ensuite adéquat pour la détermination d'une loi de commande. L'aptitude du modèle à pouvoir représenter fidèlement le comportement du processus est le gage de réussite de l'identification et de la commande du processus.

4. Généralisation à des systèmes non linéaires

4.1. Approximation linéaire

Nous envisageons dans un premier temps d'étendre l'étude à des systèmes non linéaires supposés dérivables en tout point. Ces non linéarités correspondent à des changements progressifs de la dynamique du processus et des valeurs de ses paramètres. Pour des systèmes physiques, cette sorte de non linéarité est très courante. Par exemple, la charge à la sortie d'un moteur peut être non linéaire et se comporter comme une résistance proportionnelle au carré de la vitesse. Pour chaque vitesse de rotation donnée, le processus regroupant le moteur et la charge réagit selon une certaine dynamique. Dans cet exemple les matrices d'évolution A du système et d'application B de la commande sont dépendantes de la vitesse du moteur, qui est une des composantes du vecteur d'état. La représentation générale sous forme d'équations d'état de ce type de systèmes non linéaires est alors la suivante :

$$\begin{aligned} \dot{x} &= A(x)x + B(x)u \\ y &= Cx \end{aligned} \tag{1.37}$$

En supposant le système dérivable en tout point, nous pouvons considérer le système linéaire autour d'un point de fonctionnement x_0 .

$$\begin{aligned} \dot{x} &= A_{x_0} x + B_{x_0} u \\ y &= Cx \end{aligned} \tag{1.38}$$

Avec cette hypothèse, l'identification du système non linéaire se ramène à l'identification d'un modèle linéaire. En suivant le même principe, un processus dont les caractéristiques varient avec le temps peut s'écrire sous forme d'équations d'état avec les matrices A et B dépendant du temps.

$$\begin{aligned} \dot{x} &= A(x,t)x + B(x,t)u \\ y &= Cx \end{aligned} \tag{1.39}$$

Nous supposons ces variations lentes par rapport à la dynamique du système, ou le cas échéant de faibles amplitudes par rapport aux valeurs des paramètres. A un instant donné t_0 nous pouvons donc considérer que le système a un comportement linéaire ou quasi linéaire.

$$\begin{aligned} \dot{x} &= A_{x_0, t_0} x + B_{x_0, t_0} u \\ y &= Cx \end{aligned} \tag{1.40}$$

Pour des variations brusques des caractéristiques ayant lieu seulement à un instant donné, le modèle obtenu est inexact pendant le temps nécessaire pour que le modèle converge vers les nouveaux paramètres. Ce régime transitoire doit être le plus court possible afin d'obtenir des performances répondant au mieux au cahier des charges. Cette condition pour être satisfaite met en évidence l'importance de la rapidité de l'algorithme d'adaptation.

4.2. Définition d'un modèle local

L'étude précédente montre que l'approximation linéaire n'est valable que localement dans le domaine dans lequel le processus est utilisé, et seulement sur un intervalle de temps dépendant de la vitesse des variations des paramètres du système. Pour ces raisons, les modèles linéaires sont définis et corrects seulement autour d'un point de fonctionnement.

L'intervalle de temps pendant lequel l'approximation est valide, et le domaine dans lequel l'approximation par le système linéarisé est satisfaisante est appelé domaine de validité du modèle.

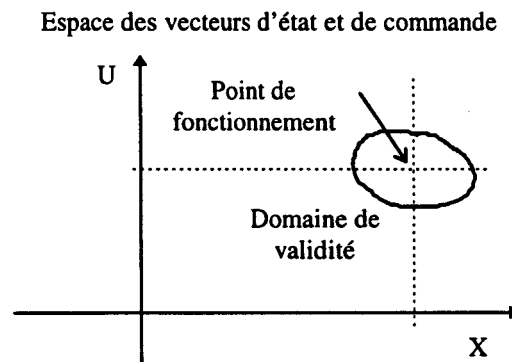


Figure 1.2 : Domaine de validité

Dans le cas des systèmes linéaires invariants, le domaine de validité correspond à l'espace entier. En revanche, l'accroissement des non linéarités du processus et l'augmentation des variations temporelles des paramètres diminuent le domaine de validité ce qui constitue la principale limite pour l'application de cette méthode à des systèmes fortement non linéaires. Dans ce dernier cas, il est conseillé d'utiliser des techniques de linéarisation, transformant un système non linéaire en un système linéaire exprimé avec de nouvelles variables d'état. Les méthodes d'identification et les lois de commande proposées s'appliquent alors sans difficulté. Cette transformation n'est cependant pas toujours possible et requiert une connaissance fine du processus et de son modèle.

4.3. Ecriture des modèles dans un repère local

Les modèles autour d'un point de fonctionnement s'obtiennent par un changement de repère en calculant les variations des points (Y_k , X_k et U_k) par rapport au point de fonctionnement noté (Y_{ope} , X_{ope} et U_{ope}) qui est déterminé lors de l'adaptation du modèle.

Les modèles précédents (1.10), (1.30) et (1.18) s'écrivent alors respectivement :

$$Y_{k+1} - Y_{ope} = J_x (X_k - X_{ope}) + J_u (U_k - U_{ope}) \quad (1.41)$$

$$\bar{Y}_{k+1} - \bar{Y}_{ope} = J_x (\bar{X}_k - \bar{X}_{ope}) + J_u (\bar{U}_k - \bar{U}_{ope}) \quad (1.42)$$

$$Y_{k+1} - Y_{ope} = \begin{bmatrix} J_x & & & \\ & J_x & & \\ & & \ddots & \\ & & & J_x \end{bmatrix} (X_k - X_{ope}) + J_u (U_k - U_{ope}) \quad (1.43)$$

Pour simplifier l'écriture des modèles, on pose :

$$\Delta Y = Y_{k+1} - Y_{ope}, \Delta X = X_k - X_{ope} \text{ et } \Delta U = U_k - U_{ope}. \quad (1.44)$$

où le symbole Δ indique que les mesures sont exprimées dans le repère local. Les modèles s'écrivent alors :

$$\Delta Y = J_x \Delta X + J_u \Delta U \quad (1.45)$$

$$\Delta \bar{Y} = J_x \Delta \bar{X} + J_u \Delta \bar{U} \quad (1.46)$$

$$\Delta Y = \begin{bmatrix} J_x & & & \\ & J_x & & \\ & & \ddots & \\ & & & J_x \end{bmatrix} \Delta X + J_u \Delta U \quad (1.47)$$

Suivant les calculs à effectuer, l'expression de ces modèles est plus intéressante sous la forme :

$$Y_{k+1} = J_x X_k + J_u U_k + v_{ope} \quad (1.48)$$

$$\bar{Y}_{k+1} = J_x \bar{X}_k + J_u \bar{U}_k + v_{ope} \quad (1.49)$$

$$Y_{k+1} = \begin{bmatrix} J_x & & & \\ & J_x & & \\ & & \ddots & \\ & & & J_x \end{bmatrix} X_k + J_u U_k + v_{ope} \quad (1.50)$$

avec v_{ope} un vecteur dépendant du point fonctionnement.

Ces modèles locaux conviennent donc pour donner une approximation d'un grand nombre de processus, dont le point de fonctionnement est différent de l'origine. Ces modèles sont donc des cas plus généraux regroupant les classes précédentes de modèles. Par conséquent, l'identification et la commande sont réalisées ensuite à partir de ces modèles.

5. Extension à certaines classes particulières de systèmes

5.1. Modélisation de systèmes à retards purs

De part leur conception, certains systèmes comportent des retards dont la présence est incontournable. Les sorties du système réagissent alors à une entrée après un certain laps de temps. Ces retards sont dus soit au positionnement des capteurs, soit à des délais de transmission des mesures et des commandes pour des processus commandés à distance. Le cas typique mais non trivial à l'origine de retards se présente lorsqu'un transfert de matière ou un écoulement de fluide a lieu dans un processus. Les commandes agissent par exemple sur le système à une position donnée mais la mesure s'effectue à une certaine distance. Le retard est alors égal au temps nécessaire pour que la matière parcourt la distance séparant le capteur de l'actionneur. Ce retard dépend de la vitesse du flux de matière qui est supposée variable a priori.

Ces quelques remarques montrent que ces retards influent aussi bien sur les mesures que sur les signaux de commande, et qu'ils peuvent être variables. Ils se modélisent pour des systèmes continus représentés par des fonctions de transfert par l'ajout au numérateur de l'expression e^{-Tp} , avec p la variable de Laplace et T le retard. Le passage dans le domaine fréquentiel nous fournit l'équation du retard $e^{-j\omega T}$. Le déphasage induit par celui-ci est proportionnel au terme ωT , ce qui dégrade fortement les marges de gain et de phase. La prise en compte des retards dans le modèle est donc déterminante pour garantir une bonne stabilité de la loi de commande. Si en continu la modélisation d'un retard est plutôt peu commode et nécessite souvent l'usage de son approximation par une fonction de Padé, en discret la modélisation est immédiate et consiste à augmenter l'ordre du système jusqu'à ce que l'intervalle de temps défini par le vecteur régression englobe le retard. Toutefois, l'ordre du modèle ne peut pas être choisi arbitrairement grand car son augmentation alourdit les calculs et entraîne des difficultés pour l'identification notamment à cause des grandes dimensions des matrices et des systèmes à résoudre. Il convient alors d'accroître dans la mesure du possible la période de régression en respectant les conditions établies dans la section précédente pour pouvoir à la fois introduire le retard dans le modèle et en diminuer l'ordre.

La présence de retard se caractérise alors par le décalage des coefficients de la matrice J_u vers les termes d'ordre plus élevé. L'exemple ci-dessous illustre le principe et les conséquences du retard sur l'identification.

Soit le système défini par l'équation récurrente :

$$y_{k+1} = ay_k + bu_k \quad \text{dont la période de régression est } h.$$

En appliquant un retard de $5h$ sur la sortie ou sur la commande, le système s'écrit :

$$y_{k+1} = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_k \\ y_{k-1} \\ y_{k-2} \\ y_{k-3} \\ y_{k-4} \\ y_{k-5} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & b \end{bmatrix} \begin{bmatrix} u_k \\ u_{k-1} \\ u_{k-2} \\ u_{k-3} \\ u_{k-4} \\ u_{k-5} \end{bmatrix} \quad (1.51)$$

Les effets d'un retard sur des systèmes multivariables d'ordre plus élevé sont similaires. La prise en compte des retards est donc réalisable en augmentant l'ordre du modèle à adapter. La loi de commande qui en résulte doit alors exploiter judicieusement ce modèle pour aboutir à des performances satisfaisantes.

5.2. Modélisation de systèmes bilinéaires

La structure de cette famille de systèmes est particulière puisque le produit des variables d'état par les commandes intervient dans les équations d'état. Ce type de systèmes est fréquent pour des machines électriques à commutation. Seule l'identification de ce genre de systèmes est traitée. Les prédictions et la commande de ce genre de systèmes sont particulières et ne sont donc pas développées.

Le modèle continu s'écrit sous la forme :

$$\begin{aligned} \dot{x} &= Ax + Bu + \left(\sum_{i=1}^{nu} N_i u_i \right) x \\ y &= Cx \end{aligned} \quad (1.52)$$

avec N_i les matrices pondérant chacun des produits $u_i x$.

On peut remarquer la présence de points singuliers, lorsque les termes B et $\Sigma(N_i x)$ s'annulent. En ces points, la commandabilité est donc critique et induit des difficultés lors du calcul de la commande.

En posant $\dot{x} \approx \frac{x_{k+1} - x_k}{h}$, la discrétisation de l'équation différentielle donne une équation du type :

$$x_{k+1} = A_d x_k + B_d u_k + \left(\sum_{i=1}^{nu} N_{i,d} u_{i,k} \right) x_k \quad (1.53)$$

Le modèle discret est alors supposé être de la forme.

$$Y_{k+1} = J_x X_k + J_u U_k + J_{xu} XU_k \quad (1.54)$$

avec

$$Y_{k+1} = [y_{k+1}], X_k = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-n} \end{bmatrix}, U_k = \begin{bmatrix} u_k \\ u_{k-1} \\ \vdots \\ u_{k-n} \end{bmatrix}, XU_k = \begin{bmatrix} xu_k \\ xu_{k-1} \\ \vdots \\ xu_{k-n} \end{bmatrix} \text{ et } xu_k = \begin{bmatrix} x_k u_{1,k} \\ x_k u_{2,k} \\ \vdots \\ x_k u_{nu,k} \end{bmatrix} \quad (1.55)$$

Nous supposons également que la structure du modèle dont nous cherchons à déterminer les paramètres J_x, J_u, J_{xu} est appropriée par rapport au processus. Cette hypothèse est nécessaire pour obtenir un modèle traduisant correctement le comportement du processus. L'ordre du système et le nombre de paramètres à adapter étant plus importants qu'avec un système linéaire (1.10), le nombre de points à utiliser pour l'identification se trouve naturellement augmenté. Les méthodes d'identification restent cependant inchangées.

5.3. Modélisation de systèmes linéaires par rapport aux paramètres à adapter

L'extension des résultats pour certaines classes de systèmes dont la structure est connue mais les paramètres sont indéterminés, est réalisable si le processus peut se modéliser sous la forme :

$$y_{k+1} = \sum_i J_i f_i(y_k, y_{k-1}, \dots, y_{k-n}, u_k, u_{k-1}, \dots, u_{k-n}) \quad (1.56)$$

avec

f_i des fonctions connues des commandes et sorties aux instants k à $k-n$.

et J_i les matrices ou paramètres à adapter.

Le modèle doit être une fonction linéaire par rapport aux paramètres à adapter, de sorte que les algorithmes d'adaptation du type moindres carrés étudiés au chapitre suivant puissent s'appliquer. Si cette condition est respectée, aucune contrainte a priori ne nous empêche de généraliser les résultats du linéaire pour identifier d'autres modèles plus complexes. Cependant, pour la détermination des prédictions et de la loi de commande, de sévères limitations pratiques nous restreignent à des types particuliers de modèles. Nous nous limitons donc à l'identification de ces

processus, sans aborder le calcul des prédictions et de la loi de commande. Néanmoins, ces calculs sont réalisables pour certains types de non linéarités.

6. Modélisation de zones inactives

6.1. Description et choix du modèle

Cette section se destine plus particulièrement à la modélisation de processus mécaniques commandés par des moteurs électriques ou des actionneurs présentant une zone inactive. Cette zone correspond à un intervalle dans lequel la commande n'a aucune action sur le processus. Les résultats obtenus peuvent également s'appliquer à tout autre système dont les commandes n'agissent sur le processus qu'au delà d'un certain seuil.

La zone inactive des actionneurs est un type particulier de discontinuité qui est due par exemple aux frottements secs entre les pièces, dans les engrenages, ou entre toutes autres parties mobiles [POP 94], [AMI 97]. La mise en mouvement du mécanisme nécessite alors des efforts ou couples supérieurs à un seuil qui n'est atteint qu'à partir d'une valeur de commande suffisante. Concrètement, la tension à appliquer aux moteurs électriques doit être supérieure au seuil pour que le moteur fournisse le couple nécessaire. Ces phénomènes sont fréquemment rencontrés pour des commandes d'axes, des problèmes d'asservissement de position, de contrôle de robots ou encore pour une multitude de divers systèmes mécaniques, comme celui étudié au cours du cinquième chapitre qui est un processus où les zones inactives ont véritablement une forte influence.

Cette zone inactive si elle n'est pas compensée entraîne une sérieuse dégradation des performances. La précision est moins bonne, mais surtout des oscillations et un dépassement important apparaissent. A très basse vitesse et lors des changements de sens de la commande, les problèmes liés à la présence de la zone inactive se manifestent. Leurs conséquences sont néfastes pour l'adaptation des paramètres du modèle linéaire. En effet dans la zone inactive, la commande bien que non nulle n'a aucune influence sur les sorties du processus. Les termes portant sur la commande voient leur valeur diminuée, et sont sous-estimés. La commande déduite de ce modèle a tendance alors à être surestimée, ce qui provoque des variations importantes des commandes et par conséquence des secousses sur les actionneurs et des oscillations du mécanisme. Une telle commande génère une fatigue des composants et une usure prématurée de l'ensemble.

La première solution pour remédier à ce problème est de compenser la zone inactive, en appliquant sur la commande une fonction de décalage (Offset). Cette méthode convient si les valeurs des seuils sont connues et fixes. Mais leur détermination précise n'est pas toujours possible aisément. De plus ces seuils peuvent évoluer à cause du vieillissement des composants, ou si le processus est utilisé dans des conditions variables. Une compensation approximative ne peut généralement pas être acceptée car une sous-évaluation des seuils conduit aux mêmes problèmes que ceux évoqués précédemment. La sous-estimation des seuils provoque des oscillations et dégrade la précision obtenue. Les conséquences sont certes atténuées mais subsistent. La sur-évaluation de la zone inactive a des effets tout aussi gênants, car les actionneurs sont dans ce cas toujours sollicités et agissent en permanence sur le processus, qui ne peut pas être stabilisé autour d'une consigne constante. Des oscillations se produisent sans cesse autour de la valeur désirée.

La prise en compte de cette discontinuité directement dans le modèle et dans la loi de commande apporte donc un gain en précision et en performance non négligeable. En plus du meilleur suivi de trajectoire, les efforts transmis au processus sont réduits, et les secousses dégradant le mécanisme sont en partie évitées.

Pour ces raisons l'étude de ce phénomène et la compensation adaptative de la zone inactive se révèlent intéressantes. Ce type de processus peut se représenter en deux sous ensembles, l'un pour la dynamique du processus, l'autre pour la zone inactive. La dynamique du processus est modélisé par des hyperplans suivant le même principe que précédemment. L'unique modification du modèle est donc due à la présence d'une non linéarité que nous cherchons à identifier puis à compenser.

Le schéma regroupant le compensateur, la zone inactive et le processus est le suivant :

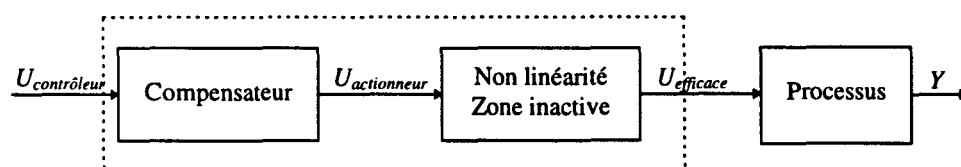


Figure 1.3 : Schéma du processus avec une non linéarité compensée.

La commande générée par le contrôleur est appelée $U_{\text{contrôleur}}$. Après compensation, nous obtenons la commande $U_{\text{actionneur}}$ qui correspond à la commande appliquée aux actionneurs, et qui doit être supérieure à un seuil d'amplitude f pour être efficace et agir sur le processus. La relation entre la commande appliquée et celle efficace (active) se modélise par les fonctions suivantes :

$$\begin{cases} U_{efficace} = U_{actionneur} + f & \text{si } U_{actionneur} < -f \\ U_{efficace} = 0 & \text{si } -f \leq U_{actionneur} \leq f \\ U_{efficace} = U_{actionneur} - f & \text{si } U_{actionneur} > f \end{cases} \quad (1.57)$$

La compensation qui doit être effectuée en sortie du contrôleur consiste à décaler la valeur de la commande de f ou $-f$, de manière à ce que la zone inactive soit évitée, et que la discontinuité disparaisse.

$$\begin{cases} U_{actionneur} = U_{contrôleur} - f & \text{si } U_{contrôleur} < 0 \\ U_{actionneur} = U_{contrôleur} + f & \text{si } U_{contrôleur} \geq 0 \end{cases} \quad (1.58)$$

On obtient ainsi une compensation parfaite, c'est à dire une relation linéaire sans discontinuité entre la commande calculée par le contrôleur et la commande efficace.

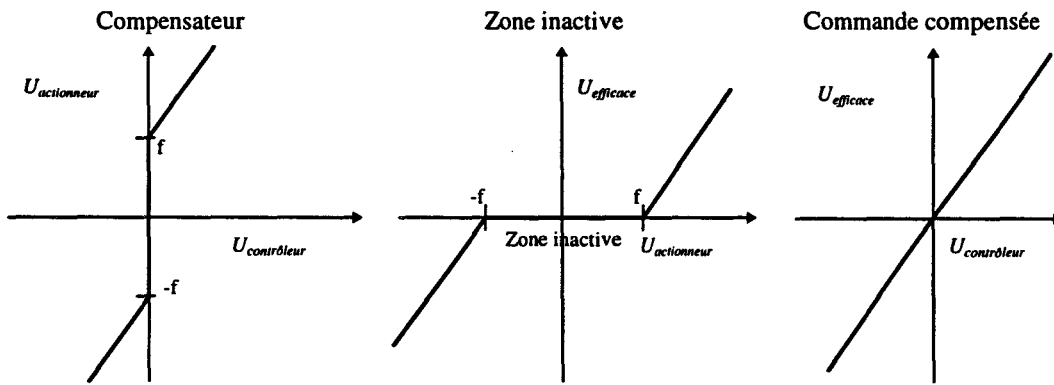


Figure 1.4 : Modélisation du compensateur, de la zone inactive et de l'ensemble compensé

La détermination de la valeur de f est la partie la plus délicate dont dépend la qualité de la compensation et donc de la loi de commande. Le modèle ci-dessus de la zone inactive ne permettant pas une détermination aisée de f , nous sommes amenés à définir un autre modèle qui doit être le plus proche possible du phénomène réel mais exploitable. Le modèle sélectionné répondant à ces deux conditions s'écrit :

$$\begin{cases} U_{efficace} = U_{actionneur} + f & \text{si } U_{actionneur} < 0 \\ U_{efficace} = 0 & \text{si } U_{actionneur} = 0 \\ U_{efficace} = U_{actionneur} - f & \text{si } U_{actionneur} > 0 \end{cases} \quad (1.59)$$

ou encore plus simplement

$$U_{efficace} = U_{actionneur} - f \operatorname{signe}(U_{actionneur}) \quad (1.60)$$

La zone centrale inactive est donc modélisée par l'ajout ou la soustraction de la valeur du seuil f suivant le signe de la commande $U_{actionneur}$.

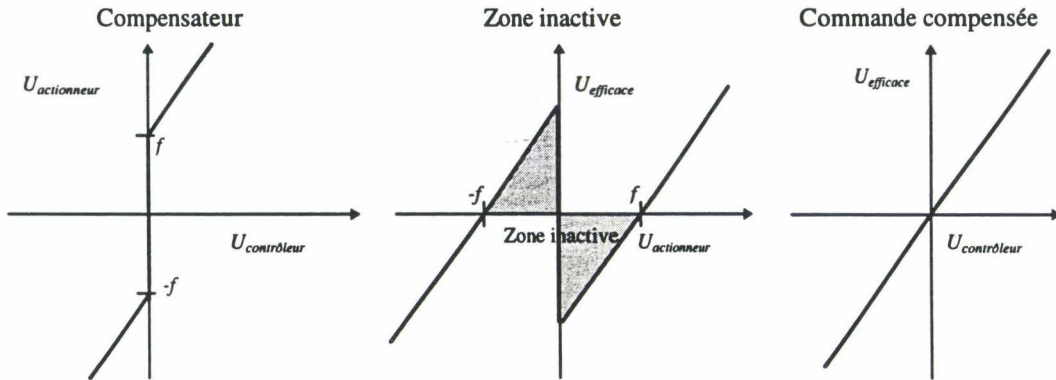


Figure 1.5 : Modélisation du compensateur, de la zone inactive modifiée et de l'ensemble compensé

Ce changement de modèle ne modifie pas le calcul de la compensation qui est toujours définie de manière à éviter l'utilisation des actionneurs dans la zone inactive. La relation entre la commande fournie par le contrôleur et la commande appliquée après compensation est linéaire et continue. Cependant, ce modèle n'est valide qu'en dehors de la zone inactive. Cette limite de validité n'est pas contraignante puisque le compensateur a pour objectif de générer une commande hors de cette zone. En cas de sous estimation de f la commande appliquée peut être insuffisante, mais l'algorithme d'adaptation du modèle a tendance alors à diminuer les valeurs des termes pondérant la commande et à augmenter la valeur de f . La commande aux instants suivants est donc de plus grande amplitude et finit par devenir supérieure au vrai seuil de la zone inactive. En cas de surestimation de f , l'adaptation des coefficients du modèle entraîne une diminution de f dont l'estimation se trouve alors affinée. L'identification, le calcul de la commande et la compensation se déroulent ensuite normalement selon les algorithmes établis dans les chapitres II, III et IV.

6.2. Estimation de la zone inactive

Le modèle complet du système regroupant le processus et la non linéarité est le suivant :

$$Y_{k+1} = J_x X_k + J_u (U_k - F \text{signe}(U_k)) \quad (1.61)$$

avec la matrice F , une matrice diagonale constituée de $nbru$ fois la matrice diagonale dont les termes correspondent aux valeurs f_i des zones inactives de chacune des commandes.

$$F = \begin{bmatrix} F_0 & & \\ & \ddots & \\ & & F_0 \end{bmatrix} \text{ et } F_0 = \begin{bmatrix} f_1 & & & \\ & \ddots & & \\ & & f_i & \\ & & & \ddots \\ & & & & f_{nbu} \end{bmatrix} \quad (1.62)$$

Cependant, le modèle ainsi écrit ne permet pas de calculer directement à la fois les matrices F et J_u . Le développement de cette expression conduit au modèle ci-dessous dont le produit de matrices $J_u F$ est remplacé par une seule matrice de dimension cohérente nommée J_{uf} .

$$Y_{k+1} = J_x X_k + J_u U_k + J_{uf} \text{signe}(U_k) \quad (1.63)$$

Le problème revient à calculer les matrices J_x , J_u et J_{uf} . La matrice F à déterminer initialement s'obtient alors à partir des matrices J_u et J_{uf} en calculant la solution optimale satisfaisant la relation $-J_u F = J_{uf}$.

Avec cette modélisation de la zone inactive, et en nommant SU_k le vecteur composé des valeurs des signes du vecteur U_k le modèle à identifier est linéaire et s'écrit :

$$Y_{k+1} = J_x X_k + J_u U_k + J_{uf} SU_k \quad (1.64)$$

Dans un repère local autour du point de fonctionnement, le modèle devient :

$$(Y_{k+1} - Y_{ope}) = J_x (X_k - X_{ope}) + J_u (U_k - U_{ope}) + J_{uf} (SU_k - SU_{ope}) \quad (1.65)$$

$$\text{avec } X_k = \begin{bmatrix} Y_k \\ \vdots \\ Y_{k-nbry+1} \end{bmatrix}, U_k = \begin{bmatrix} u_k \\ \vdots \\ u_{k-nbru+1} \end{bmatrix} \text{ et } SU_k = \begin{bmatrix} \text{signe}(u_k) \\ \vdots \\ \text{signe}(u_{k-nbru+1}) \end{bmatrix}$$

En adoptant le même principe de notations que précédemment, le modèle s'écrit :

$$\Delta Y = J_x \Delta X + J_u \Delta U + J_{uf} \Delta SU \quad (1.66)$$

Présenté sous cette forme, le problème initial est similaire à l'adaptation d'un modèle linéaire avec simplement des variables d'entrée additionnelles et une matrice de paramètres supplémentaire. Les algorithmes d'identification à employer sont par conséquent identiques à ceux d'un modèle linéaire. Seul, un traitement supplémentaire est requis pour l'obtention des valeurs f_i qui se déduisent alors à partir de l'équation ci-dessous :

$$-J_u F = J_{uf} \quad (1.67)$$

Etant donné la structure diagonale de F , on aboutit à un ensemble d'équations où chaque élément des matrices J_u et J_{uf} intervient séparément.

$$-J_{u,n,m} F_{m,m} = J_{uf,n,m} \text{ pour toutes les lignes } n \text{ et colonnes } m \text{ de } J_u \text{ et } J_{uf}.$$

En tenant compte des relations entre les matrices F et F_0 , nous déduisons les équations suivantes :

$$- J_{u,n,m} f_i = J_{uf,n,m} \text{ avec } i = m \text{ modulo } nbu, \text{ l'indice des valeurs du seuil de chaque actionneur.}$$

Le changement adéquat d'indice permet de séparer les équations suivant chacune des valeurs f_i .

$$- J_{u,n,ij} f_i = J_{uf,n,ij} \text{ pour } n = 1 \text{ à } nby, j = 1 \text{ à } nbru \text{ et } i = 1 \text{ à } nbu.$$

Pour chaque valeur f_i nous avons $nby \times nbru$ équations, dont la solution au sens des moindres carrés vaut :

$$f_i = - \frac{\sum_{n=1}^{nby} \sum_{j=1}^{nbru} J_{uf,n,ij} J_{u,n,ij}}{\sum_{n=1}^{nby} \sum_{j=1}^{nbru} J_{u,n,ij} J_{u,n,ij}} \quad (1.68)$$

L'identification en ligne de la zone inactive permet d'obtenir une estimation de la compensation à appliquer sur chacun des actionneurs pour contrebalancer les effets des zones inactives.

6.3. Compensation

La compensation est définie de la manière suivante. On suppose que la commande sur un actionneur à l'instant $k+1$ peut se mettre sous la forme.

$$U_{k+1} = U_k + \delta U \quad (1.69)$$

Le principe de la compensation est de modifier les valeurs de la commande de sorte qu'elle reste active, c'est à dire en dehors de la zone $[-f, f]$. Par conséquent les règles à appliquer s'expriment de la manière suivante.

<p><i>si</i> $\delta U > 0$ <i>et</i> $-f < U_k + \delta U < f$ <i>alors</i> $U_{k+1}^* = U_k + \delta U + 2f$</p>	<p><i>si</i> $\delta U < 0$ <i>et</i> $-f < U_k + \delta U < f$ <i>alors</i> $U_{k+1}^* = U_k + \delta U - 2f$</p>
---	---

avec U_{k+1}^* la commande compensée à l'instant $k+1$.

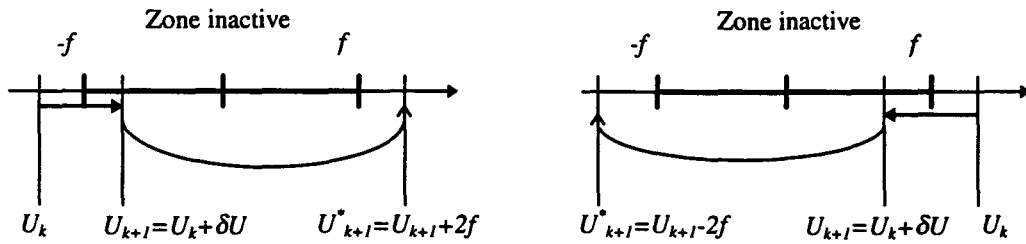


Figure 1.6 : Règles de compensation

Un autre cas peut survenir lors du franchissement total de la zone inactive. Les valeurs de la commande U_k et U_{k+1} ne sont pas comprises dans l'intervalle $[-f, f]$ mais la compensation doit avoir lieu.

si
 $\delta U > 0$
 $U_k < -f$
 $U_k + \delta U > f$
 alors
 $U_{k+1}^* = U_k + \delta U + 2f$

si
 $\delta U < 0$
 $U_k > f$
 $U_k + \delta U < -f$
 alors
 $U_{k+1}^* = U_k + \delta U - 2f$

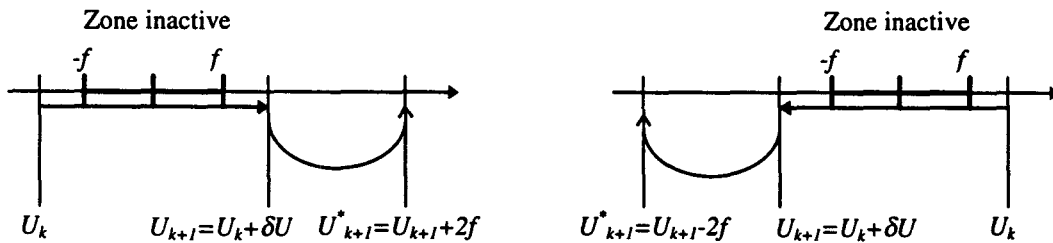


Figure 1.7 : Règles de compensation lors d'un franchissement total

A cela s'ajoute deux cas particuliers qui se produisent lorsque la valeur des seuils est croissante. Des règles supplémentaires permettant de ne pas appliquer deux fois de suite la compensation complète doivent être définies. En effet avec seulement les premières règles, l'ajout consécutif de la correction $2f$ peut survenir et conduit à des valeurs de commande bien trop importantes. Il faut alors effectuer des tests dépendant des valeurs du seuil f aux instants k et $k-1$ pour remédier à ce problème.

si

$$\delta U > 0$$

$$f_{k-1} < U_k < f_k$$

$$f_{k-1} < U_k + \delta U < f_k$$

alors

$$U_{k+1}^* = U_k + \delta U + (f_k - f_{k-1})$$

si

$$\delta U < 0$$

$$-f_k < U_k < -f_{k-1}$$

$$-f_k < U_k + \delta U < -f_{k-1}$$

alors

$$U_{k+1}^* = U_k + \delta U - (f_k - f_{k-1})$$

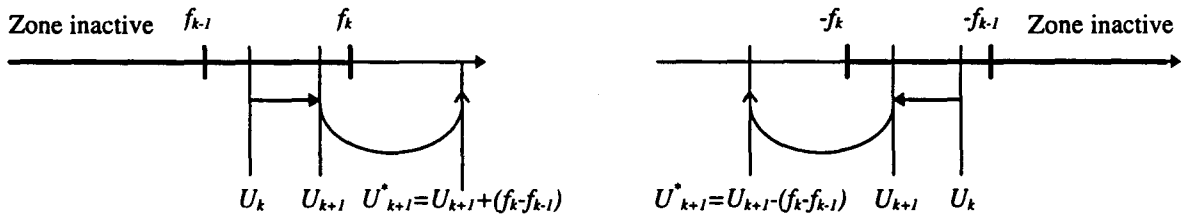


Figure 1.8 : Règles supplémentaires de compensation

La modélisation des zones inactives pour des systèmes mécaniques apporte une nette amélioration de la loi de commande. La compensation s'avère même indispensable dans les cas où la zone inactive a une influence prépondérante.

7. Validation du modèle

7.1. Influence des perturbations

Un processus est communément soumis à des perturbations provenant de l'environnement dans lequel il évolue. Les causes de ces perturbations sont alors généralement inévitables. La robustesse de la loi d'adaptation du modèle et du contrôleur vis à vis des perturbations est donc un aspect important à étudier.

Les perturbations peuvent être considérées comme des entrées supplémentaires inconnues agissant sur le système, et qui de toute évidence faussent les calculs d'adaptation du modèle. Le système perturbé s'écrit :

$$\begin{aligned} \dot{x} &= Ax + Bu + p(t) \\ y &= Cx \end{aligned} \tag{1.70}$$

Des perturbations de grande amplitude à une fréquence élevée rendent alors l'identification inefficace et le modèle complètement inopérant. Il est donc nécessaire de supposer que l'amplitude et la fréquence des perturbations soient telles que le modèle donne une approximation correcte du processus. Le calcul du degré de validité du modèle présenté ci-après permet alors de vérifier que le modèle est satisfaisant malgré les perturbations.

7.2. Détection de dysfonctionnements

L'identification en ligne du processus est une opération assez délicate qui doit être effectuée avec le plus grand soin, et qui est soumise aux aléas de fonctionnement du processus. Les perturbations ou la défaillance d'un capteur, d'un actionneur, ou leur saturation ne doivent pas conduire à une détérioration de l'ensemble du processus. La détection de toute anomalie est donc primordiale pour des raisons élémentaires de sécurité. Il est indispensable de vérifier que le modèle choisi est apte à donner une bonne estimation du comportement du processus. Une mauvaise adéquation entre le modèle et le processus doit être détectée lors de la mise en oeuvre, mais aussi lors de l'utilisation si le processus est soumis à d'importantes perturbations et s'il évolue dans des proportions élevées. Pour le calcul des prédictions et de la commande, il est également utile de définir le domaine de validité du modèle.

La validation du modèle et du système de commande du processus est généralement basée sur l'étude et la comparaison entre le modèle et les mesures. En surveillance, la détermination de relations de redondance à partir du modèle d'un processus permet de détecter et de localiser une éventuelle incohérence de fonctionnement. Des mesures provenant de plusieurs capteurs servent à la génération d'erreurs ou de résidus, qui sont nuls en fonctionnement normal mais pas lors de défaillances. Leur analyse fournit des indications sur le type de pannes et sur l'origine du problème. Mais cette méthode requiert une expertise importante et souvent une augmentation du nombre de capteurs, ce qui est caractéristique des méthodes basées sur la redondance. D'autres méthodes identifient le processus à partir des mesures pour donner une estimation du modèle et de ses paramètres que l'on compare à des valeurs de référence. Un écart conséquent signifie qu'il s'est produit une défaillance. Cette approche n'est pas envisageable, car nous supposons inconnu le modèle exact du processus, l'intérêt de la commande adaptative étant justement d'obtenir le modèle en ligne. Une autre approche [POP 96] consiste à identifier le processus sur deux fenêtres d'observation successives, par exemple sur les intervalles de temps $[t-t_{obs}, t[$ et $[t-2t_{obs}, t-t_{obs}[$. La comparaison entre ces deux modèles permet de détecter une variation brutale du comportement du processus, mais ne détecte pas une dégradation lente du processus. Dans le cas de systèmes non linéaires, variants dans le temps et perturbés, une différence importante entre ces deux modèles est normale et ne correspond pas à une défaillance. Cette méthode n'est donc pas transposable et utilisable dans le cas de systèmes variant dans le temps.

La validation du modèle ou la vérification du bon fonctionnement du processus doit donc s'effectuer uniquement à partir des erreurs de modélisation et des erreurs de suivi de trajectoire. Ces

erreurs peuvent fournir des informations pertinentes à propos d'une part de la qualité du modèle et d'autre part du fonctionnement du processus. Si les erreurs de modélisation sont inférieures à un seuil donné, nous pouvons considérer que l'identification se déroule normalement. Néanmoins, un modèle avec des erreurs faibles peut donner des résultats médiocres. Il faut alors vérifier que le contrôleur annule les erreurs entre la consigne et les sorties du processus. Si ces vérifications sont satisfaisantes, l'ensemble identification et contrôleur, capteurs, actionneurs et processus est en bon état de marche. En cas d'erreurs d'identification et de commande permanentes, il s'agit probablement d'une anomalie dont les causes doivent être déterminées le plus précisément possible.

7.3. Estimation du degré de validité

La validation du modèle s'effectue d'une part en vérifiant que les erreurs d'identification sont faibles, et d'autre part en mesurant les coefficients de corrélation, qui indiquent le degré de correspondance entre le modèle et les mesures.

La somme S des erreurs au carré entre les estimations données par le modèle et les mesures doit être la plus petite possible, et le coefficient de corrélation r doit être le plus proche de 1. Le calcul de ces deux tests s'effectue de la manière présentée ci-dessous à partir de m mesures y et des valeurs \hat{y} estimées avec le modèle :

$$S = \sum_{i=1}^m (y_i - \hat{y}_i)^2 \quad \text{et} \quad r^2 = 1 - \frac{m \sum_{i=1}^m (y_i - \hat{y}_i)^2}{m \sum_{i=1}^m y_i^2 - \left(\sum_{i=1}^m y_i \right)^2} \quad (1.71)$$

Le premier test donne des résultats dépendant de l'ordre de grandeur des mesures et ne reflète donc pas obligatoirement la qualité du modèle. En revanche le second test est plus pertinent et indique vraiment si le modèle est valide ou non. Des simulations sur plusieurs processus montrent que le seuil à partir duquel le modèle est acceptable correspond à un coefficient de corrélation r^2 supérieur à 0,95.

Le calcul de ce coefficient pour un modèle dans le repère local (1.45) peut se formuler plus simplement de manière à réduire le nombre d'opérations et à diminuer le temps de calcul. En définissant et nommant un ensemble de m points de mesures par les matrices ΔY_{ens} , ΔX_{ens} et ΔU_{ens} , le vecteur des erreurs de modélisation par rapport à la variable de sortie i , notées $E_{ens,i}$ (erreurs avec le $i^{\text{ème}}$ hyperplan correspondant à la $i^{\text{ème}}$ ligne des matrices J_x et J_u), il vient :

$$E_{ens,i} = \Delta Y_{ens,i} - J_{x,i} \Delta X_{ens} - J_{u,i} \Delta U_{ens} \quad (1.72)$$

Le coefficient de corrélation pour cet hyperplan vaut alors :

$$r_i^2 = 1 - \frac{m E_{ens,i} E_{ens,i}^T}{m \Delta Y_{ens,i} \Delta Y_{ens,i}^T - \left(\sum_{j=1}^m \Delta Y_{ens,i,j} \right)^2} \quad (1.73)$$

avec $\Delta Y_{ens,i,j}$, la $i^{\text{ème}}$ valeur du $j^{\text{ème}}$ point de l'ensemble ΔY_{ens} .

De plus par définition du point de fonctionnement pour l'identification (Chapitre II) et de Y_{ope} , le terme $\sum_{j=1}^m \Delta Y_{ens,i,j}$ est nul. Le développement et la simplification des calculs conduisent à

l'expression suivante :

$$r_i^2 = \frac{\begin{bmatrix} J_{x,i} & J_{u,i} \end{bmatrix} \left(2 \begin{bmatrix} \Delta X_{ens} \\ \Delta U_{ens} \end{bmatrix} \Delta Y_{ens,i}^T - \begin{bmatrix} \Delta X_{ens} \\ \Delta U_{ens} \end{bmatrix} \begin{bmatrix} \Delta X_{ens}^T & \Delta U_{ens}^T \end{bmatrix} \begin{bmatrix} J_{x,i}^T \\ J_{u,i}^T \end{bmatrix} \right)}{\Delta Y_{ens,i} \Delta Y_{ens,i}^T} \quad (1.74)$$

Dans le cas du modèle avec estimation de la zone inactive, ce test s'avère encore plus important car il faut vérifier que les hypothèses simplificatrices émises pour la détermination des seuils sont correctes. La première hypothèse concerne la simplification du modèle qui par construction est incorrect dans la zone inactive. En effet, on suppose que les actionneurs ne sont utilisés grâce à la compensation que dans la zone active. Une seconde approximation a été réalisée pour déterminer les seuils f à partir de la relation (1.67) en appliquant un critère du type moindres carrés. La validation du modèle s'effectue alors en calculant le coefficient de corrélation entre les mesures et l'équation suivante :

$$\Delta Y = J_x \Delta X + J_u \Delta U + J_{uf} \Delta SU \quad (1.75)$$

avec

$$J_{uf} = -J_u F \quad (1.76)$$

d'où

$$r_i^2 = \frac{\begin{bmatrix} J_{x,i} & J_{u,i} & -J_u F \end{bmatrix} \left(2 \begin{bmatrix} \Delta X_{ens} \\ \Delta U_{ens} \\ \Delta SU_{ens} \end{bmatrix} \Delta Y_{ens,i}^T - \begin{bmatrix} \Delta X_{ens} \\ \Delta U_{ens} \\ \Delta SU_{ens} \end{bmatrix} \begin{bmatrix} \Delta X_{ens}^T & \Delta U_{ens}^T & \Delta SU_{ens}^T \end{bmatrix} \begin{bmatrix} J_{x,i}^T \\ J_{u,i}^T \\ -(J_u F)^T \end{bmatrix} \right)}{\Delta Y_{ens,i} \Delta Y_{ens,i}^T} \quad (1.77)$$

Les produits de matrices entre les divers ensembles de mesures s'effectuant de manière récursive lors de l'adaptation du modèle (chapitre II), l'estimation de r en ligne se déduit facilement et ne pénalise pas le temps de calcul.

La détection de divergence et la vérification des ordres de grandeur de certains paramètres peut également donner des informations sur le fonctionnement de l'algorithme et du processus. Par exemple, la dynamique du processus est caractérisée par la matrice J_x . Pour une variable de sortie i , les coefficients de J_x portant sur les régressions de cette même sortie ne doivent pas dépasser certaines limites. Si la somme de ces coefficients est supérieure à 1, la sortie en appliquant aucune commande a tendance à diverger. Si cette même somme dépasse 10, le modèle obtenu est très divergent et les variations entre chaque pas de calcul sont telles que la commande peut difficilement contrôler le processus.

L'ensemble de ces tests n'assurent pas complètement le rôle d'un système de surveillance puisque le diagnostic porte uniquement sur quelques points. Mais ces algorithmes donnent des renseignements sur le bon fonctionnement du processus et du contrôleur, et sur la validité du modèle. Ces tests permettent donc de vérifier que les hypothèses sur lesquelles est fondée la loi de commande sont satisfaites. Le cas échéant l'ordre du modèle, le modèle lui-même, ou le processus sont à modifier et à concevoir différemment.

8. Conclusion

Dans ce chapitre nous nous sommes intéressés au problème du choix de la structure du modèle à utiliser en vue d'établir une loi de commande adaptative et prédictive. Après une remise en cause de la modélisation par réseaux de neurones qui conduit à des modèles trop complexes, nous avons opté pour un modèle discret linéaire local, qui offre les avantages de la simplicité des systèmes linéaires tout en étant capable de donner une bonne approximation locale du comportement dynamique de processus multivariables complexes.

Des extensions à ce type de modèles ont été également proposées pour remédier à l'inaptitude des modèles linéaires à modéliser les processus comportant des discontinuités telles que des zones inactives sur la commande.

Chapitre II :

Adaptation des modèles

1. Introduction

Le choix du modèle et de sa structure est une étape importante puisqu'une bonne adéquation entre le modèle et le processus réel est la première condition à respecter pour obtenir de bons résultats. L'adaptation des paramètres ou coefficients du modèle constitue la seconde étape de l'identification d'un processus. Cette étape se révèle tout aussi importante que la première. Le choix d'une loi d'adaptation appropriée est absolument nécessaire pour aboutir à des performances satisfaisantes. L'étude de diverses méthodes d'adaptation est présentée dans ce chapitre. La rapidité d'adaptation, la robustesse numérique, la stabilité, les conditions d'application ainsi que les limites et inconvénients des méthodes proposées sont les aspects principaux abordés.

2. Adaptation par rétro-propagation du gradient

Le développement de cette méthode d'apprentissage constitue une avancée très importante dans la théorie des réseaux de neurones [PAT 95]. Cependant sa stabilité insuffisante et le grand nombre d'itérations nécessaires pour obtenir une solution satisfaisante limitent son utilisation à la résolution de problèmes hors ligne, en temps différé.

Le principe est de modifier les paramètres du modèle de manière à ce que les erreurs de modélisation décroissent. Dans le cas du modèle s'écrivant sous la forme (1.10), l'erreur de modélisation e_i par rapport à la $i^{\text{ème}}$ variable de sortie s'écrit à chaque instant :

$$e_i = Y_i - J_{x,i} X - J_{u,i} U \quad (2.1)$$

Le critère pour minimiser l'erreur e_i s'écrit :

$$H = \frac{1}{2} e_i^2 \quad (2.2)$$

Les dérivées ou le gradient de H par rapport au $j^{\text{ème}}$ coefficient des matrices $J_{x,i}$ et $J_{u,i}$ valent respectivement :

$$\frac{\partial H}{\partial J_{x,ij}} = -X_j e_i \quad \text{et} \quad \frac{\partial H}{\partial J_{u,ij}} = -U_j e_i. \quad (2.3)$$

La réduction de l'erreur e_i s'obtient alors en adaptant chaque paramètre du modèle dans le sens opposé au gradient. La loi d'adaptation pour l'ensemble des paramètres et des variables de sorties s'écrit alors :

$$\begin{cases} J_{x_{k+1}} = J_{x_k} + \eta e X^T \\ J_{u_{k+1}} = J_{u_k} + \eta e U^T \end{cases} \quad (2.4)$$

avec η le gain d'adaptation ou d'apprentissage ($\eta > 0$).

Ce gain η doit être choisi suffisamment petit pour éviter des grandes variations des paramètres qui peuvent rendre instable la loi d'adaptation. Inversement une valeur trop faible du gain entraîne une adaptation très lente du modèle, qui ne représente plus correctement le processus réel lors d'un changement important des caractéristiques de fonctionnement. Le temps de convergence vers les paramètres optimaux est alors beaucoup trop long par rapport aux variations des paramètres du processus. Le réglage de ce gain est donc problématique.

Le principal avantage de l'algorithme par rétro-propagation du gradient est qu'il peut s'appliquer et être généralisé pour adapter de nombreux types de modèles linéaires ou non [OHB 96]. De plus aucune inversion de matrice est effectuée, ce qui limite les problèmes numériques. Cependant le risque de divergence où de convergence vers un minimum local est possible. La stabilité de l'identification et de la loi de commande qui en découle n'est alors pas garantie. On préfère pour ces raisons se tourner vers d'autres algorithmes d'adaptation.

3. Loi d'adaptation déduite d'une fonction de Lyapunov

La théorie de la stabilité de Lyapunov est à l'origine de nombreuses lois d'adaptation pour des réseaux de neurones dynamiques [BOL 92, KOS 95, ROV 94], dont le principal atout est de garantir la stabilité et la convergence des poids du réseau et du contrôleur. Cependant ces approches ne donnent pas entièrement satisfaction à cause des conditions d'utilisation restrictives, de la relative lenteur d'adaptation du réseau et des performances qui ne répondent pas exactement à nos attentes.

Le principe est de déterminer une fonction de Lyapunov définie positive incluant tous les termes ou erreurs que l'on souhaite annuler ou minimiser. La loi d'adaptation et la commande sont formulées de telle sorte que la fonction candidate de Lyapunov ait une dérivée négative et soit par conséquent décroissante. La loi d'adaptation illustrant le principe est établie pour un système linéaire continu supposé commandable et observable, avec le vecteur de sortie égal au vecteur d'état. Le modèle « exact » du processus est donc supposé être de la forme :

$$\begin{aligned} \dot{x}_p &= A_p x_p + B_p u_p \\ y_p &= x_p \end{aligned} \quad (2.5)$$

Chapitre II : Adaptation des modèles

avec A_p et B_p des matrices inconnues traduisant le comportement dynamique du processus, x_p et y_p l'état et la sortie du processus et u_p la commande appliquée.

Le modèle à adapter est défini par une équation différentielle du type :

$$\begin{aligned} \dot{x}_m &= Ax_m + A_m x_p + B_m u_p \\ y_m &= x_m \end{aligned} \quad (2.6)$$

avec A_m et B_m les matrices du modèle à adapter, x_m et y_m l'état et la sortie du modèle, et A une matrice définie négative.

Le problème de l'adaptation du modèle est d'obtenir les coefficients de A_m et de B_m tels que les équations (2.5) et (2.6) soient équivalentes et que les états x_p et x_m soient identiques. Les conditions à satisfaire sont :

$$\begin{aligned} x_m &= x_p \\ A_m &= A_p - A \\ B_m &= B_p \end{aligned} \quad (2.7)$$

Il faut alors réduire l'erreur e entre l'état du processus et l'état du modèle ainsi que les erreurs de modélisation e_A , e_B entre les matrices A_p , A_m et B_p , B_m . Les erreurs à minimiser sont donc :

$$\begin{aligned} e &= x_p - x_m \\ e_A &= A_p - A_m - A \\ e_B &= B_p - B_m \end{aligned} \quad (2.8)$$

La fonction candidate de Lyapunov L permettant de minimiser ces erreurs est définie ainsi :

$$L = \frac{1}{2} (e^T P e + tr(e_A^T R_A e_A) + tr(e_B^T R_B e_B)) \quad (2.9)$$

avec R_A , R_B et P des matrices diagonales définies positives avec leurs termes diagonaux identiques valant respectivement r_A , r_B et p . La fonction tr correspond à la trace d'une matrice.

La dérivée de cette fonction L par rapport au temps est égale à :

$$\dot{L} = \dot{e}^T P e + tr(\dot{e}_A^T R_A e_A) + tr(\dot{e}_B^T R_B e_B) \quad (2.10)$$

avec d'après les équations (2.6) et (2.8) :

$$\dot{e} = \dot{x}_p - \dot{x}_m \text{ d'où } \dot{e} = Ae + e_A x_p + e_B u_p \quad (2.11)$$

La substitution de l'équation (2.11) dans (2.10) fournit alors le résultat suivant :

$$\dot{L} = \frac{1}{2} e^T (A^T P + PA) e + e^T P e_A x_p + e^T P e_B u_p + tr(\dot{e}_A^T R_A e_A) + tr(\dot{e}_B^T R_B e_B) \quad (2.12)$$

L'expression de cette dérivée devient :

$$\dot{L} = \frac{1}{2} e^T (A^T P + PA) e \quad (2.13)$$

en posant :

$$\begin{aligned} \text{tr}(\dot{e}_A^T R_A e_A) &= -e^T P e_A x_p \\ \text{tr}(\dot{e}_B^T R_B e_B) &= -e^T P e_B x_p \end{aligned} \quad (2.14)$$

Par définition de A et de P , le terme $A^T P + PA$ est une matrice définie strictement négative, ce qui garantit la décroissance de la fonction de Lyapunov. Les équations (2.14) sont vérifiées en définissant pour chacun des termes i, j des matrices e_A et e_B les lois d'évolution suivantes :

$$\begin{aligned} \dot{e}_{A,i,j} &= -pr_A e_i x_{p,j} \\ \dot{e}_{B,i,j} &= -pr_B e_i u_{p,j} \end{aligned} \quad (2.15)$$

Pour un système linéaire stationnaire, les simplifications ci-dessous peuvent être déduites :

$$\begin{aligned} \dot{e}_A &= \dot{A}_p - \dot{A}_m - \dot{A} = -\dot{A}_m \\ \dot{e}_B &= \dot{B}_p - \dot{B}_m = -\dot{B}_m \end{aligned} \quad (2.16)$$

Avec ces hypothèses, les lois d'adaptation des matrices A_m et B_m s'écrivent alors :

$$\begin{aligned} \dot{A}_{m,i,j} &= pr_A e_i x_{p,j} \\ \dot{B}_{m,i,j} &= pr_B e_i u_{p,j} \end{aligned} \quad (2.17)$$

ou encore

$$\begin{aligned} \dot{A}_m &= PR_A e x_p^T \\ \dot{B}_m &= PR_B e u_p^T \end{aligned} \quad (2.18)$$

La loi d'adaptation obtenue pour un système continu avec une fonction de Lyapunov est similaire à celle déduite par la méthode de la rétro-propagation du gradient pour le modèle discret. Le gain d'adaptation η est remplacé par les coefficients de pondération P, R_A, R_B et la matrice A qui influent sur la rapidité de l'adaptation et sur la dynamique de correction des erreurs de modélisation. Leur réglage lors de la mise en oeuvre est alors assez délicat. Cependant, la formulation avec la fonction de Lyapunov permet de garantir la décroissance et l'annulation des erreurs de modélisation sous réserve que le processus soit linéaire, stationnaire, non perturbé, et qu'il n'y ait pas de dynamique négligée. Dans le cas contraire, il est démontré que la fonction de Lyapunov est décroissante [ROV 94] jusqu'à un certain seuil qui dépend du modèle utilisé, des non-linéarités du système et des perturbations. Ce seuil est difficile à quantifier et peut avoir une valeur élevée, ce qui signifie que la loi de commande déduite du modèle peut avoir une précision médiocre. Des erreurs importantes entre la consigne et la sortie du système peuvent donc apparaître.

4. Algorithme basé sur la méthode des moindres carrés

Une analyse simple des deux méthodes précédentes laisse entrevoir l'origine de la faiblesse de ces méthodes. En effet, l'adaptation des coefficients est basée seulement sur une mesure et une erreur de modélisation à un instant donné. Ces lois d'adaptation disposent donc de peu d'informations pour modifier un nombre conséquent de poids ou de coefficients. Le ratio entre les données et informations utilisées et le travail d'adaptation à effectuer est alors peu favorable pour aboutir rapidement à un modèle précis.

Avec un algorithme d'adaptation du type moindres carrés reposant sur un nombre relativement important de mesures, et pour un modèle avec un nombre réduit de paramètres à adapter, l'obtention et la mise à jour complète du modèle peut s'effectuer rapidement en quelques dizaines de pas d'échantillonnage. Cette propriété est vivement recherchée car la rapidité d'adaptation est déterminante pour identifier et commander des systèmes variant dans le temps ou non linéaires. Le choix des mesures à utiliser est alors le premier point à développer.

4.1. Choix de la fenêtre d'observation

De nombreuses variantes et formulations des algorithmes des moindres carrés ont déjà été développées [BIT 90, BOR 92, LAN 93, RIC 91, SUD 96]. La première distinction entre les méthodes est due au choix et au nombre de données utilisées pour réaliser l'adaptation. La définition du critère est également différente selon les méthodes. La manière de poser les équations et de résoudre le système jouent aussi un rôle important et influent sur la robustesse de l'algorithme. Une première approche consiste à utiliser toutes les mesures disponibles. L'adaptation se réalise en prenant en compte toutes les données passées. Seulement lorsque les caractéristiques du processus évoluent, les mesures les plus anciennes ne sont plus valables car elles ne correspondent plus au processus réel. Pour remédier à ce problème, un facteur d'oubli peut être affecté aux mesures à chaque pas de calcul afin d'éliminer l'influence des mesures les plus anciennes, et d'accorder une plus grande importance aux nouvelles. Le réglage du facteur d'oubli (λ) est parfois difficile, car pour un système fortement non linéaire dont les variations paramétriques sont importantes, ce facteur doit être petit ($0 < \lambda \ll 1$) pour ne tenir compte que des dernières mesures, alors que pour un système stationnaire en régime permanent, le facteur doit être proche de 1 afin de conserver suffisamment de données pour effectuer une estimation précise du modèle.

Pour mieux contrôler quelles sont les mesures utilisées, une fenêtre d'observation mobile avec un nombre fixe de points est une solution plus efficace. Après un changement des paramètres du processus, le renouvellement complet des mesures a lieu au bout d'un temps égal au produit entre la période d'acquisition des données et le nombre de points utilisés. Le remplacement fréquent et

complet des données permet donc d'obtenir un algorithme avec une bonne rapidité d'adaptation du modèle. Cependant en régime permanent, toutes les mesures, ou une ou plusieurs composantes de toutes les mesures sont constantes. Or des données identiques ou trop voisines n'apportent aucune information complémentaire sur le comportement du processus. Nous sommes alors confrontés au problème de la condition d'excitation permanente. Ce problème peut être en partie résolu en ajoutant une mesure que si elle est suffisamment différente de la dernière mesure y_{k-i} mémorisée dans la fenêtre d'observation. Le remplacement d'un point ne s'effectue plus systématiquement mais seulement quand l'évolution de la sortie y_k par rapport à y_{k-i} est supérieure à un seuil σ . La condition à vérifier est donc :

$$\|y_k - y_{k-i}\|_2 > \sigma \quad (2.19)$$

avec

σ le seuil à dépasser pour insérer la mesure y_k

y_{k-i} la dernière mesure mémorisée dans la fenêtre d'observation.

Les mesures utilisées et incluses dans la fenêtre d'observation sont alors réparties de manière non uniforme (Figure 2.1).

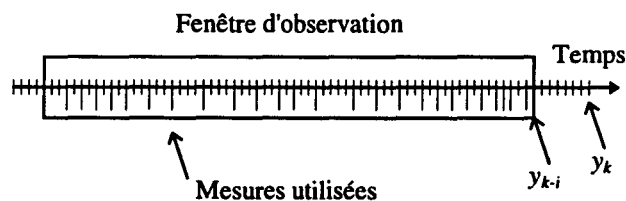


Figure 2.1 : Répartition des mesures utilisées dans la fenêtre d'observation

Pour que la mise à jour des données soit rapide, et afin que le modèle obtenu soit une approximation locale du processus autour du point de fonctionnement, les mesures utilisées pour l'identification doivent être peu nombreuses, récentes et regroupées autour du point de fonctionnement. En revanche pour que l'estimation des paramètres soit réalisable ou suffisamment précise, ces mesures doivent être réparties le plus possible dans tout l'espace des mesures. Ce compromis est résolu avec la fenêtre d'observation ainsi définie qui assure une bonne répartition spatiale et temporelle des mesures.

4.2. Problème d'excitation permanente

La condition d'excitation permanente ne peut pas être satisfaite pendant le fonctionnement normal d'un processus, puisque la consigne que doit suivre le système est souvent une constante, un échelon, une rampe ou des courbes relativement régulières. Or ces signaux ne sont pas riches en

fréquences, ou seulement sur une courte période dans le cas d'un échelon. En régime permanent et stationnaire (position ou vitesse constante), les mesures ne sont donc pas suffisamment riches pour obtenir avec précision une estimation des paramètres. Ce problème est accentué pour des modèles d'ordre élevé, où de nombreuses régressions de la même mesure sont utilisées. Pour des systèmes multivariables, une sortie peut être imposée à une valeur constante alors que les autres sorties doivent suivre des trajectoires non constantes. Le choix de consignes identiques pour deux sorties différentes peut être aussi une cause de richesse insuffisante des signaux.

Inéluctablement, les conséquences sont l'apparition de matrices singulières ou mal conditionnées qui posent de sérieuses difficultés numériques. Or ces matrices sont utilisées pour l'estimation des paramètres du modèle, dont l'adaptation devient alors impossible ou est très sensible aux erreurs d'arrondi et aux bruits. Ce problème ne peut pas être négligé, car cette situation est fréquente. Les quelques exemples cités précédemment montrent à quel point le risque d'aboutir à des matrices et des systèmes singuliers est élevé. Les causes sont en effet multiples et inévitables. La robustesse des algorithmes d'identification est donc très faible si aucune modification des critères vient corriger le problème.

Les premières précautions sont à prendre lors de l'acquisition des mesures pour éviter d'ajouter plusieurs fois des points identiques ou similaires dans les matrices mémorisant les données utilisées pour l'identification. En effet des points proches n'apportent aucune information utile. Ces précautions sont toutefois insuffisantes car elles ne permettent pas de résoudre toutes les causes d'apparition des matrices singulières. Le critère à minimiser et l'algorithme de résolution doivent alors être développés afin de fonctionner correctement même dans des conditions critiques.

Le critère pour l'adaptation des paramètres du modèle comporte alors un premier terme égal à la somme des erreurs au carré entre les mesures et les estimations fournies par le modèle. Des termes supplémentaires sont inclus dans le critère pour créer des contraintes qui résolvent les problèmes de matrices singulières. Les contraintes les plus simples consistent à limiter l'amplitude des variations des paramètres. Ces termes ne doivent cependant pas modifier ou fausser la solution du système à résoudre. Les contraintes doivent donc rester faibles afin de ne pas détériorer les résultats et la dynamique d'adaptation. Plusieurs approches basées sur ce principe sont présentées dans les sections suivantes.

4.3. Critère avec ajout de points déduits du modèle

Les mesures de la fenêtre d'observation ne permettant pas d'obtenir une matrice régulière de manière infaillible, la méthode proposée consiste à définir et à générer des points « artificiels » à partir du modèle pour aboutir à des matrices régulières définies positives.

Le nombre initial de points de mesures mémorisés dans la fenêtre d'observation est égal à m . Les matrices comprenant l'ensemble de ces mesures (Y , X , et U) sont notées avec l'indice ens ce qui donne les matrices Y_{ens} , X_{ens} et U_{ens} . Le point de fonctionnement pour l'identification est notée P_{ide} et est formé avec les vecteurs Y_{ide} , X_{ide} et U_{ide} qui se calculent respectivement en effectuant la moyenne des m mesures des ensembles Y_{ens} , X_{ens} et U_{ens} .

4.3.1. Modification de l'écriture du modèle

Le modèle linéaire du type (1.48) correspond pour chaque variable de sortie y_j à une fonction linéaire de \mathbb{R}^n dans \mathbb{R} . Le modèle peut donc être représenté par un hyperplan défini à l'aide de $n+1$ points notés P_0, P_1, \dots, P_n . Pour la sortie y_j les coordonnées des points P_i sont définies de la manière suivante :

$$P_0 = \begin{bmatrix} P_{0,y_j} \\ P_{0,x} \\ P_{0,U} \end{bmatrix} \text{ et } P_i = \begin{bmatrix} P_{i,y_j} \\ P_{i,x} \\ P_{i,U} \end{bmatrix} \text{ avec } \begin{bmatrix} P_{i,x} \\ P_{i,U} \end{bmatrix} = \begin{bmatrix} P_{0,x} \\ P_{0,U} \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ L_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ pour } i = 1 \dots n. \quad (2.20)$$

Le point P_0 est le point d'origine de l'hyperplan et les points P_i ($i=1 \dots n$) sont disposés par rapport à P_0 suivant la composante i à une distance L_i choisie en fonction de la précision désirée et de l'ordre de grandeur des mesures. En se basant sur la définition des $n+1$ points, le modèle pour la variable y_j s'écrit :

$$y_j = P_{0,y_j} + \begin{bmatrix} \frac{P_{1,y_j} - P_{0,y_j}}{L_1} & \frac{P_{2,y_j} - P_{0,y_j}}{L_1} & \dots & \frac{P_{n,y_j} - P_{0,y_j}}{L_n} \end{bmatrix} \begin{bmatrix} X - P_{0,x} \\ U - P_{0,U} \end{bmatrix} \quad (2.21)$$

ce qui est équivalent au modèle :

$$y_j = \begin{bmatrix} J_{x,j} & J_{u,j} \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} + V_{ope,j} \quad (2.22)$$

avec

$$\begin{bmatrix} J_{x,j} & J_{u,j} \end{bmatrix} = \begin{bmatrix} \frac{P_{1,y_j} - P_{0,y_j}}{L_1} & \frac{P_{2,y_j} - P_{0,y_j}}{L_2} & \dots & \frac{P_{n,y_j} - P_{0,y_j}}{L_n} \end{bmatrix} \quad (2.23)$$

$$V_{ope,j} = P_{0,y_j} - \begin{bmatrix} J_{x,j} & J_{u,j} \end{bmatrix} \begin{bmatrix} P_{0,x} \\ P_{0,U} \end{bmatrix} \quad (2.24)$$

Pour simplifier l'écriture lors du calcul de minimisation du critère, on pose :

$$L = \begin{bmatrix} L_x & L_u \end{bmatrix} = \begin{bmatrix} \frac{1}{L_1} & & & \\ & \frac{1}{L_2} & & \\ & & \ddots & \\ & & & \frac{1}{L_n} \end{bmatrix} \quad (2.25)$$

Les matrices L_x et L_u sont les inverses des distances L_i utilisées respectivement pour déduire $J_{x,j}$ et $J_{u,j}$, d'où :

$$\begin{bmatrix} J_{x,j} & J_{u,j} \end{bmatrix} = \begin{bmatrix} P_{1,y_j} - P_{0,y_j} & P_{2,y_j} - P_{0,y_j} & \dots & P_{n,y_j} - P_{0,y_j} \end{bmatrix} \begin{bmatrix} L_x \\ L_u \end{bmatrix} \quad (2.26)$$

La représentation graphique de ce modèle est présentée à la figure 2.2. Les points P_i sont les points « artificiels » définissant le modèle.

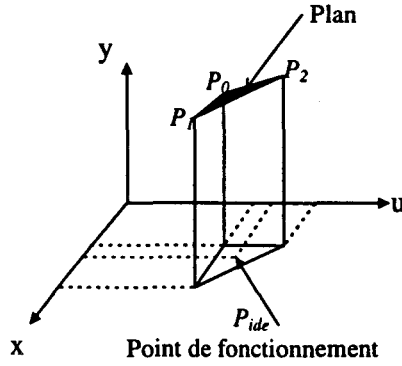


Figure 2.2 : Représentation graphique du modèle.

4.3.2. Positionnement des points du modèle

Les coordonnées $P_{0,x}$ et $P_{0,u}$ du point d'origine de ce modèle sont calculées de sorte que les points P_i soient répartis autour du point d'identification P_{ide} . Pour aboutir à ce résultat, on impose au centre de gravité

des $n+1$ points $\begin{bmatrix} P_{i,x} \\ P_{i,u} \end{bmatrix}$ d'être confondu avec le point d'identification $\begin{bmatrix} X_{ide} \\ U_{ide} \end{bmatrix}$ ce qui

donne :

$$\begin{bmatrix} P_{G,x} \\ P_{G,u} \end{bmatrix} = \begin{bmatrix} P_{0,x} \\ P_{0,u} \end{bmatrix} + \frac{1}{n+1} \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{bmatrix} \text{ d'où } \begin{bmatrix} P_{0,x} \\ P_{0,u} \end{bmatrix} = \begin{bmatrix} X_{ide} \\ U_{ide} \end{bmatrix} - \frac{1}{n+1} \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{bmatrix} \quad (2.27)$$

A chaque mise à jour du modèle, les coordonnées X et U des points P_i subissent un déplacement qui est égal aux variations du point d'identification notées $\begin{bmatrix} \Delta X_{ide} \\ \Delta U_{ide} \end{bmatrix}$. Ces variations entraînent alors une

modification des coordonnées P_{i,y_j} des points P_i calculée à partir du modèle de la manière suivante :

$$\Delta P_{i,y_j} = \begin{bmatrix} J_{x,j} & J_{u,j} \end{bmatrix} \begin{bmatrix} \Delta X_{ide} \\ \Delta U_{ide} \end{bmatrix} \quad (2.28)$$

Cette première adaptation des coordonnées à appliquer aux points P_i est due au déplacement du point d'identification et se résume de la manière suivante :

$$\Delta P_i = \begin{bmatrix} \begin{bmatrix} J_{x,j} & J_{u,j} \end{bmatrix} \begin{bmatrix} \Delta X_{ide} \\ \Delta U_{ide} \end{bmatrix} \\ \Delta X_{ide} \\ \Delta U_{ide} \end{bmatrix} \text{ pour } i=0\dots n. \quad (2.29)$$

4.3.3. Définition du critère

L'identification du processus et l'adaptation des paramètres du modèle pour la sortie y_j sont réalisées en corrigeant les coordonnées P_{i,y_j} des $n+1$ points P_i du modèle. Ces variations sont notées Δ_{y_j} qui est un vecteur de dimension $n+1$ que l'on sépare en trois parties, l'une pour le point P_0 , la seconde pour les points P_i permettant de déduire $J_{x,j}$ et la dernière pour les points P_i permettant de calculer $J_{u,j}$.

$$\Delta_{y_j} = [\Delta_0 \quad \Delta_1 \quad \Delta_2] \quad (2.30)$$

Les erreurs de prédiction entre le modèle (2.22) de la sortie y_j ($j^{\text{ème}}$ hyperplan) et les ensembles de mesures Y_{ens} , X_{ens} et U_{ens} s'écrivent :

$$E_{ens,j} = Y_{ens,j} - \begin{bmatrix} J_{x,j} & J_{u,j} \end{bmatrix} \begin{bmatrix} X_{ens} \\ U_{ens} \end{bmatrix} - V_{ope,j} 1_m^T \quad (2.31)$$

avec 1_n un vecteur de dimension n comportant que des 1.

La correction des erreurs générée en modifiant le modèle vaut :

$$\Delta E_{ens,j} = -[\Delta_1 \quad \Delta_2] L \begin{bmatrix} X_{ens} - P_{0,x} 1_m^T \\ U_{ens} - P_{0,u} 1_m^T \end{bmatrix} + [\Delta_0 \quad \dots \quad \Delta_0] L \begin{bmatrix} X_{ens} - P_{0,x} 1_m^T \\ U_{ens} - P_{0,u} 1_m^T \end{bmatrix} - \Delta_0 1_m^T \quad (2.32)$$

En regroupant chacun des termes du vecteur Δ_{y_j} et en nommant M_0 , M_1 et M_2 les matrices multipliant respectivement les termes Δ_0 , Δ_1 et Δ_2 , ces variations d'erreurs s'écrivent :

$$\Delta E_{ens,j} = \Delta_0 \left(1_n^T L \begin{bmatrix} X_{ens} - P_{0,x} 1_m^T \\ U_{ens} - P_{0,u} 1_m^T \end{bmatrix} - 1_m^T \right) - \Delta_1 L_x (X_{ens} - P_{0,x} 1_m^T) - \Delta_2 L_u (U_{ens} - P_{0,u} 1_m^T) \quad (2.33)$$

soit

$$\Delta E_{ens,j} = \Delta_0 M_0 + \Delta_1 M_1 + \Delta_2 M_2 \quad (2.34)$$

Le critère est alors défini de manière à ce que le vecteur Δ_{y_j} minimise les erreurs $E_{ens,j} + \Delta E_{ens,j}$.

Les points P_i du modèle sont aussi pris en compte dans le critère car on souhaite réduire les variations Δ_{y_j} . Le critère s'écrit donc :

$$H = \frac{1}{2} \left(\Delta_{y_j} \Delta_{y_j}^T + (E_{ens,j} + \Delta E_{ens,j}) P (E_{ens,j} + \Delta E_{ens,j})^T \right) \quad (2.35)$$

avec P une matrice de pondération des erreurs qui peut être une matrice identité pour simplifier la mise en oeuvre. Cependant une légère amélioration de l'identification est obtenue en accordant un poids plus élevé aux données récentes de la fenêtre d'observation. Le modèle s'adapte alors plus rapidement lors d'un changement des caractéristiques du processus.

4.3.4. Minimisation du critère

La minimisation d'un critère quadratique s'obtient en annulant la dérivée du critère par rapport aux variables. Les conditions à respecter sont donc :

$$\frac{\partial H}{\partial \Delta_0} = 0, \frac{\partial H}{\partial \Delta_1} = 0, \frac{\partial H}{\partial \Delta_2} = 0 \text{ et } \frac{\partial^2 H}{\partial \Delta_0^2} > 0, \frac{\partial^2 H}{\partial \Delta_1^2} > 0, \frac{\partial^2 H}{\partial \Delta_2^2} > 0 \quad (2.36)$$

Ces calculs conduisent aux équations suivantes :

$$\begin{cases} \frac{\partial H}{\partial \Delta_0} = \Delta_0^T + M_0 P (E_{ens,j} + \Delta_0 M_0 + \Delta_1 M_1 + \Delta_2 M_2)^T = 0 \\ \frac{\partial H}{\partial \Delta_1} = \Delta_1^T + M_1 P (E_{ens,j} + \Delta_0 M_0 + \Delta_1 M_1 + \Delta_2 M_2)^T = 0 \\ \frac{\partial H}{\partial \Delta_2} = \Delta_2^T + M_2 P (E_{ens,j} + \Delta_0 M_0 + \Delta_1 M_1 + \Delta_2 M_2)^T = 0 \end{cases} \quad (2.37)$$

Pour calculer les nouvelles ordonnées P_{i,y_j} des points du modèle, le système suivant est à résoudre :

$$\begin{bmatrix} I + M_0 P M_0^T & M_0 P M_1^T & M_0 P M_2^T \\ M_1 P M_0^T & I + M_1 P M_1^T & M_1 P M_2^T \\ M_2 P M_0^T & M_2 P M_1^T & I + M_2 P M_2^T \end{bmatrix} \begin{bmatrix} \Delta_0^T \\ \Delta_1^T \\ \Delta_2^T \end{bmatrix} = - \begin{bmatrix} M_0 \\ M_1 \\ M_2 \end{bmatrix} P E_{ens,j}^T \quad (2.38)$$

Ce calcul s'applique également de manière similaire pour les autres sorties y_j du modèle. Les matrices M_0 , M_1 et M_2 restent alors identiques. Seul le vecteur des erreurs $E_{ens,j}$ est à calculer pour chaque valeur de j . La formulation plus générale pour des systèmes linéaires résultant de la

minimisation d'un critère quadratique est du type $Ax = b$ avec A une matrice symétrique définie positive, b le vecteur désiré et x la solution. L'algorithme de Cholesky pour la résolution de ce type de systèmes est rappelée au chapitre II.4.7

La mise à jour du modèle se déroule donc en calculant au préalable le point de fonctionnement et les coordonnées des points P_i . Le calcul des variations du modèle se déduit dans un second temps en minimisant le critère (2.35) par rapport aux variations Δy_j pour chaque sortie y_j du modèle.

4.4. Critère pour un modèle exprimé autour du point de fonctionnement

L'adaptation du modèle formulé dans le repère local autour du point de fonctionnement c'est-à-dire sous la forme (1.45), (1.46) et (1.47) se réalise plus aisément et les calculs présentent l'avantage de pouvoir s'exprimer en récursif. Les matrices ΔY_{ens} , ΔX_{ens} et ΔU_{ens} correspondent à l'ensemble de mesures centrées autour du point de fonctionnement pour l'identification P_{ide} dont les coordonnées Y_{ide} , X_{ide} et U_{ide} correspondent à la moyenne des m mesures Y_{ens} , X_{ens} et U_{ens} .

Le vecteur des erreurs de modélisation par rapport à la variable de sortie y_j noté $E_{ens,j}$ correspond à l'ensemble des erreurs entre les mesures et le $j^{\text{ème}}$ hyperplan défini par la $j^{\text{ème}}$ ligne des matrices J_x et J_u . Ce vecteur est calculé de la manière suivante :

$$E_{ens,j} = \Delta Y_{ens,j} - J_{x,j} \Delta X_{ens} - J_{u,j} \Delta U_{ens} \quad (2.39)$$

La correction du modèle $\Delta J_{x,j}$ et $\Delta J_{u,j}$ influe sur les erreurs dont les variations valent :

$$\Delta E_{ens,j} = -\Delta J_{x,j} \Delta X_{ens} - \Delta J_{u,j} \Delta U_{ens} \quad (2.40)$$

Le critère est alors défini pour que les erreurs de modélisation par rapport à l'ensemble des mesures incluses dans la fenêtre d'observation soient minimisées et que les variations des coefficients du modèle soient atténuées.

$$H = \frac{1}{2} \left(\Delta J_{x,j} R_x \Delta J_{x,j}^T + \Delta J_{u,j} R_u \Delta J_{u,j}^T + (E_{ens,j} + \Delta E_{ens,j}) (E_{ens,j} + \Delta E_{ens,j})^T \right) \quad (2.41)$$

avec R_x et R_u des matrices positives pondérant les variations du modèle.

La minimisation de ce critère par rapport à $\Delta J_{x,j}$ et $\Delta J_{u,j}$ donne un système à résoudre qui s'écrit :

$$\begin{bmatrix} R_x + \Delta X_{ens} \Delta X_{ens}^T & \Delta X_{ens} \Delta U_{ens}^T \\ \Delta U_{ens} \Delta X_{ens}^T & R_u + \Delta U_{ens} \Delta U_{ens}^T \end{bmatrix} \begin{bmatrix} \Delta J_{x,j}^T \\ \Delta J_{u,j}^T \end{bmatrix} = \begin{bmatrix} \Delta X_{ens} \\ \Delta U_{ens} \end{bmatrix} E_{ens,j}^T \quad (2.42)$$

Les coefficients R_x et R_u se trouvent sur la diagonale de la matrice à inverser qui est par conséquent définie strictement positive. Cependant ces coefficients doivent être choisis petits pour ne pas nuire

à la rapidité d'adaptation du modèle. Le développement de l'équation précédente permet d'aboutir à un système dont les termes peuvent se calculer en récursif :

$$\begin{bmatrix} R_x + \Delta X_{ens} \Delta X_{ens}^T & \Delta X_{ens} \Delta U_{ens}^T \\ \Delta U_{ens} \Delta X_{ens}^T & R_u + \Delta U_{ens} \Delta U_{ens}^T \end{bmatrix} \begin{bmatrix} \Delta J_{x,j}^T \\ \Delta J_{u,j}^T \end{bmatrix} = \begin{bmatrix} \Delta X_{ens} \Delta Y_{ens,j}^T \\ \Delta U_{ens} \Delta Y_{ens,j}^T \end{bmatrix} - \begin{bmatrix} \Delta X_{ens} \Delta X_{ens}^T & \Delta X_{ens} \Delta U_{ens}^T \\ \Delta U_{ens} \Delta X_{ens}^T & \Delta U_{ens} \Delta U_{ens}^T \end{bmatrix} \begin{bmatrix} J_{x,j}^T \\ J_{u,j}^T \end{bmatrix} \quad (2.43)$$

Ce système linéaire peut également s'écrire sous la forme plus générale :

$$Ax = b \text{ avec } A \text{ une matrice symétrique.} \quad (2.44)$$

Cette seconde approche permet d'adapter directement les coefficients du modèle alors que la méthode d'adaptation précédente agit sur les ordonnées de points définissant le modèle.

4.5. Critère pour le modèle avec la zone inactive

Dans le cas du modèle avec la zone inactive défini par l'équation $\Delta Y = J_x \Delta X + J_u \Delta U + J_{uf} \Delta SU$, le principe d'adaptation reste le même. La matrice ΔSU_{ens} avec l'ensemble des mesures ΔSU est définie et intégrée dans le critère de manière à adapter la matrice J_{uf} . Les erreurs de modélisation et le critère valent alors :

$$\Delta E_{ens,j} = -\Delta J_{x,j} \Delta X_{ens} - \Delta J_{u,j} \Delta U_{ens} - \Delta J_{uf,j} \Delta SU_{ens} \quad (2.45)$$

et

$$H = \frac{1}{2} \left(\Delta J_{x,j} R_x \Delta J_{x,j}^T + \Delta J_{u,j} R_u \Delta J_{u,j}^T + \Delta J_{uf,j} R_{uf} \Delta J_{uf,j}^T + (E_{ens,j} + \Delta E_{ens,j})(E_{ens,j} + \Delta E_{ens,j})^T \right) \quad (2.46)$$

Les calculs de minimisation de ce critère donne le système suivant à résoudre :

$$\begin{bmatrix} R_x + \Delta X_{ens} \Delta X_{ens}^T & \Delta X_{ens} \Delta U_{ens}^T & \Delta X_{ens} \Delta SU_{ens}^T \\ \Delta U_{ens} \Delta X_{ens}^T & R_u + \Delta U_{ens} \Delta U_{ens}^T & \Delta U_{ens} \Delta SU_{ens}^T \\ \Delta SU_{ens} \Delta X_{ens}^T & \Delta SU_{ens} \Delta U_{ens}^T & R_{uf} + \Delta SU_{ens} \Delta SU_{ens}^T \end{bmatrix} \begin{bmatrix} \Delta J_{x,j}^T \\ \Delta J_{u,j}^T \\ \Delta J_{uf,j}^T \end{bmatrix} = \begin{bmatrix} \Delta X_{ens} \Delta Y_{ens,j}^T \\ \Delta U_{ens} \Delta Y_{ens,j}^T \\ \Delta SU_{ens} \Delta Y_{ens,j}^T \end{bmatrix} - \begin{bmatrix} \Delta X_{ens} \Delta X_{ens}^T & \Delta X_{ens} \Delta U_{ens}^T & \Delta X_{ens} \Delta SU_{ens}^T \\ \Delta U_{ens} \Delta X_{ens}^T & \Delta U_{ens} \Delta U_{ens}^T & \Delta U_{ens} \Delta SU_{ens}^T \\ \Delta SU_{ens} \Delta X_{ens}^T & \Delta SU_{ens} \Delta U_{ens}^T & \Delta SU_{ens} \Delta SU_{ens}^T \end{bmatrix} \begin{bmatrix} J_{x,j}^T \\ J_{u,j}^T \\ J_{uf,j}^T \end{bmatrix} \quad (2.47)$$

Le coefficient de pondération R_{uf} influant sur les termes J_{uf} doit être choisi plus grand que R_x et R_u . En effet on considère que l'évolution des seuils f de la zone inactive est plus lente que l'évolution des autres paramètres. Les variations de la matrice J_{uf} doivent donc être plus fortement atténuées. De plus quand toutes les valeurs des commandes sont positives ou négatives, la matrice ΔSU_{ens} est nulle. Aucune information n'est accessible pour adapter les valeurs de J_{uf} qui sont donc figées. De

même, les seuils f restent inchangés et ne doivent pas être recalculés. Lorsque ce cas se produit, les troisièmes lignes et colonnes des matrices du système (2.47) sont nulles et peuvent être supprimées. Le système à résoudre se simplifie et est alors similaire au système (2.43) étudié au chapitre précédent.

4.6. Critère pour le modèle déduit des fonctions de transfert

Pour le modèle déduit par conversion d'une fonction de transfert, le système à résoudre est particulier. En effet pour un système multivariable à ny sorties, ce modèle autour du point de fonctionnement s'écrit de la façon suivante :

$$\Delta Y = \begin{bmatrix} J_x & & & \\ & J_x & & \\ & & \ddots & \\ & & & J_x \end{bmatrix} \Delta X + J_u \Delta U \quad (2.48)$$

La matrice J_x est par définition de ce modèle la même pour toutes les sorties alors que les lignes $J_{u,j}$ de J_u diffèrent pour chaque sortie y_j . Les ensembles de mesures autour du point d'identification s'écrivent $\Delta Y_{ens,j}, \Delta X_{ens,j}$ pour la sortie y_j et ΔU_{ens} pour les commandes. Les erreurs de modélisation $E_{ens,j}$ pour les ny sorties valent alors :

$$\begin{cases} E_{ens,1} = \Delta Y_{ens,1} - J_x \Delta X_{ens,1} - J_{u,1} \Delta U_{ens} \\ E_{ens,2} = \Delta Y_{ens,2} - J_x \Delta X_{ens,2} - J_{u,2} \Delta U_{ens} \\ \vdots \\ E_{ens,ny} = \Delta Y_{ens,ny} - J_x \Delta X_{ens,ny} - J_{u,ny} \Delta U_{ens} \end{cases} \quad (2.49)$$

Le vecteur d'adaptation du modèle est défini comme suit :

$$\Delta J = \begin{bmatrix} \Delta J_x & \Delta J_{u,1} & \Delta J_{u,2} & \cdots & \Delta J_{u,ny} \end{bmatrix} \quad (2.50)$$

Ces variations modifient les vecteurs d'erreurs des quantités suivantes :

$$\begin{cases} \Delta E_{ens,1} = -\Delta J_x \Delta X_{ens,1} - \Delta J_{u,1} \Delta U_{ens} \\ \Delta E_{ens,2} = -\Delta J_x \Delta X_{ens,2} - \Delta J_{u,2} \Delta U_{ens} \\ \vdots \\ \Delta E_{ens,ny} = -\Delta J_x \Delta X_{ens,ny} - \Delta J_{u,ny} \Delta U_{ens} \end{cases} \quad (2.51)$$

Le critère pour l'identification comporte un premier terme pour minimiser les variations ΔJ du modèle et une somme de ny termes pour minimiser les erreurs $E_{ens,j} + \Delta E_{ens,j}$.

$$H = \frac{1}{2} \left(\Delta J R \Delta J^T + \sum_{j=1}^{ny} (E_{ens,j} + \Delta E_{ens,j}) (E_{ens,j} + \Delta E_{ens,j})^T \right) \quad (2.52)$$

La matrice R est une matrice diagonale positive qui pondère et limite les variations du modèle.

La minimisation de H par rapport aux variations ΔJ est alors obtenue en résolvant le système suivant :

$$\left(\begin{array}{cccc} \sum_{j=1}^{ny} \Delta X_{ens,j} \Delta X_{ens,j}^T & \Delta X_{ens,1} \Delta U_{ens}^T & \Delta X_{ens,2} \Delta U_{ens}^T & \cdots & \Delta X_{ens,ny} \Delta U_{ens}^T \\ \Delta U_{ens} \Delta X_{ens,1}^T & \Delta U_{ens} \Delta U_{ens}^T & & & \\ \Delta U_{ens} \Delta X_{ens,2}^T & & \Delta U_{ens} \Delta U_{ens}^T & & \\ \vdots & & & \ddots & \\ \Delta U_{ens} \Delta X_{ens,ny}^T & & & & \Delta U_{ens} \Delta U_{ens}^T \end{array} \right) + R \Delta J^T = \begin{array}{c} \sum_{j=1}^{ny} \Delta X_{ens,j} \Delta E_{ens,j}^T \\ \Delta U_{ens} \Delta E_{ens,1}^T \\ \Delta U_{ens} \Delta E_{ens,2}^T \\ \vdots \\ \Delta U_{ens} \Delta E_{ens,ny}^T \end{array} \quad (2.53)$$

Le calcul de ΔJ permet alors d'adapter le modèle pour les ny sorties. L'inconvénient de cette approche réside dans la dimension du système à résoudre qui est plus importante que celle des systèmes obtenus avec les lois d'adaptation précédentes. Toutefois, la résolution du système peut être simplifiée grâce à la structure particulière des matrices de l'équation (2.53). Les termes $\Delta J_{u,i}$ peuvent s'écrire en fonction de ΔJ_x pour donner les relations suivantes :

$$\Delta J_{u,i}^T = (\Delta U_{ens} \Delta U_{ens}^T + R_u)^{-1} \Delta U_{ens} \Delta E_{ens,i}^T - (\Delta U_{ens} \Delta U_{ens}^T + R_u)^{-1} \Delta U_{ens} \Delta X_{ens,j}^T \Delta J_x^T \quad (2.54)$$

La substitution de ces termes dans la première équation du système (2.55) fournit la relation ci-dessous dont la résolution donne ΔJ_x :

$$\begin{aligned} \sum_{j=1}^{ny} (\Delta X_{ens,j} \Delta X_{ens,j}^T + R_x - \Delta X_{ens,j} \Delta U_{ens}^T (\Delta U_{ens} \Delta U_{ens}^T + R_u)^{-1} \Delta U_{ens} \Delta X_{ens,j}^T) \Delta J_x^T = \\ \sum_{j=1}^{ny} (\Delta X_{ens,j} \Delta E_{ens,j}^T - \Delta X_{ens,j} \Delta U_{ens}^T (\Delta U_{ens} \Delta U_{ens}^T + R_u)^{-1} \Delta U_{ens} \Delta E_{ens,j}^T) \end{aligned} \quad (2.56)$$

Les termes ΔJ_u^T se déduisent alors par substitution du terme ΔJ_x dans les équations (2.54). Cette méthode de résolution requiert l'inversion complète de la matrice $\Delta U_{ens} \Delta U_{ens}^T + R_u$ mais permet d'aboutir à un algorithme d'adaptation du modèle plus performant en temps de calcul.

4.7. Rappels sur la résolution de systèmes linéaires

Malgré l'ajout de contraintes visant à empêcher l'apparition de matrices singulières, la résolution du système $Ax = b$ sur calculateur n'est pas toujours possible si les ordres de grandeur des coefficients de la matrice A sont fortement différents. Le conditionnement des matrices se révèle alors indispensable pour obtenir un algorithme robuste. Le conditionnement se réalise en multipliant A à gauche et à droite par une matrice diagonale D_c , définie pour que les termes diagonaux de la matrice A conditionnée notée A_c valent 1.

Le système $Ax = b$ s'écrit alors $D_c A D_c D_c^{-1} x = D_c b$ avec la matrice D_c choisie de la manière suivante :

$$D_c = \begin{bmatrix} \sqrt{A_{1,1}} & & & \\ & \sqrt{A_{2,2}} & & \\ & & \ddots & \\ & & & \sqrt{A_{n,n}} \end{bmatrix}^{-1} \quad (2.57)$$

Lorsqu'un terme $A_{i,i}$ est nul ou tend vers une valeur très faible, le calcul de D_c n'est pas réalisable ou tend vers des valeurs très importantes, ce qui risque de donner des résultats peu précis pour les calculs ultérieurs. Cette situation est évitée grâce à l'ajout des termes limitant les variations des paramètres dans les différents critères définis précédemment. En effet ces termes viennent ajouter sur la diagonale de la matrice A les coefficients positifs notés R , R_x , R_u ou R_{uf} suivant le critère employé.

En posant $A_c = D_c A D_c$, $x_c = D_c^{-1} x$ et $b_c = D_c b$ le système conditionné à résoudre devient :

$$A_c x_c = b_c \quad (2.58)$$

La solution x du système initial s'obtient alors avec la relation :

$$x = D_c x_c \quad (2.59)$$

La résolution du système après conditionnement peut s'effectuer en calculant l'inverse de la matrice A_c . Cette solution n'est pas satisfaisante car le nombre d'opérations nécessaires pour l'inversion est important et de plus les algorithmes d'inversion de matrices ne donnent pas toujours des résultats précis. La résolution du système par une méthode de Cholesky [GOL 89] s'avère plus efficace et plus précise. L'objectif est de décomposer la matrice A symétrique définie positive en un produit de matrices LDL^T avec L une matrice triangulaire inférieure avec les termes diagonaux égaux à 1, et D une matrice diagonale. Avec cette factorisation qui est une décomposition correspondant à un cas particulier de la décomposition LU d'une matrice, la solution du système s'obtient en trois étapes qui consistent à résoudre trois systèmes dont les solutions intermédiaires sont appelées x_1 et x_2 .

$$\begin{cases} Lx_1 = b_c \\ Dx_2 = x_1 \\ L^T x_c = x_2 \end{cases} \quad (2.60)$$

La résolution des systèmes $Lx_1 = b_c$ et $L^T x_c = x_2$ est immédiate car les matrices L et L^T sont triangulaires et ont une diagonale dont les termes valent 1 par construction et définition de la factorisation. Les solutions de ces systèmes se déduisent donc par simples substitutions. La résolution du système par cette technique est efficace en temps de calcul, et convient pour l'identification en ligne.

5. Expression des calculs en récursif

5.1. Principe et intérêts

L'objectif principal est d'effectuer les calculs des matrices du type $\Delta X_{ens} \Delta X_{ens}^T$, $\Delta X_{ens} \Delta U_{ens}^T$ ou $\Delta X_{ens} \Delta Y_{ens}^T$ dans des temps plus courts et d'obtenir un algorithme dont le nombre de points de mesures peut être choisi arbitrairement grand sans augmenter le nombre de calculs requis. L'intérêt est donc double, car d'un point de vue pratique ou économique, la mise en oeuvre de la loi de commande nécessite des processeurs moins puissants, le second avantage d'un point de vue algorithmique est d'aboutir à des calculs dont la complexité est indépendante du nombre de points de mesures. On s'affranchit donc d'une contrainte pour la mise en oeuvre. Cependant les calculs récursifs engendrent d'autres problèmes qui n'apparaissent qu'à très long terme ou seulement dans des conditions extrêmes, mais qu'il convient de résoudre afin de garantir la robustesse numérique de l'algorithme. Ce problème est lié aux erreurs d'arrondi qui biaisent les estimations et qui s'accumulent dans certaines circonstances. Une étude détaillée sur ce problème doit être menée. La transcription des calculs en récursif s'effectue en trois étapes. La première étape consiste à mettre à jour les valeurs du point de fonctionnement pour l'identification. Ce changement introduit un terme dans la matrice de covariance. La seconde phase correspond à l'ajout du nouveau point, et la dernière étape tient compte du point le plus ancien qu'il faut retirer. Le calcul de covariance à l'instant k sur m points entre deux signaux x et y autour de leur valeur moyenne donne le résultat suivant :

$$C_{xy,k} = \sum_{i=k+1-m}^k (x_i - x_{ide,k})(y_i - y_{ide,k}) \quad (2.61)$$

$$\text{avec } x_{ide,k} = \frac{1}{m} \sum_{i=k+1-m}^k x_i \text{ et } y_{ide,k} = \frac{1}{m} \sum_{i=k+1-m}^k y_i$$

A l'instant $k+1$, le résultat est égal à:

$$C_{xy,k+1} = \sum_{i=k+2-m}^{k+1} (x_i - x_{ide,k+1})(y_i - y_{ide,k+1}) \quad (2.62)$$

$$\text{avec } x_{ide,k+1} = \frac{1}{m} \sum_{i=k+2-m}^{k+1} x_i \text{ et } y_{ide,k+1} = \frac{1}{m} \sum_{i=k+2-m}^{k+1} y_i$$

Le passage entre les instants k et $k+1$ s'obtient par comparaison entre les deux expressions.

Nous avons donc d'une part pour la mise à jour de la moyenne les relations:

$$\begin{aligned} x_{ide,k+1} &= x_{ide,k} + \frac{x_{k+1} - x_{k+1-m}}{m} \\ y_{ide,k+1} &= y_{ide,k} + \frac{y_{k+1} - y_{k+1-m}}{m} \end{aligned} \quad (2.63)$$

et d'autre part pour le calcul de la covariance :

$$\begin{aligned} C_{xy,k+1} &= C_{xy,k} + (x_{k+1} - x_{ide,k+1})(y_{k+1} - y_{ide,k+1}) - \\ &- (x_{k+1-m} - x_{ide,k+1})(y_{k+1-m} - y_{ide,k+1}) + \frac{1}{m}(x_{k+1} - x_{k+1-m})(y_{k+1} - y_{k+1-m}) \end{aligned} \quad (2.64)$$

Ces équations s'appliquent ensuite pour le calcul des matrices de covariance notées A_{xx} et B_{xy} .

$$\begin{aligned} A_{xx,k+1} &= A_{xx,k} + (X_{k+1} - X_{ide,k+1})(X_{k+1} - X_{ide,k+1})^T - \\ &- (X_{k+1-m} - X_{ide,k+1})(X_{k+1-m} - X_{ide,k+1})^T + \frac{1}{m}(X_{k+1} - X_{k+1-m})(X_{k+1} - X_{k+1-m})^T \end{aligned} \quad (2.65)$$

$$\begin{aligned} B_{xy,k+1} &= B_{xy,k} + (X_{k+1} - X_{ide,k+1})(Y_{k+1} - Y_{ide,k+1})^T - \\ &- (X_{k+1-m} - X_{ide,k+1})(Y_{k+1-m} - Y_{ide,k+1})^T + \frac{1}{m}(X_{k+1} - X_{k+1-m})(Y_{k+1} - Y_{k+1-m})^T \end{aligned} \quad (2.66)$$

La conversion des calculs en récursif ne pose alors aucune difficulté en ce qui concerne le formalisme. En revanche, des problèmes numériques et une dégradation de la précision des résultats surviennent lors de la mise en oeuvre, car l'algorithme devient plus sensible aux erreurs d'arrondi.

5.2. Robustesse de l'algorithme

La mise en oeuvre d'un algorithme récursif sur calculateur doit être effectuée en tenant compte de la précision à laquelle sont effectués les calculs. Les erreurs d'arrondi se produisent sur les chiffres les moins significatifs et sont par conséquent souvent jugées négligeables. Par exemple, lors d'une somme de valeurs de même ordre de grandeur, les erreurs d'arrondi ont une valeur moyenne nulle en posant comme hypothèse que le bruit de calcul est un bruit blanc de moyenne nulle. Cette hypothèse apparaît cependant trop simplificatrice car l'erreur d'arrondi est un phénomène non linéaire [BLA 91]. Des exemples triviaux montrent quelles peuvent être les conséquences d'une succession d'erreurs commises à chaque pas de calcul. En effet la précision e d'un réel codé sur un calculateur est définie comme étant le plus grand nombre pour lequel la relation « $1+e = 1$ » est vérifiée. L'addition du terme e répétée $1/e$ fois conduirait donc non pas à la valeur 2 mais donnerait sur calculateur toujours la valeur 1. Cet exemple trivial mais typique a le mérite d'illustrer simplement un des dangers du calcul numérique. Un autre problème classique appelé « cancellation » arrive lorsqu'on effectue la soustraction entre deux valeurs très proches l'une de l'autre. On constate alors une perte des chiffres les plus significatifs pour aboutir à une valeur peu précise et entachée d'erreurs relatives plus importantes.

Chapitre II : Adaptation des modèles

Un troisième problème numérique survient à cause des variations de l'ordre de grandeur des mesures. Par conséquent, avec des nombres en virgule flottante, l'ordre de grandeur des erreurs d'arrondi fluctue aussi. Lors de variations importantes, et suite à l'ajout et au retrait des points de mesure, la moyenne des erreurs n'est plus nulle et on observe à nouveau une perte de chiffres significatifs, et ceci même sur des opérations aussi simples que la moyenne récursive. Après de brusques augmentations et diminutions des mesures, une erreur résiduelle non négligeable apparaît. Ces problèmes sont aggravés pour les calculs avec des produits entre les mesures ou avec des termes au carré. Les erreurs sont alors encore plus importantes et créent des dérives faussant nettement les résultats de l'adaptation, ce qui peut provoquer la divergence des paramètres. La notion de précision des calculs est donc à prendre en considération lors de l'établissement des algorithmes.

Ces problèmes d'analyse numérique bien que très connus, sont souvent sous estimés et négligés. Nous nous plaçons dans les conditions les moins favorables afin de faire apparaître volontairement ces phénomènes, et de vérifier que les améliorations et méthodes proposées permettent bien de résoudre ces problèmes. En l'occurrence, pour tester la robustesse numérique, des mesures évoluant artificiellement dans une grande plage de valeurs sont générées. De plus, le codage utilisé pour les réels est en simple précision, ce qui correspond à une précision e égale approximativement à $1.19e-07$. Les calculs s'effectuent alors avec seulement 7 chiffres significatifs. L'essai illustrant ces problèmes compare la covariance d'un signal calculée récursivement et entièrement. Le signal de test est une succession de rampes dont les pentes sont définies positives puis négatives. Lors des changements de pente qui s'effectuent toutes les secondes, le signal de test vaut 1, 10, 95, 100, 950, 1000, 9500, 10000 puis 9500, 1000, 950, 100, 95, 10 et 1. Le calcul de la covariance du signal s'effectue sur 50 points à une période de 0.01 seconde. Le signal, la moyenne du signal et sa covariance calculés avec l'algorithme récursif et non récursif sur 50 points sont représentés sur la figure 2.3. Les erreurs relatives entre les deux méthodes de calcul montrent qu'après des variations importantes du signal, la précision des résultats est amoindrie. Pour le calcul de la covariance, les erreurs d'arrondi deviennent même supérieures à la valeur exacte.

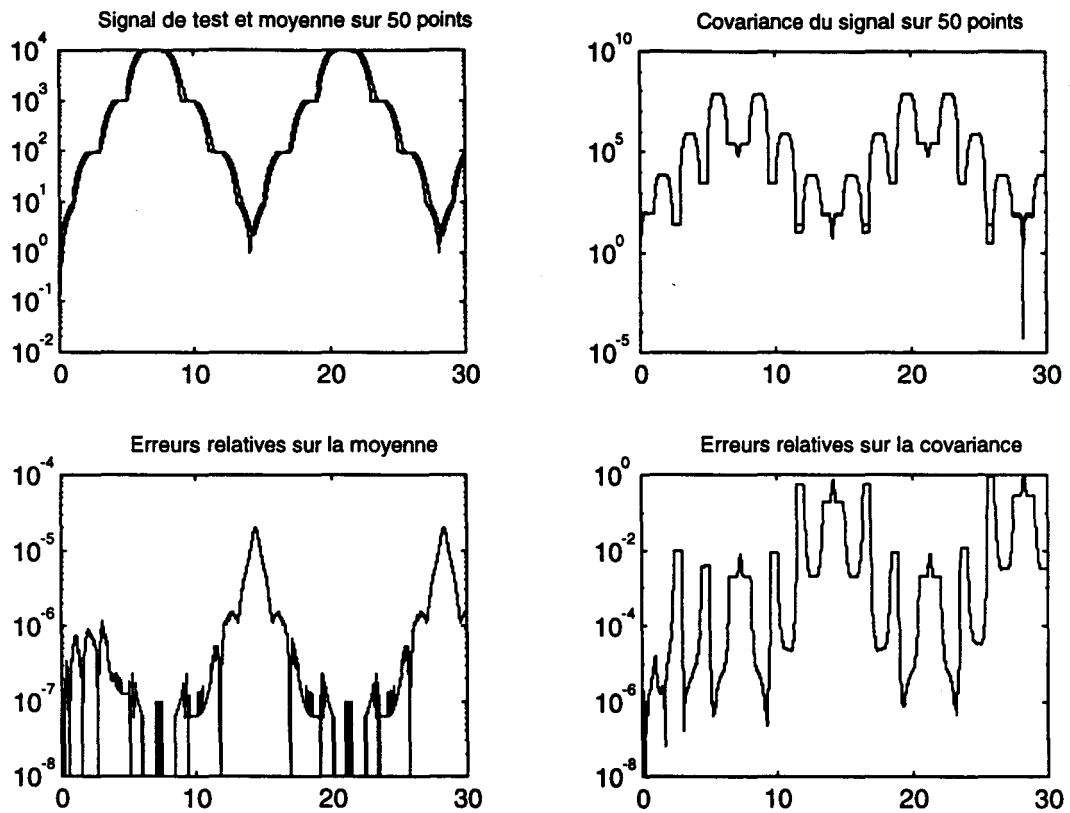


Figure 2.3 : Erreurs entre les algorithmes récursif et non récursif.

Pour remédier aux difficultés du calcul récursif et corriger les dérives à long terme ou survenant après des variations importantes des signaux, une première solution est d'effectuer à intervalle régulier les calculs complets, pour réactualiser les résultats en se basant sur l'ensemble des m derniers points mémorisés. Cependant lors de la correction, si les erreurs d'arrondi sont importantes, il se produit une brusque variation des valeurs des matrices et donc des coefficients du modèle puis de la commande. Ces variations sont évidemment indésirables. Inversement, si les erreurs sont faibles, les calculs de réactualisation deviennent inutiles. Une autre méthode de correction des erreurs est de recalculer complètement les matrices que lorsque des variations importantes des mesures ont lieu. En effet les erreurs d'arrondi se produisent principalement lors de fortes variations. Mais, l'estimation du seuil à partir duquel il s'avère nécessaire de corriger les dérives constitue une difficulté supplémentaire embarrassante. Quand bien même ce problème serait surmonté, le principal inconvénient de ces deux premières approches est qu'elles nécessitent encore d'effectuer le calcul complet à certains instants, ce qui limite l'intérêt du calcul récursif.

La solution préconisée pour garder les avantages des méthodes purement récursives tout en corrigeant les erreurs d'arrondi, est de réinitialiser les calculs de manière récursive. On définit alors des matrices primaires et secondaires, qui correspondent respectivement aux matrices en cours d'utilisation et aux matrices en cours de réinitialisation. Le principe est alors le suivant. Les calculs

effectués sur les matrices primaires sont identiques à ceux présentés précédemment. En revanche, tous les m points, les matrices secondaires sont remises à zéro, en considérant comme point de fonctionnement la mesure courante. Les $m-1$ points suivants servent alors à réinitialiser récursivement les matrices secondaires. Après $m-1$ itérations, les matrices secondaires et primaires sont donc formellement équivalentes. Seule la manière dont elles sont calculées change. Les différences numériques sont donc dues aux erreurs d'arrondi, qui sont commises en toute rigueur à la fois sur les matrices primaires et secondaires. Cependant comme les matrices secondaires sont remises à zéro et réinitialisées tous les m points, leurs valeurs peuvent être supposées moins entachées d'erreurs. Les valeurs des matrices primaires peuvent alors être remplacées par celles des matrices secondaires. Ce principe se schématise de la manière suivante.

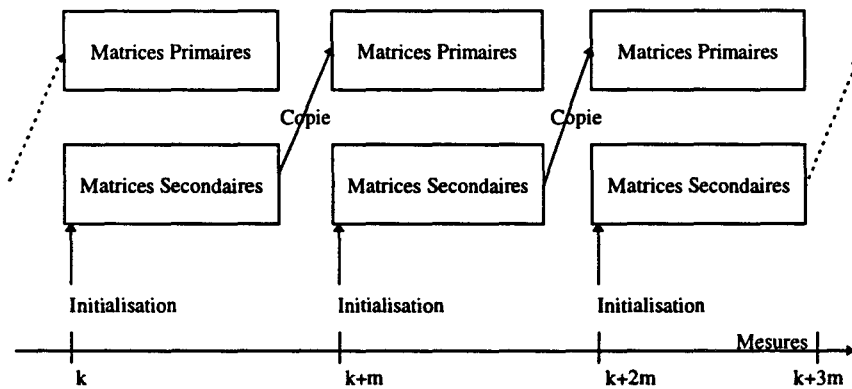


Figure 2.4 : Principe de correction des dérives

L'initialisation des matrices secondaires s'effectue alors toutes les m mesures aux instants k_{ini} en prenant :

$$X_{sec,ide} = X_{k_{ini}} \quad (2.67)$$

$$Y_{sec,ide} = Y_{k_{ini}} \quad (2.68)$$

$$A_{sec,xx} = [0] \quad (2.69)$$

$$\text{et } B_{sec,xy} = [0] \quad (2.70)$$

La mise à jour des matrices secondaires entre l'instant k_{ini} et $k_{ini}+m-1$ se calcule à partir des équations suivantes :

$$X_{sec,ide,k+1} = X_{sec,ide,k} + \frac{1}{m}(X_{k+1} - X_{k_{ini}}) \quad (2.71)$$

$$Y_{sec,ide,k+1} = Y_{sec,ide,k} + \frac{1}{m}(Y_{k+1} - Y_{k_{ini}})$$

Chapitre II : Adaptation des modèles

$$A_{sec,xx,k+1} = A_{sec,xx,k} + (X_{k+1} - X_{sec,ide,k+1})(X_{k+1} - X_{sec,ide,k+1})^T - (X_{kini} - X_{sec,ide,k+1})(X_{kini} - X_{sec,ide,k+1})^T + \frac{1}{m}(X_{k+1} - X_{kini})(X_{k+1} - X_{kini})^T \quad (2.72)$$

$$B_{sec,xy,k+1} = B_{sec,xy,k} + (X_{k+1} - X_{sec,ide,k+1})(Y_{k+1} - Y_{sec,ide,k+1})^T - (X_{kini} - X_{sec,ide,k+1})(Y_{kini} - Y_{sec,ide,k+1})^T + \frac{1}{m}(X_{k+1} - X_{kini})(Y_{k+1} - Y_{kini})^T \quad (2.73)$$

Cette technique reste donc entièrement récursive et permet d'annuler les dérives dues aux imprécisions des calculs numériques et aux erreurs d'arrondi. La figure 2.5 ci-dessous représente les résultats obtenus avec l'algorithme récursif modifié, pour le signal de test défini précédemment à la figure 2.3.

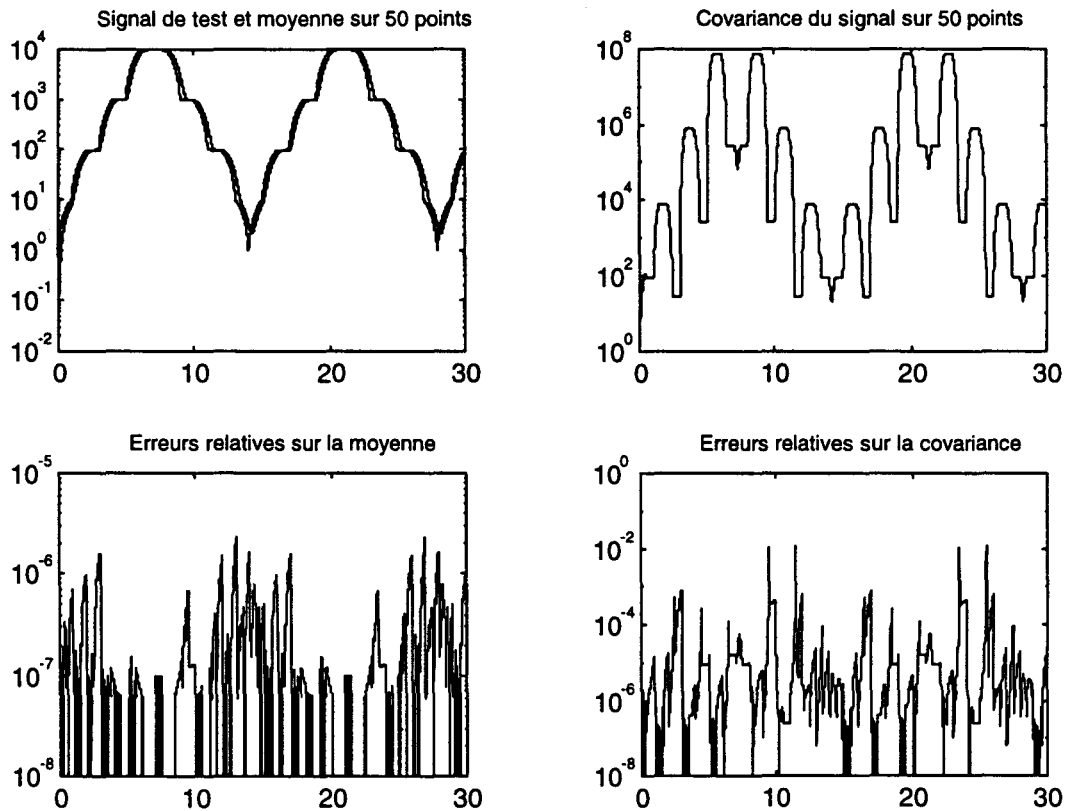


Figure 2.5 : Erreurs entre les algorithmes récursif corrigé et non récursif.

Les erreurs relatives restent inférieures à 10^{-2} pour le calcul de la covariance même après de fortes variations du signal. L'algorithme modifié résout donc le problème de la dégradation de la précision mais au prix d'un doublement des calculs. Ce surcoût en nombre de calculs n'est pas un obstacle rédhibitoire, car la méthode récursive reste plus intéressante et plus rapide dès que le nombre de points utilisés est supérieur à une quarantaine. En dessous de ce seuil, le calcul récursif n'apporte pas de réel gain et par conséquent le calcul complet des matrices est préférable. Cependant comme

le nombre de points utilisés est généralement supérieur à une cinquantaine, l'implémentation robuste des calculs en récursif est avantageuse et fournit des résultats probants.

5.3. Inversion en récursif

Les calculs récursifs du chapitre précédent portent sur les matrices du type $\Delta X_{ens} \Delta X_{ens}^T$, $\Delta X_{ens} \Delta U_{ens}^T$ et $\Delta X_{ens} \Delta Y_{ens}^T$ utilisées pour former les matrices A et b du système à résoudre $Ax = b$. Pour éviter d'inverser la matrice A à chaque instant d'identification ou d'avoir à résoudre ce système, qui est un traitement coûteux en calcul, il est judicieux de calculer récursivement A^{-1} pour déterminer x ensuite en effectuant un simple produit $x = A^{-1}b$. Les méthodes récursives employées pour calculer l'inverse de la matrice A sont basées sur le lemme d'inversion matricielle [RIC 91], [BOR 92], dont la preuve est immédiate en multipliant le résultat par la matrice $[A + BCD]$.

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1} \quad (2.74)$$

Pour se ramener à une expression calculable aisément et plus spécifique au problème de l'identification en ligne, le lemme d'inversion peut s'écrire :

$$[A + zz^T]^{-1} = A^{-1} - \frac{vv^T}{1 + z^T v} \quad (2.75)$$

avec

z le vecteur correspondant au point à inclure dans le calcul de l'inverse de A .

et $v = A^{-1}z$

Dans le cas où l'on souhaite retirer un point z de la matrice A , l'expression devient :

$$[A - zz^T]^{-1} = A^{-1} - \frac{vv^T}{-1 + z^T v} \quad (2.76)$$

Dans le cas de l'adaptation du modèle présenté au chapitre II.4.4, la matrice A est la suivante :

$$A = \begin{bmatrix} R_x + \Delta X_{ens} \Delta X_{ens}^T & \Delta X_{ens} \Delta U_{ens}^T \\ \Delta U_{ens} \Delta X_{ens}^T & R_u + \Delta U_{ens} \Delta U_{ens}^T \end{bmatrix} \quad (2.77)$$

La première étape pour le calcul de A^{-1} en récursif consiste à initialiser la matrice A^{-1} en posant :

$$A^{-1} = \begin{bmatrix} R_x^{-1} & \\ & R_u^{-1} \end{bmatrix} \quad (2.78)$$

La mise à jour à chaque instant d'identification de l'inverse de A s'effectue ensuite en calculant la nouvelle valeur du point de fonctionnement, en ajoutant avec la formule (2.75) le nouveau point noté z_1 , en ajoutant également avec l'équation (2.75) le terme z_2 dû au changement du point de

fonctionnement, et en retirant avec la formule (2.76) le point le plus ancien z_3 . Les trois points z_1, z_2, z_3 valent respectivement :

$$z_1 = \begin{bmatrix} X_{k+1} - X_{ide,k+1} \\ U_{k+1} - U_{ide,k+1} \end{bmatrix}, z_2 = \frac{1}{\sqrt{m}} \begin{bmatrix} X_{k+1} - X_{k+1-m} \\ U_{k+1} - U_{k+1-m} \end{bmatrix} \text{ et } z_3 = \begin{bmatrix} X_{k+1-m} - X_{ide,k+1} \\ U_{k+1-m} - U_{ide,k+1} \end{bmatrix} \quad (2.79)$$

Le même genre de problèmes numériques se produit et nous oblige à corriger les dérives résultant des erreurs d'arrondi. La technique employée pour le calcul de A en récursif s'adapte sans difficulté au calcul de A^{-1} . Cependant l'initialisation pose de plus nombreux problèmes, car les valeurs initiales de A^{-1} sont très grandes et peuvent décroître rapidement. Les ordres de grandeur des coefficients de A^{-1} varient donc fortement ce qui est à l'origine d'erreurs d'arrondi plus importantes. On constate alors que la précision et la robustesse de la loi d'adaptation sont meilleures en résolvant le système $Ax=b$ avec le conditionnement des matrices et la factorisation de Cholesky. Le calcul récursif de A^{-1} , malgré ses avantages en terme de temps de calcul, se révèle donc être une solution moins intéressante du point de vue de la précision et de la robustesse.

6. Robustesse de la loi d'adaptation

6.1. Influence du bruit et filtrage des mesures

La présence de bruits sur les mesures influe sur les matrices Y_{ens} et X_{ens} et par conséquent sur les matrices A et b des systèmes définis précédemment. Les bruits sont supposés être des signaux aléatoires, non corrélés, centrés et de moyennes nulles.

Dans le cas de l'adaptation du modèle (1.80), les bruits définis à l'aide des matrices $B_{y,ens}$ et $B_{x,ens}$ s'ajoutent aux mesures et modifient la matrice A initiale en une nouvelle matrice $A+A_b$, avec A_b les termes traduisant l'influence du bruit.

$$A_b = \begin{bmatrix} b_{x,ens} b_{x,ens}^T + b_{x,ens} \Delta X_{ens}^T + \Delta X_{ens} b_{x,ens}^T & b_{x,ens} \Delta U_{ens}^T \\ b_{x,ens} \Delta X_{ens}^T & 0 \end{bmatrix} \quad (2.81)$$

Sur le vecteur b , les bruits génèrent des variations notées b_b qui valent :

$$b_b = \begin{bmatrix} b_{x,ens} b_{y,ens,j}^T + b_{x,ens} \Delta Y_{ens,j}^T + \Delta X_{ens} b_{y,ens,j}^T \\ \Delta U_{ens} b_{y,ens,j}^T \end{bmatrix} - A_b \begin{bmatrix} J_{x,j}^T \\ J_{u,j}^T \end{bmatrix} \quad (2.82)$$

Les hypothèses formulées sur les bruits permettent d'effectuer des simplifications et d'aboutir aux conclusions suivantes. Seul le produit $b_{x,ens} b_{x,ens}^T$ est non nul et donne une matrice diagonale notée D_b avec des valeurs positives. Les autres termes ont une espérance mathématique nulle car ils

correspondent à des produits entre les mesures et les bruits qui ne sont pas corrélés entre eux, et d'autre part parce que les bruits sont centrés et de moyenne nulle. Les matrices A_b et b_b s'écrivent donc :

$$A_b = \begin{bmatrix} D_b & 0 \\ 0 & 0 \end{bmatrix} \text{ et } b_b = - \begin{bmatrix} D_b J_{x,j}^T \\ 0 \end{bmatrix} \text{ avec } D_b = b_{x,ens} b_{x,ens}^T \quad (2.83)$$

Les résultats obtenus pour l'adaptation du modèle sont donc faussés par les bruits de mesures. Pour réduire leur influence, il est nécessaire de filtrer les données. Les techniques de filtrage basées sur un modèle du processus ne peuvent pas s'appliquer car le modèle est supposé inconnu et doit en l'occurrence être déterminé. L'utilisation du modèle estimé pour filtrer les mesures servant à l'adaptation de ce même modèle s'avère délicate et conduit à des résultats instables si des perturbations, des non linéarités ou des variations de paramètres se produisent.

Les bruits sont des variations à hautes fréquences, que l'on peut atténuer en appliquant un filtre passe-bas sur les mesures. Cette solution est efficace si la fréquence de coupure du filtre est bien réglée. Les résultats obtenus sont alors bien meilleurs avec le filtrage que sans. Pourtant ce traitement déforme les signaux et introduit une dynamique supplémentaire qui fausse l'estimation des coefficients du modèle. La solution retenue est d'appliquer une matrice de filtrage F sur les mesures et d'estimer les paramètres du modèle suivant.

$$\Delta Y_{set} F = J_x \Delta X_{set} F + J_u \Delta U_{set} F \quad (2.84)$$

Avec ces données filtrées, le système à résoudre devient :

$$\begin{bmatrix} R_x + \Delta X_{ens} F F^T \Delta X_{ens}^T & \Delta X_{ens} F F^T \Delta U_{ens}^T \\ \Delta U_{ens} F F^T \Delta X_{ens}^T & R_u + \Delta U_{ens} F F^T \Delta U_{ens}^T \end{bmatrix} \begin{bmatrix} \Delta J_{x,j}^T \\ \Delta J_{u,j}^T \end{bmatrix} = \begin{bmatrix} \Delta X_{ens} F F^T \Delta Y_{ens,j}^T \\ \Delta U_{ens} F F^T \Delta Y_{ens,j}^T \end{bmatrix} - \begin{bmatrix} \Delta X_{ens} F F^T \Delta X_{ens}^T & \Delta X_{ens} F F^T \Delta U_{ens}^T \\ \Delta U_{ens} F F^T \Delta X_{ens}^T & \Delta U_{ens} F F^T \Delta U_{ens}^T \end{bmatrix} \begin{bmatrix} J_{x,j}^T \\ J_{u,j}^T \end{bmatrix} \quad (2.85)$$

Les matrices A_{bf} et b_{bf} dues aux bruits valent après filtrage :

$$A_{bf} = \begin{bmatrix} b_{x,ens} F F^T b_{x,ens}^T + b_{x,ens} F F^T \Delta X_{ens}^T + \Delta X_{ens} F F^T b_{x,ens}^T & b_{x,ens} F F^T \Delta U_{ens}^T \\ b_{x,ens} F F^T \Delta X_{ens}^T & 0 \end{bmatrix} \quad (2.86)$$

$$b_{bf} = \begin{bmatrix} b_{x,ens} F F^T b_{y,ens,j}^T + b_{x,ens} F F^T \Delta Y_{ens,j}^T + \Delta X_{ens} F F^T b_{y,ens,j}^T \\ \Delta U_{ens} F F^T b_{y,ens,j}^T \end{bmatrix} - A_{bf} \begin{bmatrix} J_{x,j}^T \\ J_{u,j}^T \end{bmatrix} \quad (2.87)$$

d'où en tenant compte des hypothèses sur le bruit :

$$A_{bf} = \begin{bmatrix} b_{x,ens} F F^T b_{x,ens}^T & 0 \\ 0 & 0 \end{bmatrix} \text{ et } b_{bf} = \begin{bmatrix} b_{x,ens} F F^T b_{y,ens,j}^T - b_{x,ens} F F^T b_{x,ens}^T J_{x,j}^T \\ 0 \end{bmatrix} \quad (2.88)$$

La matrice F de filtrage est déterminée de manière à réduire les normes des matrices A_b et b_b afin d'amoinrir la sensibilité de l'adaptation par rapport aux bruits. Ceux-ci étant supposés de

moyennes nulles, la réduction de leur influence peut s'effectuer en calculant la moyenne des mesures sur nbf points consécutifs.

$$F = \frac{1}{nbf} \begin{bmatrix} I & & & & \\ \vdots & I & & & \\ I & \vdots & \ddots & & \\ & I & & I & \\ & & \ddots & \vdots & \\ & & & & I \end{bmatrix} \quad (2.89)$$

Ce filtrage s'effectue sur les mesures de la fenêtre d'observation lors de l'acquisition. Cette solution est simple et atténue efficacement les bruits, mais le modèle obtenu est biaisé à cause du traitement réalisé. Les matrices A et b du système à résoudre sont effectivement modifiées par la matrice de filtrage F . Le nombre nbf de points utilisés pour le calcul de la moyenne doit alors être le plus petit possible, et le filtrage ne doit donc être réalisé que si les bruits sont importants et nuisent véritablement aux calculs d'adaptation du modèle.

Pour tester l'influence des bruits sur l'adaptation des coefficients du modèle, un bruit blanc d'écart-type 0.1 est ajouté sur la sortie du système $y_{k+1} = ay_k + bu_k$ avec $a = 0.95$ et $b = 1$. Le système est simulé à une période d'échantillonnage de 0.01s. Le seuil σ de variation minimale de la sortie par rapport au dernier point inséré dans la fenêtre d'observation doit être supérieur à l'écart-type du bruit. Celui-ci est alors choisi égal à 0.2. L'identification s'effectue en ligne avec le contrôleur défini au chapitre IV.6 et en imposant au système de suivre une consigne qui est une suite de créneaux d'amplitude ± 1 à une fréquence de 1 rad/s.

Les valeurs des paramètres a et b sont tracées sur les figures 2.6 et 2.7 pour montrer l'influence du bruit. Sans filtrage les paramètres évoluent dans de fortes proportions et peuvent être très éloignés des valeurs réelles. La commande qui en résulte est alors très irrégulière et fournit des résultats médiocres.

Chapitre II : Adaptation des modèles

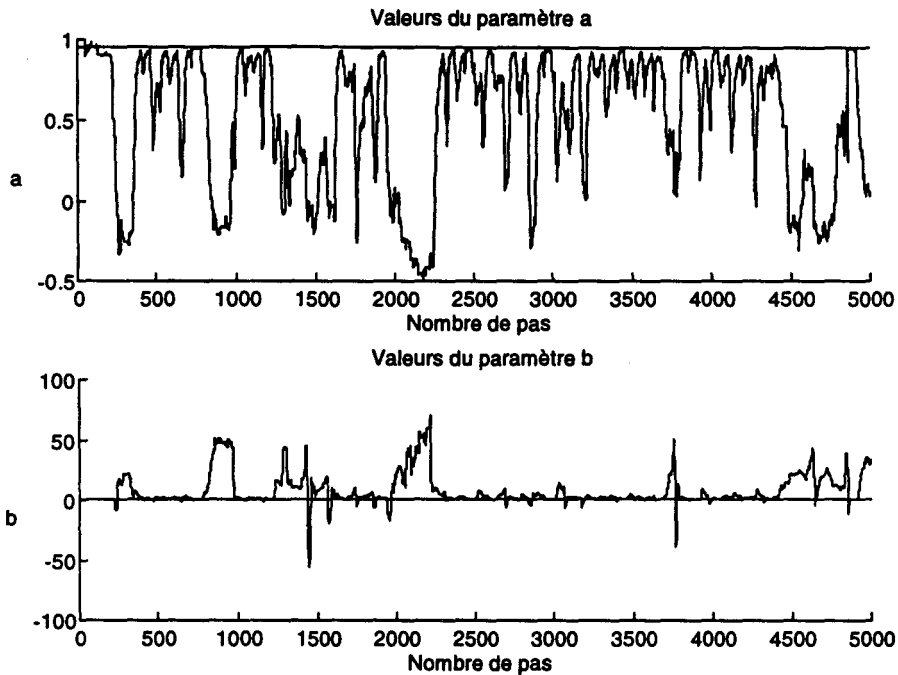


Figure 2.6 : Identification sans filtrage ($nbf=1$).

Avec un filtrage sur 10 points, les paramètres a et b sont plus réguliers et plus proches des valeurs réelles.

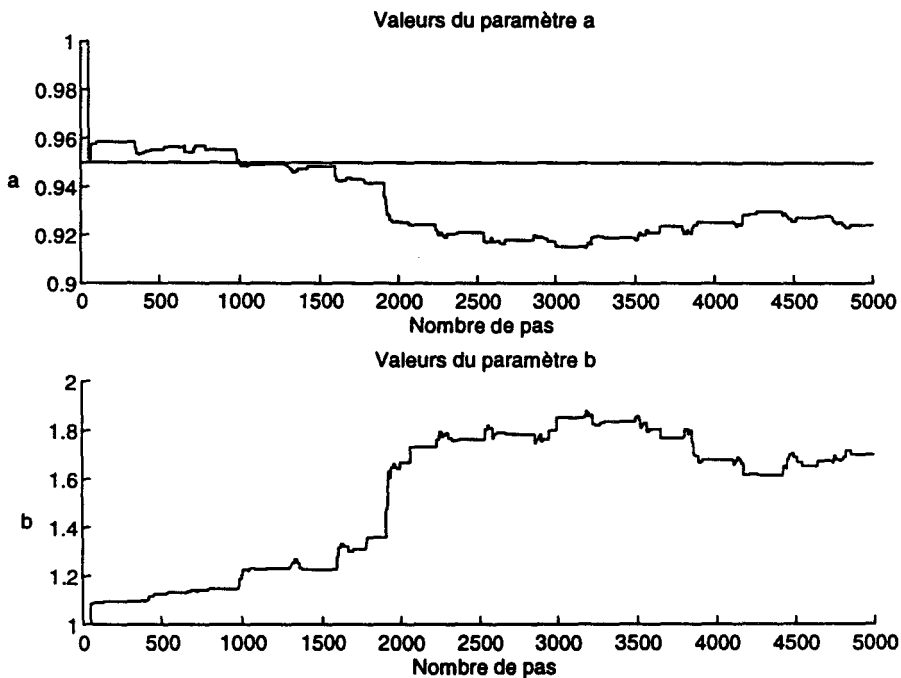


Figure 2.7 : Identification avec un filtrage sur 10 mesures ($nbf=10$).

Le filtrage introduit cependant un biais sur a et b . Par conséquent le modèle obtenu ne correspond pas précisément au système réel, mais donne une approximation suffisamment correcte du processus pour en déduire une loi de commande satisfaisante. L'identification avec filtrage des mesures améliore donc la robustesse de l'algorithme d'adaptation par rapport aux bruits de mesures.

6.2. Vitesse de convergence

La vitesse de convergence de l'identification est principalement liée à la rapidité de la mise à jour des données. Après une variation importante d'un paramètre, les données sont complètement réactualisées au bout de $m + \max(nbry, nbbru)$ points, avec m le nombre de points dans la fenêtre d'observation et $nbry$ et $nbbru$ le nombre de régressions sur les vecteurs de sortie y et d'entrée u . Les simulations présentées ci-dessous permettent de vérifier que le temps d'adaptation est, conformément à nos attentes, court et que les divers problèmes numériques évoqués sont correctement résolus.

La robustesse des algorithmes est vérifiée en provoquant volontairement des variations exagérées des paramètres par rapport à des variations observées normalement sur des processus. Pour le modèle du premier ordre représenté par l'équation récurrente $y_{k+1} = ay_k + bu_k$, une variation du paramètre b entre 10^{-5} , 1 et 10^5 est simulée. La valeur de a est constante et fixée à 0.95. La loi de commande qui est utilisée pendant l'identification pour contrôler le système est présentée au chapitre IV.6.

Pour vérifier que les coefficients convergent vers les valeurs a et b réelles, les normes des erreurs

relatives $\frac{\|a_{réelle} - a\|_2}{\|a_{réelle}\|_2}$ et $\frac{\|b_{réelle} - b\|_2}{\|b_{réelle}\|_2}$ sont représentées sur la figure 2.8.

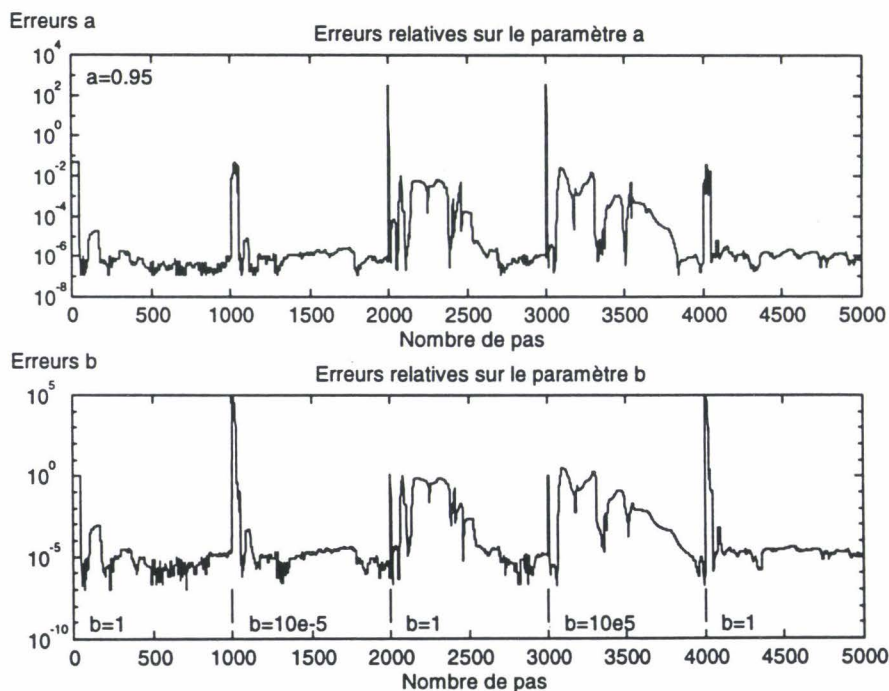


Figure 2.8 : Erreurs relatives sur les paramètres a et b lors de variations de b

Chapitre II : Adaptation des modèles

Après une forte augmentation ou diminution des valeurs de b , les erreurs sont très importantes, mais diminuent en quelques centaines de pas pour donner des erreurs relatives d'estimation des coefficients inférieures à 10^{-5} .

En fixant ensuite b à 1, une variation de a entre 0, 0.4, 0.8 puis 1.2 est appliquée. Une valeur de a supérieure à 1.2 peut être choisie sans altérer l'identification, mais une telle valeur signifie que le système diverge très rapidement entre deux instants d'échantillonnage. Ce cas est exclu ou peut être simplement résolu en prenant une période de régression plus petite. Les erreurs de modélisation lors de la variation du paramètre a sont présentées à la figure 2.9.

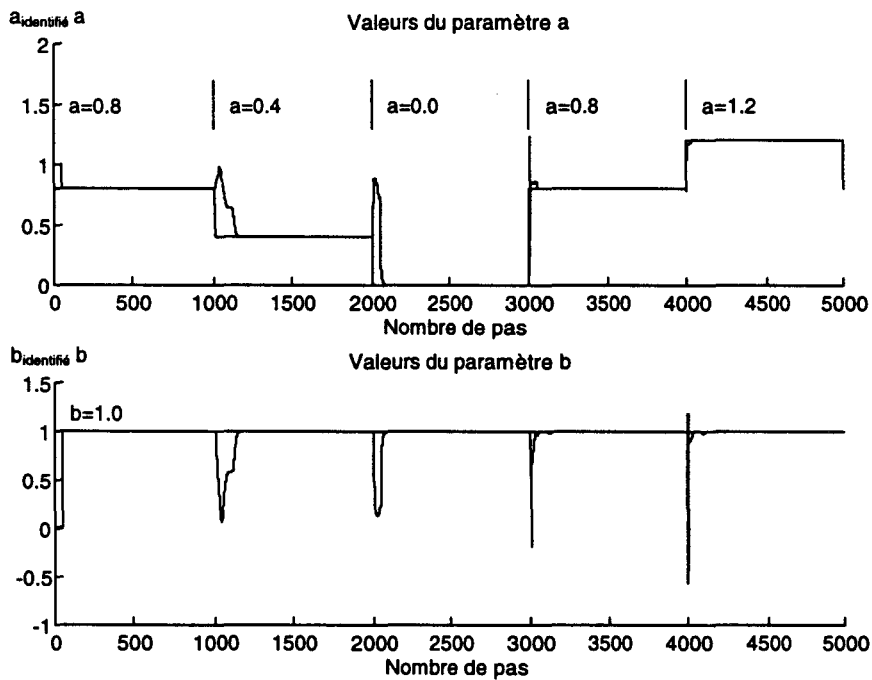


Figure 2.9 : Paramètres a et b lors de variations de a .

Ces simulations montrent que la loi d'adaptation développée est robuste et que la convergence des paramètres du modèle vers les valeurs optimales est rapide. D'autres essais sur des systèmes multivariables d'ordre élevé conduisent aux mêmes conclusions.

7. Conclusion

Après la présentation d'algorithmes d'adaptation obtenue par la rétro-propagation du gradient et à l'aide d'une fonction de Lyapunov, sont proposées des méthodes d'adaptation fondées sur le principe des moindres carrés. La robustesse de l'algorithme, le temps de convergence et le temps de calcul requis sont les principaux aspects étudiés.

La définition d'une fenêtre d'observation avec un nombre de points fixe, et un test sur le choix des mesures à partir desquels l'adaptation s'effectue, permettent alors de garantir des temps d'adaptation courts et de résoudre en partie les problèmes liés à l'absence d'excitation et à la présence de bruits sur les mesures. Lorsque l'amplitude de ces bruits est élevée, le filtrage proposé se révèle alors nécessaire pour aboutir à un modèle satisfaisant. Le choix du critère quadratique à minimiser et de l'algorithme de résolution est également primordial pour remédier aux problèmes numériques. Les méthodes et améliorations développées permettent alors d'aboutir à une loi d'adaptation des paramètres rapide et robuste, qui sont des propriétés indispensables pour déterminer ensuite la loi de commande. L'écriture récursive de l'algorithme est également présentée afin de réduire les temps de calcul et de faciliter l'implémentation de l'algorithme en temps réel.

Chapitre III :

Méthodes prédictives

1. Prédications de la consigne

Afin d'obtenir une loi de commande robuste et précise, l'approche développée repose sur le calcul des prédictions et sur l'optimisation de la commande pour minimiser les erreurs futures. Le fait d'anticiper sur l'évolution du système permet d'améliorer les performances du contrôleur, mais nécessite d'estimer non seulement les valeurs futures des sorties du système (Y_{fut}) mais aussi les valeurs futures de la consigne qui correspondent aux valeurs désirées des sorties ($Y_{d,fut}$).

Pour l'estimation des valeurs désirées futures, la solution la plus simple et la plus directe est de connaître a priori la consigne et donc les valeurs désirées des sorties. Ce cas peut se présenter lorsque la consigne est prédéfinie. En robotique par exemple, les opérations demandées sont souvent de nature cyclique. On impose alors continuellement la même trajectoire au robot. La notion de prédiction de la consigne est dans ce cas sans intérêt. La seconde façon de s'affranchir du calcul des prédictions est d'imposer au processus de suivre la trajectoire désirée avec h pas de retard, h étant le nombre de pas utilisé pour réaliser le calcul de la commande. Mais suivre une consigne avec un retard est une solution qui n'est pas entièrement satisfaisante.

1.1. Extrapolation de la consigne

Nous nous plaçons donc dans le cas le plus général où seule la consigne à l'instant présent et aux instants passés est connue. Les valeurs futures de la consigne sont donc à estimer. Ne disposant d'aucune autre information, la démarche à suivre est d'extrapoler les valeurs de consigne. Sachant que des valeurs erronées des prédictions engendrent des résultats finaux également erronés, une attention toute particulière doit être accordée à cet algorithme, même si les techniques employées sont élémentaires. Le principe consiste à estimer les dérivées première et seconde à l'instant présent pour déduire les valeurs futures.

L'extrapolation du premier ordre indique la tendance de la consigne à long terme, mais ne fournit pas une bonne estimation à court terme. La précision peut être améliorée en tenant compte des

termes du second ordre. Cependant l'allure de la trajectoire prédite devient parabolique, ce qui peut conduire à long terme à des valeurs très éloignées de la consigne et donc aberrantes. L'extrapolation avec un terme du troisième ordre aurait des effets similaires prédominants et n'est donc pas conseillée.

Pour combiner les avantages offerts par les termes du premier et du second ordre, le principe retenu est d'atténuer après chaque pas de prédiction la valeur de la dérivée seconde par un facteur β . Les prédictions à court terme sont alors précises, et à long terme nous évitons l'apparition de valeurs incohérentes qui seraient éventuellement en contradiction avec la consigne réellement désirée.

L'estimation numérique à l'instant k des variations du premier et du second ordre de la consigne, notées respectivement $\Delta Y_{d,k}$ et $\Delta_2 Y_{d,k}$ se déduit de la manière suivante :

$$\begin{aligned} \Delta Y_{d,k} &= Y_{d,k} - Y_{d,k-1} \\ \Delta_2 Y_{d,k} &= Y_{d,k} - 2Y_{d,k-1} + Y_{d,k-2} \end{aligned} \quad (3.1)$$

On suppose que la consigne n'est pas bruitée et que par conséquent les estimations de ces variations sont précises. L'extrapolation avec l'atténuation du terme du second ordre conduit à la suite récurrence suivante pour $i = 1$ à h :

$$\begin{bmatrix} Y_{d,k+i} \\ \Delta Y_{d,k+i} \\ \Delta_2 Y_{d,k+i} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & \beta \end{bmatrix} \begin{bmatrix} Y_{d,k+i-1} \\ \Delta Y_{d,k+i-1} \\ \Delta_2 Y_{d,k+i-1} \end{bmatrix} \quad (3.2)$$

que l'on peut réécrire sous la forme :

$$Y_{d,k+i} = \begin{bmatrix} 2+\beta & -1-2\beta & \beta \end{bmatrix} \begin{bmatrix} Y_{d,k+i-1} \\ Y_{d,k+i-2} \\ Y_{d,k+i-3} \end{bmatrix} \quad (3.3)$$

Les h prédictions de la consigne sont notées $Y_{d,fut}$, et se formulent également de la manière suivante :

$$Y_{d,fut} = \begin{bmatrix} Y_{d,k+h} \\ Y_{d,k+h-1} \\ \vdots \\ Y_{d,k+2} \\ Y_{d,k+1} \end{bmatrix} = \begin{bmatrix} 0 & 2+\beta & -1-2\beta & \beta & & & & \\ & 0 & 2+\beta & -1-2\beta & \beta & & & \\ & & & \ddots & \ddots & & & \\ & & & & 0 & 2+\beta & -1-2\beta & \beta \\ & & & & 0 & 2+\beta & -1-2\beta & \beta \end{bmatrix} \begin{bmatrix} Y_{d,k+h} \\ Y_{d,k+h-1} \\ \vdots \\ Y_{d,k+1} \\ Y_{d,k} \\ Y_{d,k-1} \\ Y_{d,k-2} \end{bmatrix} \quad (3.4)$$

Soient les matrices d'extrapolation $P_{1,ext}$ et $P_{2,ext}$:

$$P_{1,ext} = \begin{bmatrix} 0 & 2+\beta & -1-2\beta & \beta & & & \\ & \ddots & \ddots & \ddots & & & \\ & & 0 & 2+\beta & -1-2\beta & \beta & \\ & & & 0 & 2+\beta & -1-2\beta & \beta \\ & & & & 0 & 2+\beta & \\ & & & & & 0 & \\ & & & & & & 0 \end{bmatrix}, P_{2,ext} = \begin{bmatrix} \beta & & & & & & \\ -1-2\beta & \beta & & & & & \\ 2+\beta & -1-2\beta & \beta & & & & \end{bmatrix} \quad (3.5)$$

, le calcul de $Y_{d,fut}$ en fonction de $Y_{d,k}$, $Y_{d,k-1}$ et $Y_{d,k-2}$ s'écrit alors :

$$Y_{d,fut} = P_{ext} \begin{bmatrix} Y_{d,k} \\ Y_{d,k-1} \\ Y_{d,k-2} \end{bmatrix} \text{ avec } P_{ext} = (I - P_{1,ext})^{-1} P_{2,ext} \quad (3.6)$$

Cette dernière formulation n'est pas celle retenue pour les calculs des valeurs futures de la consigne, qui s'effectuent bien plus efficacement avec la suite récurrente (3.3). Cependant l'analyse des lois de commande et de la sensibilité par rapport à des variations de consigne se trouve facilitée en adoptant cette écriture.

Cette méthode et l'influence du coefficient β est illustrée par l'exemple ci-dessous qui montre les trajectoires prédites sans le terme du second ordre ($\beta=0$), avec la dérivée seconde non atténuée ($\beta=1$) et avec une atténuation ($\beta=0.6$).

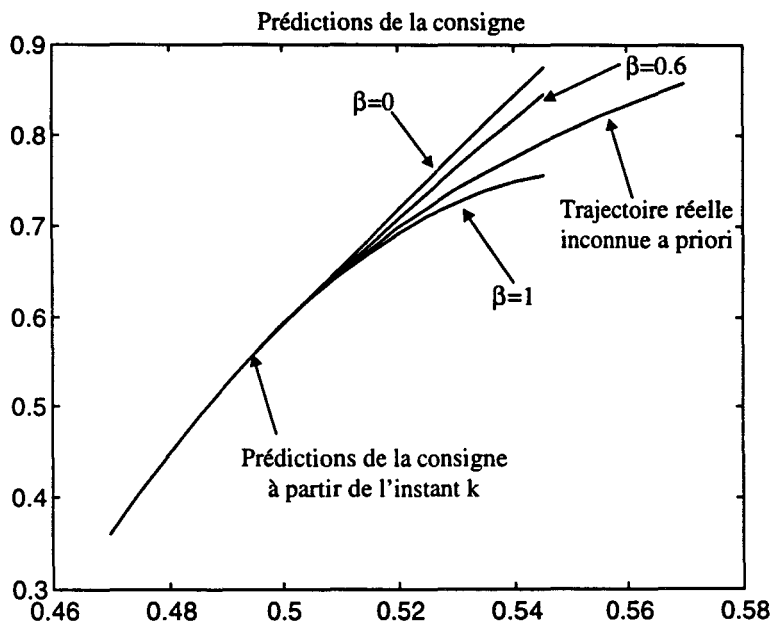


Figure 3.1 : Influence du coefficient β d'atténuation.

Le coefficient $\beta=0.6$ permet d'aboutir à un compromis satisfaisant entre les erreurs de prédiction à court terme et à long terme.

1.2. Détection de discontinuité

Nous devons également prendre en considération le cas où la consigne est définie par des échelons. Ce type de consigne est fréquemment rencontré et ne peut pas être ignoré. La méthode précédente est inadaptée car la discontinuité provoque une brusque augmentation des dérivées qui par extrapolation conduisent à des valeurs prédites croissantes ou décroissantes très importantes ne correspondant pas à l'échelon désiré. Pour remédier à ce problème un test sur la dérivée troisième permet de détecter la présence d'une variation brusque de la consigne. Le traitement adéquat présenté ci-dessus supprime les effets indésirables que peut provoquer une telle situation.

Ne connaissant pas a priori, l'ordre de grandeur des signaux de consigne et de leur variation, le test de discontinuité s'effectue par comparaison entre la somme des valeurs absolues de la dérivée troisième aux instants k et $k-1$, et la valeur filtrée de cette somme par un premier ordre du type passe-bas. L'estimation de la dérivée troisième et les deux valeurs sur lesquelles le test est basé se calculent de la manière suivante :

$$\Delta_3 Y_{d,k} = Y_{d,k} - 3Y_{d,k-1} + 3Y_{d,k-2} - Y_{d,k-3} \quad (3.7)$$

$$\Delta_{test1,k} = |\Delta_3 Y_{d,k}| + |\Delta_3 Y_{d,k-1}| \quad (3.8)$$

$$\Delta_{test2,k+1} = (1 - \alpha)\Delta_{test2,k} + \alpha\Delta_{test1,k} \text{ avec } \alpha \text{ le coefficient du filtre.} \quad (3.9)$$

Une discontinuité de la consigne ou toute variation brusque se traduit alors par une augmentation de la valeur $\Delta_{test1,k}$ qui devient supérieure à la valeur filtrée $\Delta_{test2,k}$. Une discontinuité est donc détectée si l'inégalité suivante est vérifiée :

$$\Delta_{test1,k} > \sigma\Delta_{test2,k} \quad \text{avec } \sigma \text{ le seuil de détection.}$$

Les paramètres α et σ sont à ajuster en fonction du type de consigne appliquée. Des résultats satisfaisants sont obtenus pour $\alpha = 0.1$ et $\sigma = 10$.

Quand ce test est vérifié, le calcul des prédictions de la consigne est réalisé en prenant comme estimation de la dérivée première, la dernière estimation notée $\Delta Y_{d,k0}$ pour laquelle le test est faux.

Les valeurs prédites se calculent alors de la manière suivante.

$$Y_{d,k+i} = Y_{d,k+i-1} + \Delta Y_{d,k0} \text{ pour } i = 1 \text{ à } h. \quad (3.10)$$

La méthode est illustrée ci-dessous par les exemples de trajectoires prédites sur 6 pas, d'une part dans le cas où les discontinuités ne sont pas détectées, et d'autre part avec l'algorithme de détection.

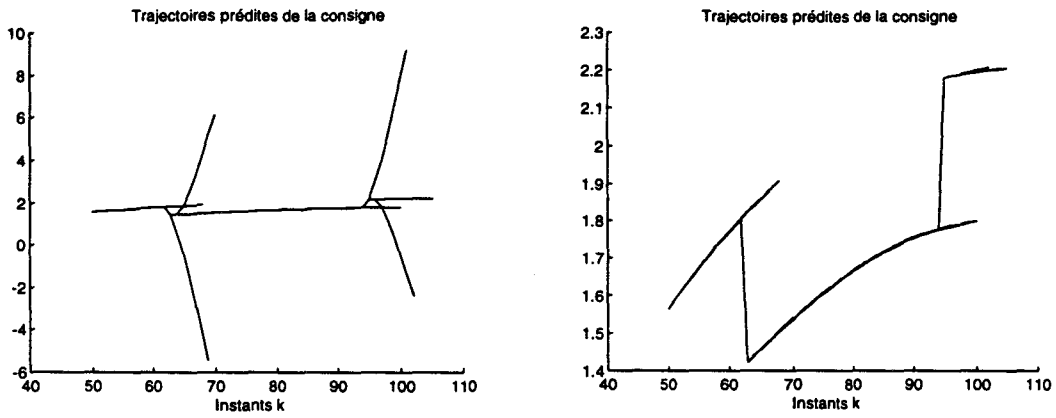


Figure 3.2 : Trajectoires prédites sans et avec détection des discontinuités

Lors d'un changement de consigne en échelon, le test permet d'aboutir à une trajectoire satisfaisante alors que l'algorithme sans test sur les discontinuités donne des valeurs prédites très éloignées des valeurs futures réelles et entraîne inéluctablement une dégradation des performances de la loi de commande.

2. Prédiction des sorties du système

Par définition, prédire est une opération délicate et hasardeuse où les erreurs sont inévitables. La première condition pour obtenir des résultats cohérents est d'avoir un modèle satisfaisant du processus dont on cherche à estimer les valeurs futures [BOUANI 96a]. Or les modèles définis et adaptés aux premier et deuxième chapitres ne donnent qu'une approximation linéaire du système valable localement autour du point d'identification seulement si les non-linéarités et perturbations ne sont pas trop importantes. Les prédictions ne peuvent donc se calculer avec une bonne précision que sur un horizon assez bref et dans un domaine limité dont les dimensions dépendent du domaine de validité du modèle.

2.1. Prédiction dans le repère global

Le modèle linéaire (1.48) s'écrit :

$$Y_{k+1} = J_x X_k + J_u U_k + v_{ope} \quad (3.11)$$

avec

$$v_{ope} = Y_{ope} - J_x X_{ope} - J_u U_{ope} \quad (3.12)$$

L'expression plus détaillée des matrices et vecteurs J_x , J_u , X_k et U_k permet de déduire le modèle suivant :

$$Y_{k+1} = \begin{bmatrix} J_{x,l} & \dots & J_{x,nbry} \end{bmatrix} \begin{bmatrix} Y_k \\ \vdots \\ Y_{k-nbry+1} \end{bmatrix} + \begin{bmatrix} J_{u,l} & \dots & J_{u,nbru} \end{bmatrix} \begin{bmatrix} U_k \\ \vdots \\ U_{k-nbru+1} \end{bmatrix} + v_{ope} \quad (3.13)$$

qui peut être développé sur h pas puis écrit sous forme matricielle :

$$\begin{bmatrix} Y_{k+h} \\ Y_{k+h-1} \\ \vdots \\ Y_{k+2} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & J_{x,1} & \cdots & J_{x,nbry} & & & \\ & 0 & J_{x,1} & \cdots & J_{x,nbry} & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 0 & J_{x,1} & \cdots & J_{x,nbry} \\ & & & & 0 & J_{x,1} & \cdots & J_{x,nbry} \end{bmatrix} \begin{bmatrix} Y_{k+h} \\ \vdots \\ Y_{k+2} \\ Y_{k+1} \\ \hline Y_k \\ \vdots \\ Y_{k-nbry+1} \end{bmatrix} + \begin{bmatrix} J_{u,1} & \cdots & J_{u,nbru} & & & \\ & J_{u,1} & \cdots & J_{u,nbru} & & \\ & & \ddots & \ddots & & \\ & & & J_{u,1} & \cdots & J_{u,nbru} \\ & & & & J_{u,1} & \cdots & J_{u,nbru} \end{bmatrix} \begin{bmatrix} u_{k+h-1} \\ \vdots \\ u_{k+2} \\ u_{k+1} \\ \hline u_k \\ \vdots \\ u_{k-nbru+1} \end{bmatrix} + \begin{bmatrix} v_{ope} \\ \vdots \\ v_{ope} \\ \vdots \\ v_{ope} \\ \vdots \\ v_{ope} \end{bmatrix} \quad (3.14)$$

Par souci de clarté, cette expression est présentée sous une forme matricielle plus compacte :

$$Y_{fut} = J_{x,fut} Y_{fut} + J_{u,fut} U_{fut} + J_{x,pas} Y_{pas} + J_{u,pas} U_{pas} + V_{ope} \quad (3.15)$$

avec

$$\begin{aligned}
 Y_{fut} &= \begin{bmatrix} Y_{k+h} \\ \vdots \\ Y_{k+1} \end{bmatrix}, U_{fut} = \begin{bmatrix} u_{k+h-1} \\ \vdots \\ u_{k+1} \end{bmatrix}, U_{pas} = \begin{bmatrix} u_k \\ \vdots \\ u_{k-nbru+1} \end{bmatrix}, Y_{pas} = \begin{bmatrix} Y_k \\ \vdots \\ Y_{k-nbry+1} \end{bmatrix}, V_{ope} = \begin{bmatrix} v_{ope} \\ \vdots \\ v_{ope} \end{bmatrix} \\
 J_{x,fut} &= \begin{bmatrix} 0 & J_{x,1} & \cdots & J_{x,nbry} & & \\ & 0 & J_{x,1} & \cdots & J_{x,nbry} & \\ & & \ddots & \ddots & \ddots & \\ & & & 0 & J_{x,1} & \cdots & J_{x,nbry} \\ & & & & 0 & J_{x,1} & \cdots & J_{x,nbry} \end{bmatrix}, J_{x,pas} = \begin{bmatrix} 0 & & & \\ J_{x,nbry} & 0 & & \\ \vdots & \ddots & & 0 \\ J_{x,1} & \cdots & J_{x,nbry} & \end{bmatrix} \\
 J_{u,fut} &= \begin{bmatrix} J_{u,1} & \cdots & J_{u,nbru} & 0 \\ & J_{u,1} & \cdots & J_{u,nbru} \\ & & \ddots & \ddots \\ & & & J_{u,1} \\ & & & & 0 \end{bmatrix}, J_{u,pas} = \begin{bmatrix} 0 & & & \\ J_{u,nbru} & 0 & & \\ \vdots & \ddots & & 0 \\ J_{u,1} & \cdots & J_{u,nbru} & \end{bmatrix}
 \end{aligned}$$

Les termes dépendant des mesures passées sont regroupés dans le vecteur V_{pas} , ce qui donne :

$$V_{pas} = J_{x,pas} Y_{pas} + J_{u,pas} U_{pas} + V_{ope} \quad (3.16)$$

Les valeurs prédites sont donc estimées de la manière suivante :

$$Y_{fut} = (I - J_{x,fut})^{-1} (J_{u,fut} U_{fut} + V_{pas}) \quad (3.17)$$

Nous posons pour simplifier l'écriture :

$$K_{x,fut} = (I - J_{x,fut})^{-1} \quad (3.18)$$

qui est une matrice triangulaire dont le calcul s'effectue de la manière suivante :

$$I - J_{x,fut} = \begin{bmatrix} I & -J_{x,1} & \cdots & -J_{x,nbry} \\ & I & -J_{x,1} & \cdots & -J_{x,nbry} \\ & & \ddots & \ddots & \ddots \\ & & & I & -J_{x,1} \\ & & & & I \end{bmatrix}, K_{x,fut} = \begin{bmatrix} I & K_1 & K_2 & \cdots & K_n \\ & I & K_1 & \cdots & K_{n-1} \\ & & \ddots & & \vdots \\ & & & I & K_1 \\ & & & & I \end{bmatrix} \quad (3.19)$$

avec

$$K_1 = J_{x,1} \quad (3.20)$$

$$K_i = \sum_{k=1}^{i-1} K_{i-k} J_{x,k} + J_{x,i} \text{ pour } i \leq nbry, \quad (3.21)$$

$$K_i = \sum_{k=1}^{nbry} K_{i-k} J_{x,k} \text{ pour } i > nbry, \quad (3.22)$$

Les valeurs prédites utilisées ensuite pour déterminer la loi de commande s'écrivent :

$$Y_{fut} = K_{x,fut} J_{u,fut} U_{fut} + K_{x,fut} V_{pas} \quad (3.23)$$

2.2. Prédiction dans le repère local

Lorsque le point de fonctionnement est éloigné de l'origine le terme V_{ope} prend des valeurs importantes et devient sensible aux erreurs d'estimation des matrices J_x et J_u . Par conséquent, les calculs qui en découlent sont aussi sensibles à ces erreurs. La solution consiste donc à ne plus utiliser le modèle dans le repère global afin d'éviter le calcul de V_{ope} . La réécriture dans un repère local et la présentation différente des calculs permet alors de réduire les erreurs d'estimation. Nous profitons également de cette transformation pour introduire dans le calcul des prédictions les termes d'identification de la zone inactive présentée dans le chapitre I.6.

Le modèle est de la forme :

$$Y_{k+1} - Y_{ope} = J_x (X_k - X_{ope}) + J_u (U_k - U_{ope}) + J_{uf} (SU_k - SU_{ope}) \quad (3.24)$$

ou encore

$$Y_{k+1} - Y_{ope} = \begin{bmatrix} J_{x,1} & \cdots & J_{x,nbry} \end{bmatrix} \begin{bmatrix} Y_k - X_{ope,1} \\ \vdots \\ Y_{k-nbry+1} - X_{ope,nbry} \end{bmatrix} + \begin{bmatrix} J_{u,1} & \cdots & J_{u,nbru} \end{bmatrix} \begin{bmatrix} u_k - u_{ope,1} \\ \vdots \\ u_{k-nbru+1} - u_{ope,nbru} \end{bmatrix} + \begin{bmatrix} J_{uf,1} & \cdots & J_{uf,nbru} \end{bmatrix} \begin{bmatrix} SU_k - SU_{ope,1} \\ \vdots \\ SU_{k-nbru+1} - SU_{ope,nbru} \end{bmatrix} \quad (3.25)$$

Au lieu d'écrire ce modèle autour du point de fonctionnement $(Y_{ope}, X_{ope}, U_{ope}, SU_{ope})$, ce modèle peut s'écrire autour du point $(Y_{ope}, [Y_{ope} \ Y_{ope} \dots \ Y_{ope}]^T, U_k, SU_k)$ qui devient alors l'origine du repère local.

$$\begin{aligned}
 Y_{k+1} - Y_{ope} &= \begin{bmatrix} J_{x,1} & \dots & J_{x,nbry} \end{bmatrix} \begin{bmatrix} Y_k - Y_{ope} \\ \vdots \\ Y_{k-nbry+1} - Y_{ope} \end{bmatrix} + \begin{bmatrix} J_{x,1} & \dots & J_{x,nbry} \end{bmatrix} \begin{bmatrix} Y_{ope} - X_{ope,1} \\ \vdots \\ Y_{ope} - X_{ope,nbry} \end{bmatrix} \\
 &+ \begin{bmatrix} J_{u,1} & \dots & J_{u,nbru} \end{bmatrix} \begin{bmatrix} u_k - u_k \\ \vdots \\ u_{k-nbru+1} - u_k \end{bmatrix} + \begin{bmatrix} J_{u,1} & \dots & J_{u,nbru} \end{bmatrix} \begin{bmatrix} u_k - u_{ope,1} \\ \vdots \\ u_k - u_{ope,nbru} \end{bmatrix} \\
 &+ \begin{bmatrix} J_{uf,1} & \dots & J_{uf,nbru} \end{bmatrix} \begin{bmatrix} su_k - su_k \\ \vdots \\ su_{k-nbru+1} - su_k \end{bmatrix} + \begin{bmatrix} J_{uf,1} & \dots & J_{uf,nbru} \end{bmatrix} \begin{bmatrix} su_k - su_{ope,1} \\ \vdots \\ su_k - su_{ope,nbru} \end{bmatrix}
 \end{aligned} \tag{3.26}$$

On définit un nouveau vecteur Δv_{ope} déduit à partir de l'origine du repère local.

$$\begin{aligned}
 \Delta v_{ope} &= \begin{bmatrix} J_{x,1} & \dots & J_{x,nbry} \end{bmatrix} \begin{bmatrix} Y_{ope} - X_{ope,1} \\ \vdots \\ Y_{ope} - X_{ope,nbry} \end{bmatrix} \\
 &+ \begin{bmatrix} J_{u,1} & \dots & J_{u,nbru} \end{bmatrix} \begin{bmatrix} u_k - u_{ope,1} \\ \vdots \\ u_k - u_{ope,nbru} \end{bmatrix} + \begin{bmatrix} J_{uf,1} & \dots & J_{uf,nbru} \end{bmatrix} \begin{bmatrix} su_k - su_{ope,1} \\ \vdots \\ su_k - su_{ope,nbru} \end{bmatrix}
 \end{aligned} \tag{3.27}$$

La relation précédente écrite sur h instants permet de déduire les équations suivantes :

$$\begin{aligned}
 \begin{bmatrix} Y_{k+h} - Y_{ope} \\ Y_{k+h-1} - Y_{ope} \\ \vdots \\ Y_{k+2} - Y_{ope} \\ Y_{k+1} - Y_{ope} \end{bmatrix} &= \begin{bmatrix} 0 & J_{x,1} & \dots & J_{x,nbry} & & & \\ & 0 & J_{x,1} & \dots & J_{x,nbry} & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 0 & J_{x,1} & \dots & J_{x,nbry} \\ & & & & 0 & J_{x,1} & \dots & J_{x,nbry} \end{bmatrix} \begin{bmatrix} Y_{k+h} - Y_{ope} \\ \vdots \\ \vdots \\ Y_{k+2} - Y_{ope} \\ Y_{k+1} - Y_{ope} \\ \hline Y_k - Y_{ope} \\ \vdots \\ Y_{k-nbry+1} - Y_{ope} \end{bmatrix} \\
 &+ \begin{bmatrix} J_{u,1} & \dots & J_{u,nbru} & & & & \\ & J_{u,1} & \dots & J_{u,nbru} & & & \\ & & \ddots & \ddots & & & \\ & & & J_{u,1} & \dots & J_{u,nbru} & \\ & & & & J_{u,1} & \dots & J_{u,nbru} \end{bmatrix} \begin{bmatrix} u_{k+h-1} - u_k \\ \vdots \\ \vdots \\ u_{k+2} - u_k \\ u_{k+1} - u_k \\ \hline u_k - u_k \\ \vdots \\ u_{k-nbru+1} - u_k \end{bmatrix} \\
 &+ \begin{bmatrix} J_{uf,1} & \dots & J_{uf,nbru} & & & & \\ & J_{uf,1} & \dots & J_{uf,nbru} & & & \\ & & \ddots & \ddots & & & \\ & & & J_{uf,1} & \dots & J_{uf,nbru} & \\ & & & & J_{uf,1} & \dots & J_{uf,nbru} \end{bmatrix} \begin{bmatrix} su_{k+h-1} - su_k \\ \vdots \\ \vdots \\ su_{k+2} - su_k \\ su_{k+1} - su_k \\ \hline su_k - su_k \\ \vdots \\ su_{k-nbru+1} - su_k \end{bmatrix} + \begin{bmatrix} \Delta v_{ope} \\ \Delta v_{ope} \\ \vdots \\ \Delta v_{ope} \\ \Delta v_{ope} \end{bmatrix}
 \end{aligned} \tag{3.28}$$

L'écriture de ce système peut se simplifier en définissant les matrices $J_{x,fut}$, $J_{u,fut}$, $J_{uf,fut}$, $J_{x,pas}$, $J_{u,pas}$ et $J_{uf,pas}$ portant sur les valeurs futures et passées des sorties et des commandes. On aboutit alors à l'équation suivante :

$$\begin{aligned} \Delta Y_{fut} &= J_{x,fut} \Delta Y_{fut} + J_{u,fut} \Delta U_{fut} + J_{uf,fut} \Delta SU_{fut} \\ &+ J_{x,pas} \Delta Y_{pas} + J_{u,pas} \Delta U_{pas} + J_{uf,pas} \Delta SU_{pas} + \Delta V_{ope} \end{aligned} \quad (3.29)$$

En posant :

$$\Delta V_{pas} = J_{x,pas} \Delta Y_{pas} + J_{u,pas} \Delta U_{pas} + J_{uf,pas} \Delta SU_{pas} + \Delta V_{ope} \quad (3.30)$$

l'expression (3.29) devient :

$$(I - J_{x,fut}) \Delta Y_{fut} = J_{u,fut} \Delta U_{fut} + J_{uf,fut} \Delta SU_{fut} + \Delta V_{pas} \quad (3.31)$$

ou encore

$$\Delta Y_{fut} = K_{x,fut} (J_{u,fut} \Delta U_{fut} + J_{uf,fut} \Delta SU_{fut} + \Delta V_{pas}) \quad (3.32)$$

Le signe de la commande est considéré comme constant car la prise en compte d'un changement de signe a lieu directement lors du calcul de la compensation de la zone inactive. Les valeurs prédites dans le repère local sont donc égales à :

$$\Delta Y_{fut} = K_{x,fut} J_{u,fut} \Delta U_{fut} + K_{x,fut} \Delta V_{pas} \quad (3.33)$$

L'introduction des zones inactives dans le modèle ne modifie alors que légèrement le calcul des prédictions. Seul le calcul du vecteur ΔV_{pas} subit un changement dû à l'ajout de nouveaux termes dépendant de la matrice J_{uf} .

2.3. Sensibilité du calcul des prédictions

Le calcul des prédictions est sensible aux erreurs de modélisation à cause en particulier de la matrice $K_{x,fut}$ qui s'obtient en inversant la matrice $I - J_{x,fut}$. La moindre erreur de modélisation ou d'arrondi est en effet réintroduite et amplifiée pour le calcul suivant. Par exemple pour le système du premier ordre $y_{k+1} = ay_k + bu_k$, les matrices $J_{x,fut}$, $J_{u,fut}$ et $K_{x,fut}$ pour h prédictions se calculent de la manière suivante :

$$J_{x,fut} = \begin{bmatrix} 0 & a & & & \\ & \ddots & \ddots & & \\ & & 0 & a & \\ & & & 0 & a \\ & & & & 0 \end{bmatrix} \quad J_{u,fut} = \begin{bmatrix} b & & & & \\ & \ddots & & & \\ & & b & & \\ & & & b & \\ 0 & \dots & & & 0 \end{bmatrix} \quad K_{x,fut} = \begin{bmatrix} I & a & a^2 & \dots & a^{h-1} \\ & \ddots & \ddots & \ddots & \vdots \\ & & I & a & a^2 \\ & & & I & a \\ & & & & I \end{bmatrix} \quad (3.34)$$

En prenant les conditions initiales du système égales à y_k et u_k , et un point de fonctionnement en y_{ope} , x_{ope} et u_{ope} , l'application des équations (3.16) et (3.23) pour ce système donne la valeur prédite y_{k+h} suivante :

$$y_{k+h} = b \sum_{i=0}^{h-2} (a^i u_{k+h-i-1}) + a^h y_k + a^{h-1} b u_k + (y_{ope} - a x_{ope} - b u_{ope}) \sum_{i=0}^{h-1} a^i \quad (3.35)$$

L'influence sur y_{k+h} des paramètres a et b , des commandes futures, des valeurs passées et du point de fonctionnement apparaît clairement. Le calcul des dérivées de y_{k+h} par rapport à a et b montre alors qu'une modification de ces paramètres influe fortement sur le calcul des valeurs prédites et met en évidence les difficultés pour calculer les prédictions du système. En effet le coefficient a élevé à la puissance h ou $h-1$ intervient dans le calcul de y_{k+h} et de ses dérivées par rapport à a et b :

$$\begin{aligned} \frac{\partial y_{k+h}}{\partial a} &= b \sum_{i=1}^{h-2} (i a^{i-1} u_{k+h-i-1}) + h a^{h-1} y_k + (h-1) a^{h-2} b u_k + (y_{ope} - b u_{ope}) \sum_{i=1}^{h-1} (i a^{i-1}) - x_{ope} \sum_{i=0}^{h-1} (i+1) a^i \\ \frac{\partial y_{k+h}}{\partial b} &= \sum_{i=0}^{h-2} (a^i u_{k+h-i-1}) + a^{h-1} u_k - u_{ope} \sum_{i=0}^{h-1} a^i \end{aligned}$$

Pour cette raison, l'estimation des valeurs futures à très long terme est peu précise. Par conséquent un horizon de prédiction lointain est déconseillé. Les erreurs ou imprécisions sur les paramètres du modèle et la très grande sensibilité de la matrice $K_{x,fut}$ sont donc les principaux inconvénients des méthodes prédictives.

3. Calcul des erreurs prédites

Les erreurs prédites sont égales à la différence entre la trajectoire désirée prédite et la trajectoire future du système. En se basant sur les résultats obtenus précédemment, le vecteur des erreurs futures e_{fut} dans le repère global vaut :

$$e_{fut} = Y_{d,fut} - Y_{fut} \quad , \quad (3.36)$$

ou encore après substitution de Y_{fut} par l'équation (3.37) :

$$e_{fut} = Y_{d,fut} - K_{x,fut} V_{pas} - K_{x,fut} J_{u,fut} U_{fut} \quad (3.38)$$

De même, les erreurs futures dans le repère local valent :

$$e_{fut} = \Delta Y_{d,fut} - \Delta Y_{fut} \quad , \quad (3.39)$$

d'où

$$e_{fut} = \Delta Y_{d,fut} - K_{x,fut} \Delta V_{pas} - K_{x,fut} J_{u,fut} \Delta U_{fut} \quad (3.40)$$

4. Prédiction avec des commandes en échelon ou en rampe

Pour une prédiction sur h pas, l'ensemble des commandes aux instants $k+1$ à $k+h-1$ influent sur les valeurs des sorties du système. Une simplification des calculs peut s'effectuer en imposant aux valeurs futures de la commande de suivre un échelon ou une rampe. Le premier avantage porte sur la détermination de la loi de commande où seules les variations de la commande notées δu par rapport à la commande u_k sont à calculer et à optimiser. Le second avantage porte sur l'allure de la commande qui est plus lisse et plus régulière puisque les prédictions et l'optimisation s'effectuent en considérant que les commandes futures sont constantes ou ont une pente constante.

4.1. Contrainte sur l'allure de la commande

Les commandes aux instants $k+1$ à $k+h-1$ sont supposées évoluer en suivant un échelon ou une rampe, ce qui se traduit par les relations suivantes :

$$U_{fut} = \begin{bmatrix} u_k + \delta u \\ \vdots \\ u_k + \delta u \\ u_k + \delta u \end{bmatrix} = \begin{bmatrix} u_k \\ \vdots \\ u_k \\ u_k \end{bmatrix} + \begin{bmatrix} I \\ \vdots \\ I \\ I \end{bmatrix} \delta u \text{ pour un échelon,} \quad (3.41)$$

et

$$U_{fut} = \begin{bmatrix} u_k + (h-1)\delta u \\ \vdots \\ u_k + 2\delta u \\ u_k + \delta u \end{bmatrix} = \begin{bmatrix} u_k \\ \vdots \\ u_k \\ u_k \end{bmatrix} + \begin{bmatrix} (h-1)I \\ \vdots \\ 2I \\ I \end{bmatrix} \delta u \text{ pour une rampe.} \quad (3.42)$$

En posant :

$$H_{fut,0} = \begin{bmatrix} I \\ \vdots \\ I \\ I \end{bmatrix} \text{ pour un échelon et } H_{fut,1} = \begin{bmatrix} (h-1)I \\ \vdots \\ 2I \\ I \end{bmatrix} \text{ pour une rampe}$$

alors le vecteur U_{fut} s'écrit :

$$U_{fut} = \begin{bmatrix} u_k \\ \vdots \\ u_k \end{bmatrix} + H_{fut} \delta u \text{ avec } H_{fut} = H_{fut,0} \text{ ou } H_{fut,1}. \quad (3.43)$$

Il est intéressant de considérer le cas où les commandes futures U_{fut} peuvent évoluer suivant un échelon ou une rampe, ou une combinaison des deux en affectant un coefficient de pondération entre $H_{fut,0}$ et $H_{fut,1}$. L'influence de ce choix est étudiée dans le chapitre consacré à l'élaboration de la loi de commande. Ce choix modifie les caractéristiques du contrôleur, et peut contribuer à améliorer la stabilité, la robustesse de la commande, ou la précision du suivi de trajectoire.

4.2. Formulation dans le repère global

Dans le cas où les prédictions s'effectuent dans le repère global, le calcul des estimations des sorties du système en fonction de δu devient :

$$Y_{fut} = K_{x,fut} J_{u,fut} H_{fut} \delta u + K_{x,fut} V_{pas} + K_{x,fut} J_{u,fut} \begin{bmatrix} u_k \\ \vdots \\ u_k \end{bmatrix} \quad (3.44)$$

4.3. Formulation dans le repère local

Lorsque les calculs s'effectuent autour de l'origine du repère local (Y_{ope} , $[Y_{ope} \ Y_{ope} \dots \ Y_{ope}]^T$, U_k), les matrices H_{fut} sont identiques, mais l'expression des valeurs futures de la commande est simplifiée :

$$\Delta U_{fut} = H_{fut} \delta u \quad (3.45)$$

d'où

$$\Delta Y_{fut} = K_{x,fut} J_{u,fut} H_{fut} \delta u + K_{x,fut} \Delta V_{pas} \quad (3.46)$$

Avec la matrice H_{fut} égale à $H_{fut,0}$ ou $H_{fut,1}$, le terme $J_{u,fut} H_{fut}$ se calcule aisément par récurrence.

Une allure différente de la commande H_{fut} peut également être envisagée, mais conduit à une expression de la matrice $J_{u,fut} H_{fut}$ plus longue à calculer, car il n'est plus possible de définir des suites récurrentes pour obtenir le résultat final.

5. Conclusion

L'estimation du comportement futur du système en fonction du vecteur de commande U_{fut} ou ΔU_{fut} est essentielle pour établir la loi de commande, car le rôle du contrôleur est de minimiser les erreurs en anticipant sur l'évolution de la consigne et du processus. Cependant l'étude menée au cours de ce chapitre montre que le calcul des valeurs futures du système est sensible aux erreurs de modélisation, et nous indique que les estimations à long terme sont peu précises. La loi de commande qui émane du calcul des valeurs prédites doit donc être déterminée en limitant l'horizon de prédiction et à l'aide de critères établis pour réduire la sensibilité de la commande aux erreurs de modélisation.

Chapitre IV :

Détermination de la loi de commande

1. Introduction

La loi de commande a premièrement pour rôle de contraindre les sorties du système à converger vers les valeurs désirées. Deuxièmement, elle doit être établie de manière à obtenir des valeurs de commande les plus régulières possibles. Cette seconde contrainte est prépondérante quand il s'agit de systèmes mécaniques pour lesquels il faut limiter les sollicitations sur les actionneurs, réduire la fatigue des composants et minimiser la consommation d'énergie. Les perturbations agissant sur le processus doivent être contrebalancées et la présence inévitable de bruit sur les mesures ne doit pas entraver le bon fonctionnement du contrôleur.

Pour un système monovisible, une première possibilité pour calculer la commande est d'inverser le modèle pour que le terme u_k apparaisse explicitement en fonction des sorties et des commandes aux instants précédents. En substituant alors à la valeur de sortie y_{k+1} la valeur désirée $y_{d,k+1}$, on aboutit à une loi de commande simple, mais peu robuste et très sensible aux bruits et aux erreurs de modélisation, et donc totalement inefficace. Cette approche ne tient pas compte des valeurs ultérieures de la consigne, et néglige l'influence de la commande appliquée à un instant donné sur l'état futur du processus. Pour ces raisons, des critères doivent être étudiés spécialement afin d'aboutir à une loi de commande satisfaisante, assurant un bon compromis entre le temps de réponse, la précision, la stabilité et la sensibilité aux bruits et aux perturbations [ORT 89].

Quatre critères quadratiques visant à déterminer une loi de commande répondant aux objectifs fixés, ainsi que des variantes de ces critères reflétant diverses approches ont été développés. Les lois de commande qui en découlent sont testées et validées en simulation sur des exemples illustrant les caractéristiques, les avantages et les limites de chacun des critères. La validation expérimentale de ces lois de commande est présentée au cinquième chapitre.

2. Principes et règles de base de la commande

2.1. Détermination de la consigne et du modèle de référence

La consigne doit être définie de manière à ce que les sorties désirées et les variations des sorties désirées soient acceptables par rapport à la dynamique et aux caractéristiques du processus. Dans le

cas de systèmes mécaniques tels que les robots, les changements de consigne et les déplacements imposés doivent être filtrés ou définis de sorte que la trajectoire désirée reste cohérente avec la trajectoire accessible avec des commandes physiquement réalisables. Cette première condition est impérative pour aboutir à une commande satisfaisante. La méthode la plus classique est de définir un modèle de référence qui filtre et adoucit les transitions lors des changements de consigne. On considère alors pour la suite des développements et des simulations que la consigne imposée et le modèle de référence utilisé sont choisis correctement par rapport au processus.

2.2. Dynamique d'annulation des erreurs

Les erreurs entre la trajectoire désirée et la trajectoire obtenue doivent être annulées avec une dynamique rapide pour obtenir une bonne précision. Cependant la sensibilité de la commande par rapport aux bruits et aux perturbations est réduite si on choisit une dynamique plus lente. La stabilité de la commande et sa robustesse par rapport aux erreurs de modélisation dépend également de la dynamique d'annulation des erreurs que l'on impose. Les méthodes employées pour définir comment le système doit suivre ou rejoindre la consigne sont donc à établir avec une attention toute particulière. La première méthode consiste à lisser la trajectoire désirée. Les secondes méthodes étudiées sont basées sur la définition du critère quadratique à minimiser. L'utilisation conjointe des deux approches est également possible.

Dans le premier cas, la dynamique d'annulation des erreurs est définie telle que le vecteur de sortie du processus Y_k rejoigne la trajectoire désirée $Y_{d,fut}$ calculée au chapitre III, en obéissant aux instants $k+1$ à $k+h$ à la loi suivante :

$$Y_{k+1} = (1 - \alpha)Y_k + \alpha Y_{d,k+1} \quad (4.1)$$

On définit alors une trajectoire désirée filtrée et lissée notée $Y_{dl,fut}$ entre la sortie du processus à l'instant k et la trajectoire désirée $Y_{d,fut}$. Le premier terme $Y_{dl,k+1}$, et les termes suivants $Y_{dl,k+2}$ à $Y_{dl,k+h}$ se déduisent des équations suivantes :

$$Y_{dl,k+1} = (1 - \alpha)Y_k + \alpha Y_{d,k+1} \quad (4.2)$$

$$Y_{dl,k+i} = (1 - \alpha)Y_{dl,k+i-1} + \alpha Y_{d,k+i} \text{ pour } i=2 \text{ à } h. \quad (4.3)$$

Le coefficient α détermine le taux de décroissance souhaitée des erreurs prédites qui se déduisent alors de la manière suivante :

$$e_{fut} = Y_{dl,fut} - Y_{fut} \quad (4.4)$$

La décroissance des erreurs est également fixée lors de la définition du critère qui est un élément clé de la loi de commande. Pour régler la dynamique d'annulation des erreurs, il est possible d'ajouter

des contraintes sur la commande, ou sur les variations de la commande afin de limiter leur amplitude (Chapitre IV.3). L'intérêt d'une commande future choisie en échelon ou en rampe a été évoqué au troisième chapitre. Cette contrainte sur la commande est alors étudiée et intégrée dans les divers critères définis ci-après.

Le réglage de la dynamique d'annulation des erreurs peut aussi s'obtenir en insérant une contrainte limitant les variations des erreurs (Chapitre IV.4) ou imposant l'allure de la décroissance des erreurs (Chapitre IV.5). En présence de bruit important sur les mesures, d'autres types de contraintes filtrant les erreurs sont à ajouter dans le critère pour limiter l'influence du bruit (Chapitre IV.6).

2.3. Erreur statique et détermination du point de fonctionnement

Le point de fonctionnement est le point d'origine du repère local, autour duquel est écrit le modèle du processus. Afin d'annuler les erreurs statiques survenant à cause des éventuelles erreurs de modélisation, ce point doit rejoindre en régime permanent la valeur des sorties du processus et évidemment de la consigne. Le point de fonctionnement pour le calcul de la commande doit donc être différent du point de fonctionnement utilisé pour l'identification, car ce dernier n'est pas constamment mis à jour pour éviter les problèmes de matrices singulières lors de l'adaptation du modèle.

Pour arriver à une erreur nulle en régime permanent, le point de fonctionnement se détermine soit en effectuant la moyenne des m derniers points de mesures dont la mise à jour est alors effectuée en permanence, soit à l'aide d'un filtre du premier ordre avec un facteur d'oubli défini pour accorder de l'importance principalement aux mesures récentes.

La première méthode donne alors un point de fonctionnement égal à la moyenne de m points, qui est calculée conformément à l'algorithme récursif défini au chapitre II avec par exemple le nombre de points utilisés égal à celui défini pour l'identification. Un nombre restreint de points peut également être choisi mais le bruit est moins filtré et a alors plus d'influence sur le calcul final de la commande.

La seconde méthode qui consiste à appliquer un filtre passe-bas sur les mesures pour déduire la position du point de fonctionnement ($Y_{ope}, X_{ope}, U_{ope}$) est plus simple à mettre en oeuvre :

$$\begin{aligned} Y_{ope,k} &= (1 - \alpha)Y_{ope,k-1} + \alpha Y_k \\ X_{ope,k} &= (1 - \alpha)X_{ope,k-1} + \alpha X_k \\ U_{ope,k} &= (1 - \alpha)U_{ope,k-1} + \alpha U_k \end{aligned} \tag{4.5}$$

Un coefficient α égal à 0.1 filtre les bruits suffisamment et permet d'obtenir une convergence rapide du point de fonctionnement vers les sorties du processus, ce qui garantit une erreur statique nulle.

3. Critère avec limitations des dérivées successives des commandes

3.1. Définition du critère

Ce critère est une traduction directe des spécifications formulées à propos de la loi de commande que l'on souhaite telle que le processus soit contrôlé avec une bonne précision et avec de faibles sollicitations sur les actionneurs. Deux termes apparaissent donc dans le critère, l'un sur les commandes, l'autre sur les erreurs entre la trajectoire désirée lissée et la trajectoire du processus. La pondération entre les deux termes permet d'aboutir au compromis désiré [CAN 97a].

Le premier terme du critère doit donc minimiser les erreurs entre la trajectoire désirée lissée $Y_{dl,fut}$ et les valeurs prédites du processus Y_{fut} dont le calcul a été établi au chapitre précédent. Les erreurs futures e_{fut} s'expriment donc de la manière suivante :

$$e_{fut} = Y_{dl,fut} - Y_{fut} \quad (4.6)$$

En considérant le cas où le modèle est défini dans le repère global, les erreurs prédites s'écrivent :

$$e_{fut} = Y_{dl,fut} - K_{x,fut} (J_{u,fut} U_{fut} + V_{pas}) \quad (4.7)$$

D'autre part on souhaite avoir les dérivées successives des commandes les plus petites possibles, afin de réduire dans le cas de systèmes mécaniques les sollicitations sur les actionneurs et d'éviter le plus possible les secousses et les vibrations créées par une commande trop irrégulière. Pour les seconds termes intervenant dans le critère, il est donc nécessaire d'estimer la dérivée $i^{\text{ème}}$ des commandes notée $\Delta_i(u_k)$ qui s'obtient à partir de la relation :

$$\Delta_i(u_k) = \frac{1}{c_i} (1 - q^{-1})^i (u_k) \quad (4.8)$$

avec q^{-1} l'opérateur de retard, et c_i un coefficient dépendant de la période d'échantillonnage.

Les $i+1$ coefficients du polynôme en q^{-1} obtenu sont notés $c_{i,j}$ avec $j=0$ à i . Ces coefficients se calculent directement en construisant le triangle de Pascal jusqu'à l'ordre désiré et en affectant un signe négatif pour les indices j impairs. L'estimation de la dérivée $i^{\text{ème}}$ d'une commande s'écrit alors :

$$\Delta_i(u_k) = \frac{1}{c_i} (c_{i,0}u_k + c_{i,1}u_{k-1} + c_{i,2}u_{k-2} + \dots + c_{i,i-1}u_{k-i+1} + c_{i,i}u_{k-i}) \quad (4.9)$$

Le calcul des variations Δ_i pour toutes les commandes futures aux instants $k+1$ à $k+h-1$, avec h le nombre de prédictions, donne une expression matricielle de la forme :

$$\begin{bmatrix} \Delta_i(u_{k+h-1}) \\ \Delta_i(u_{k+h-2}) \\ \vdots \\ \Delta_i(u_{k+2}) \\ \Delta_i(u_{k+1}) \end{bmatrix} = \frac{1}{c_i} \begin{bmatrix} c_{i,0}I & c_{i,1}I & \cdots & c_{i,i}I & & & \\ & c_{i,0}I & c_{i,1}I & \cdots & c_{i,i}I & & \\ & & \ddots & \ddots & & & \\ & & & c_{i,0}I & c_{i,1}I & \cdots & c_{i,i}I \\ & & & & c_{i,0}I & c_{i,1}I & \cdots & c_{i,i}I \\ & & & & & \ddots & & \\ & & & & & & c_{i,1}I & \cdots & c_{i,i}I \end{bmatrix} \begin{bmatrix} u_{k+h-1} \\ u_{k+h-2} \\ \vdots \\ u_{k+2} \\ \hline u_{k+1} \\ u_k \\ \vdots \\ u_{k-i+1} \end{bmatrix} \quad (4.10)$$

que l'on peut réécrire en séparant les termes futurs (aux instants $k+1$ à $k+h-1$) et les termes passés (aux instants $k-i+1$ à k). On obtient alors l'équation suivante :

$$\Delta_{i,U_{fut}} = S_{i,fut} U_{fut} + S_{i,pas} U_{i,pas} \quad (4.11)$$

avec $S_{i,fut}$ et $S_{i,pas}$ les matrices des coefficients utilisés pour estimer les dérivées $i^{ème}$ des commandes futures.

En définissant s l'indice maximal des dérivées successives à minimiser, le critère H permettant d'aboutir à la loi de commande s'écrit :

$$H = \frac{1}{2} \left(e_{fut}^T e_{fut} + \sum_{i=1}^s \Delta_{i,U_{fut}}^T Q_i \Delta_{i,U_{fut}} \right) \quad (4.12)$$

avec Q_i les matrices pondérant les estimations des dérivées de la commande.

3.2. Réglage des coefficients de pondération

Par souci de simplicité, les coefficients c_i dépendant de la période d'échantillonnage peuvent être inclus dans les matrices de pondération Q_i . De plus chacune de ces matrices est choisie diagonale avec tous les termes identiques. Leur réglage reste néanmoins délicat et une valeur mal choisie peut rendre la loi de commande oscillante ou instable.

Ce critère peut en effet avoir des effets néfastes indésirables puisque la minimisation de la dérivée $i^{ème}$ signifie que la valeur de la dérivée $i-1$ doit être constante. Par conséquent, imposer une dérivée troisième nulle revient à imposer une dérivée seconde constante. En suivant le même raisonnement pour la dérivée seconde, l'annulation de ce terme signifie que l'on souhaite une dérivée première constante. Après une brusque augmentation de la commande, les valeurs des dérivées premières et secondes sont élevées, et vont le rester si on accorde trop d'importance à la minimisation de la dérivée troisième. On constate alors un emballement de la commande. Pour cette raison les coefficients de pondération sont à choisir avec précaution. De plus la minimisation des dérivées quatrièmes et supérieures n'apporte pas d'améliorations notables. En pratique, on peut se limiter à la

minimisation des dérivées premières et secondes et choisir les matrices de pondération Q_1 et Q_2 petites avec en plus $Q_2 < Q_1$.

3.3. Calcul de la loi de commande

La commande s'obtient en minimisant le critère quadratique H par rapport aux variables à optimiser U_{fut} . Le minimum de ce critère est atteint quand les conditions suivantes sont satisfaites :

$$\frac{\partial H}{\partial U_{fut}} = 0 \text{ et } \frac{\partial^2 H}{\partial U_{fut}^2} > 0 \quad (4.13)$$

Ce calcul déduit des équations (4.7), (4.11) et (4.13) conduit au système suivant :

$$\left(J_{u,fut}^T K_{x,fut}^T K_{x,fut} J_{u,fut} + \sum_{i=1}^s S_{i,fut}^T Q_i S_{i,fut} \right) U_{fut} = J_{u,fut}^T K_{x,fut}^T Y_{dl,fut} - \sum_{i=1}^s S_{i,fut}^T Q_i S_{i,pas} U_{i,pas} \quad (4.14)$$

En choisissant les matrices Q_i constantes, les termes $S_{i,fut}^T Q_i S_{i,fut}$ et $S_{i,fut}^T Q_i S_{i,pas}$ sont fixes et ne sont pas à calculer à chaque pas de calcul. Les commandes U_{fut} sont alors déterminées en résolvant le système obtenu. La commande à appliquer à l'instant $k+1$ est la dernière commande du vecteur U_{fut} qui correspond par définition de l'algorithme au terme u_{k+1} .

3.4. Exemple d'application sur un robot PUMA 560

Cette loi de commande est validée en simulation sur plusieurs systèmes mécaniques, dont en particulier un robot PUMA 560 qui est un robot à 6 degrés de liberté. La description et la modélisation dynamique de ce robot [COR 96] permettent de simuler ce système et de tester la méthode d'identification et la loi de commande établie auparavant.

Le modèle linéaire discret utilisé pour obtenir une approximation du système comporte six sorties et six entrées correspondant aux commandes et angles de rotation des six axes. Pour chacune de ces variables, deux régressions dont la période est de 0.01s sont insérées dans le modèle, ce qui donne au total 144 paramètres à adapter. Leur adaptation est réalisée avec l'algorithme du chapitre II.4.3 avec 60 points mémorisés dans la fenêtre d'observation auxquels s'ajoutent les 25 points « artificiels » déduits du modèle. Pour simplifier l'algorithme d'adaptation, les coefficients L_i qui correspondent à la distance entre les points « artificiels » sont choisis identiques et valent 0.01. Au lieu de régler ces distances L_i , il est en effet plus commode et avantageux d'appliquer des gains multiplicateurs sur les commandes des axes 1,2,3 et 4,5,6 valant respectivement 1000 et 100. Les couples à appliquer sur chaque axe du robot sont calculés pour réduire les erreurs sur un horizon de prédiction de 6 pas, avec les matrices de pondération Q_1 et Q_2 limitant les variations des commandes égales à :

$$Q_1 = 10^{-2} I \text{ et } Q_2 = 10^{-4} I$$

Chapitre IV : Détermination de la loi de commande

Les trajectoires désirées et obtenues pour les angles des axes 1, 2, 3, 4, 5 et 6 ainsi que les couples calculés sont représentés sur les figures 4.1 et 4.2.

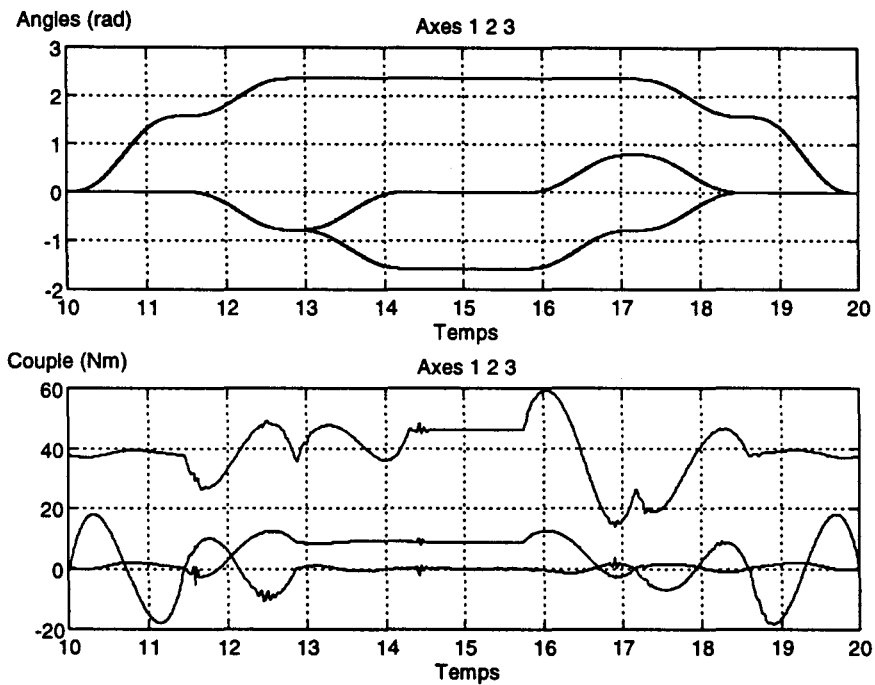


Figure 4.1 : Angles et couples des axes 1, 2 et 3

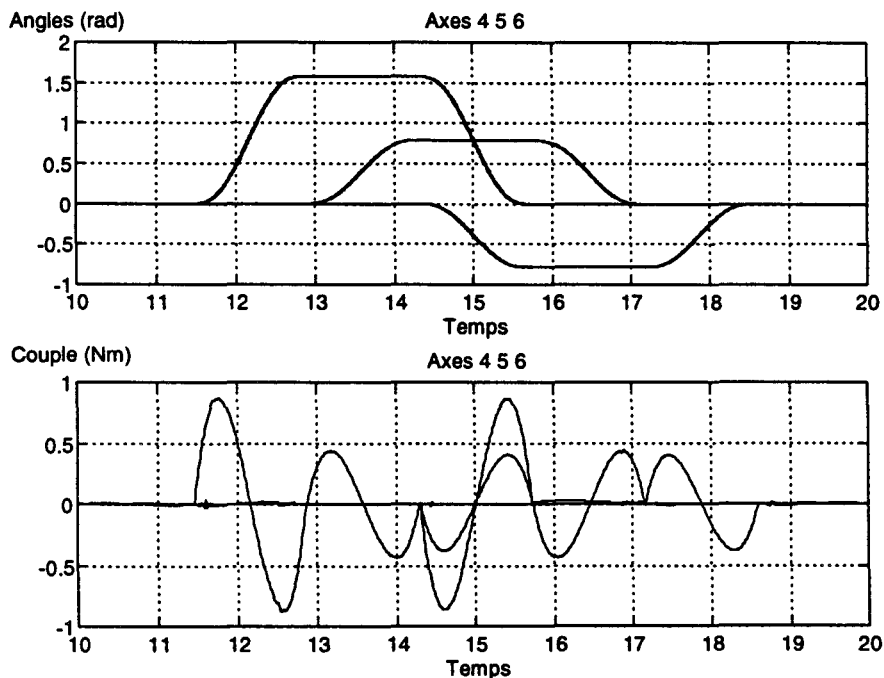


Figure 4.2 : Angles et couples des axes 4,5 et 6

Ces simulations montrent la capacité du contrôleur à commander des systèmes multivariables non linéaires. Les commandes obtenues sont quasiment lisses et présentent de faibles irrégularités uniquement lors des changements de consigne.

On constate également que la précision du suivi de trajectoire est satisfaisante. La norme des erreurs angulaires pour chaque axe est représentée sur la figure 4.3.

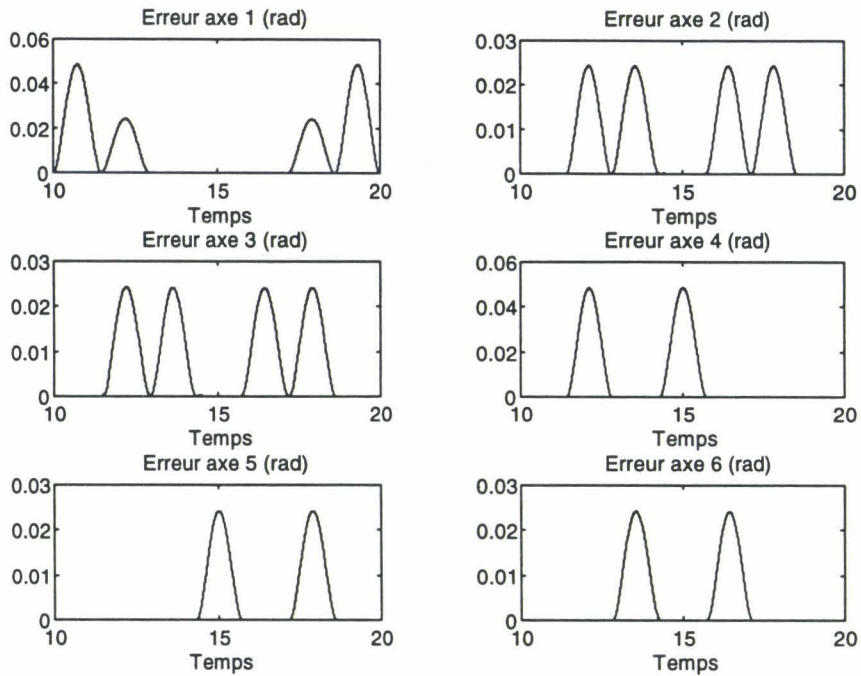


Figure 4.3 : Norme des erreurs angulaires des axes 1, 2, 3, 4, 5 et 6

Les erreurs angulaires moyennes et maximales sont respectivement inférieures à 0.6 et à 3 degrés sur tous les axes. On observe seulement un léger retard entre la trajectoire désirée et celle obtenue, ce qui explique l'allure des erreurs en forme de « bosses » lors des variations de consigne.

4. Commande minimisant les erreurs et les variations des erreurs

4.1. Définition du critère

L'introduction dans un même critère de termes portant sur la commande et sur les erreurs rend difficile la détermination appropriée des coefficients de pondération. En effet, le principal inconvénient de la loi de commande précédente provient de la difficulté pour choisir correctement les coefficients limitant les dérivées successives de la commande. De plus un important changement sur le processus implique un nouveau réglage de ces coefficients pour conserver des performances satisfaisantes et éviter une dégradation de la loi de commande. Le problème du réglage limite donc fortement l'intérêt et les performances du contrôleur ainsi déduit lorsque le système fonctionne dans des conditions variables. Pour profiter pleinement des avantages de la commande adaptative, les coefficients qui pondèrent les termes intervenant dans le critère ne doivent pas dépendre des paramètres du modèle, mais seulement des performances désirées. L'indépendance entre les coefficients du critère et le modèle permet alors d'utiliser sans modification le même critère et la

Chapitre IV : Détermination de la loi de commande

même loi de commande pour un processus dont le gain statique, par exemple, évolue dans des proportions très élevées.

Le rôle du critère est cependant toujours d'aboutir à une loi de commande minimisant les erreurs entre la consigne et les sorties mesurées. L'allure de la commande obtenue doit également être la plus régulière possible afin de limiter les efforts sur les actionneurs et le processus. Pour tenter de parvenir à ce double objectif, les contraintes ajoutées dans le critère sont telles que la correction des erreurs soit graduelle et adoucie. Le critère est composé d'un premier terme qui minimise les erreurs alors qu'un second terme contraint les variations des erreurs notées Δe_{fut} à être les plus faibles possibles. Les erreurs sont donc annulées progressivement. L'ajout d'un coefficient de pondération q affectant le premier terme permet de jouer sur la rapidité de convergence des sorties vers la consigne. Un troisième terme est inséré dans le critère pour limiter les variations de la commande δu , et pour résoudre les éventuels problèmes numériques survenant lors la résolution du système final. Le critère s'écrit donc :

$$H = \frac{1}{2} \left(q e_{fut}^T e_{fut} + \Delta e_{fut}^T \Delta e_{fut} + \delta u^T R_u \delta u \right) \quad (4.15)$$

avec

e_{fut} les erreurs prédites,

Δe_{fut} les variations des erreurs futures définies ainsi :

$$\Delta e_{fut} = \begin{bmatrix} e_{fut,k+h} - e_{fut,k+h-1} \\ \vdots \\ e_{fut,k+2} - e_{fut,k+1} \\ e_{fut,k+1} - e_k \end{bmatrix} = \begin{bmatrix} I & -I & & & \\ & \ddots & \ddots & & \\ & & I & -I & \\ & & & I & -I \end{bmatrix} \begin{bmatrix} e_{fut,k+h} \\ \vdots \\ e_{fut,k+1} \\ e_k \end{bmatrix} \quad (4.16)$$

e_k l'erreur à l'instant k : $e_k = Y_{d,k} - Y_k$

δu la variation de la commande : $\delta u = u_{k+1} - u_k$

et R_u une matrice de pondération des variations des commandes choisie égale à $10^{-5}I$.

Les variations des erreurs se formulent plus simplement sous forme matricielle :

$$\Delta e_{fut} = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} e_{fut} \\ e_k \end{bmatrix} \text{ avec } P_1 = \begin{bmatrix} I & -I & & & \\ & I & -I & & \\ & & \ddots & \ddots & \\ & & & I & -I \\ & & & & I \end{bmatrix} \text{ et } P_2 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -I \end{bmatrix}. \quad (4.17)$$

Avec ces notations, le critère H s'écrit alors :

$$H = \frac{1}{2} \left(q e_{fut}^T e_{fut} + e_{fut}^T P_1^T P_1 e_{fut} + e_k^T P_2^T P_2 e_k + e_{fut}^T P_1^T P_2 e_k + e_k^T P_2^T P_1 e_{fut} + \delta u^T R_u \delta u \right) \quad (4.18)$$

Chapitre IV : Détermination de la loi de commande

D'après les résultats du troisième chapitre et en choisissant l'allure du vecteur des commandes futures en rampe ou en échelon, le calcul des erreurs prédites dans le repère global vaut:

$$e_{fut} = V - K_{x,fut} J_{u,fut} H_{fut} \delta u \text{ avec } V = Y_{d,fut} - K_{x,fut} V_{pas} - K_{x,fut} J_{u,fut} \begin{bmatrix} u_k \\ \vdots \\ u_k \end{bmatrix} \quad (4.19)$$

avec $H_{fut} = H_{fut,0}$ pour un échelon et $H_{fut} = H_{fut,1}$ pour une rampe.

Le calcul de la dérivée du critère par rapport à δu donne alors :

$$\begin{aligned} \frac{\partial H}{\partial \delta u} = & -q H_{fut}^T J_{u,fut}^T K_{x,fut}^T (V - K_{x,fut} J_{u,fut} H_{fut} \delta u) \\ & - H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_1 (V - K_{x,fut} J_{u,fut} H_{fut} \delta u) \\ & - H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_2 e_k + R_u \delta u \end{aligned} \quad (4.20)$$

La structure triangulaire de la matrice $K_{x,fut}$, et la définition des matrices P_2 et $J_{u,fut}$ permettent de déduire le résultat suivant :

$$H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_2 e_k = 0 \quad (4.21)$$

En imposant $\frac{\partial H}{\partial \delta u} = 0$ pour obtenir la solution optimale, on déduit le système à résoudre dont

découle la variation de la commande δu à appliquer au processus :

$$\begin{aligned} & (H_{fut}^T J_{u,fut}^T K_{x,fut}^T (P_1^T P_1 + qI) K_{x,fut} J_{u,fut} H_{fut} + R_u) \delta u = \\ & H_{fut}^T J_{u,fut}^T K_{x,fut}^T (P_1^T P_1 + qI) \left(Y_{d,fut} - K_{x,fut} V_{pas} - K_{x,fut} J_{u,fut} \begin{bmatrix} u_k \\ \vdots \\ u_k \end{bmatrix} \right) \end{aligned} \quad (4.22)$$

La loi de commande dans le repère local s'obtient en suivant la même démarche que précédemment et donne le système suivant à résoudre :

$$\begin{aligned} & (H_{fut}^T J_{u,fut}^T K_{x,fut}^T (P_1^T P_1 + qI) K_{x,fut} J_{u,fut} H_{fut} + R_u) \delta u = \\ & H_{fut}^T J_{u,fut}^T K_{x,fut}^T (P_1^T P_1 + qI) (\Delta Y_{d,fut} - K_{x,fut} \Delta V_{pas}) \end{aligned} \quad (4.23)$$

Seul le calcul du terme de droite diffère entre les deux formulations (4.22) et (4.23). Le calcul dans le repère local est plus précis et moins sensible aux erreurs d'arrondi et aux erreurs d'identification. Son utilisation est donc préférable.

4.2. Influence de l'allure des commandes futures

Une commande future en échelon permet d'obtenir des temps de réponse généralement plus courts et une meilleure dynamique d'annulation des erreurs par rapport à une commande supposée évoluer en rampe. Cependant les améliorations des performances s'effectuent au détriment de la robustesse

vis à vis du bruit et des erreurs de modélisation. Les variations de la commande sont en effet plus irrégulières et plus brusques en prenant une allure de la commande en échelon.

En partant du principe que l'on accorde une plus grande importance à la robustesse de la loi de commande qu'aux performances dynamiques, il est préférable lors du calcul de la minimisation des erreurs futures d'imposer une allure de la commande en rampe.

4.3. Commande d'un système multivariable

Le système multivariable que l'on souhaite commander avec le critère minimisant les variations d'erreurs et les erreurs est celui déjà étudié au premier chapitre (1.31). Le système est donc défini par l'équation d'état ci-dessous :

$$\begin{cases} \dot{x} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -2 & -5 & -2 & 0 \\ -1 & 0 & 0 & 1 \\ -1 & -1 & -12 & -2 \end{bmatrix} x + \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 2 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x \end{cases} \quad (4.24)$$

Le modèle utilisé est adapté à partir d'un ensemble de 100 points, et comporte deux régressions de période 0.02s pour chacune des sorties et des commandes. Le seuil σ pour insérer un nouveau point dans la fenêtre d'observation est fixé à 0.1. Le coefficient de pondération q introduit dans le critère vaut également 0.1. La commande est alors calculée pour minimiser ce critère sur un horizon de 8 prédictions. Les commandes obtenues et les réponses du système par rapport aux consignes sinusoïdales et en créneaux sont représentées ci-dessous :

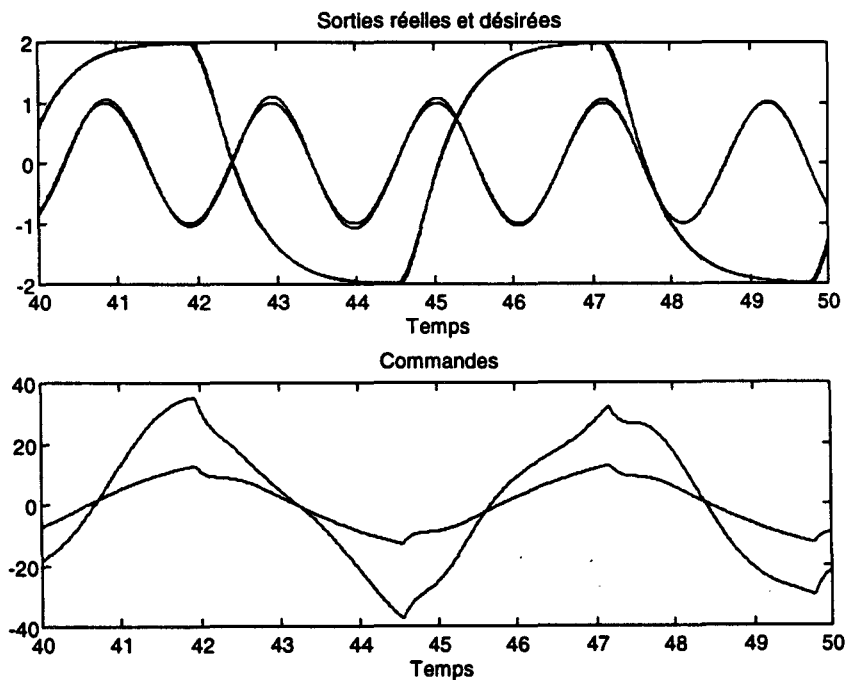


Figure 4.4 : Commande d'un processus multivariable

Ces simulations montrent que l'algorithme de commande contrôle correctement ce processus multivariable. Il est alors intéressant de considérer le cas où le système est instable.

4.4. Commande d'un système multivariable instable

L'initialisation pour l'identification de ces systèmes est problématique car une bonne commande doit être appliquée immédiatement pour éviter que le système diverge pendant l'adaptation préliminaire du modèle. Les solutions retenues pour commander ce type de processus sont les suivantes. La connaissance a priori d'un modèle du processus permet en utilisant les formules rappelées au chapitre sur la modélisation, de convertir un modèle continu en modèle discret. Ce modèle initial peut alors servir au calcul de la commande pendant les premiers instants. La mise en marche du contrôleur est ainsi moins problématique. D'autre part, l'acquisition des mesures à une fréquence plus élevée permet de réduire le temps initial d'adaptation du modèle et contribue à la résolution des problèmes de divergence rencontrés lors de l'initialisation ou lors d'une modification du processus.

Le système à commander est similaire au précédent mais le quatrième coefficient de la première ligne de la matrice A est modifié pour avoir une des valeurs propres de A positive. Le système instable est alors décrit par les équations différentielles suivantes :

$$\left\{ \begin{array}{l} \dot{x} = \begin{bmatrix} 0 & 1 & 0 & 3 \\ -2 & -5 & -2 & 0 \\ -1 & 0 & 0 & 1 \\ -1 & -1 & -12 & -2 \end{bmatrix} x + \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 2 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x \end{array} \right. \quad (4.25)$$

A cause du pôle instable, la période de régression du modèle est choisie égale à 0.01s au lieu de 0.02s mais les autres paramètres de réglage sont conservés. Les réponses du système et les commandes valent alors :

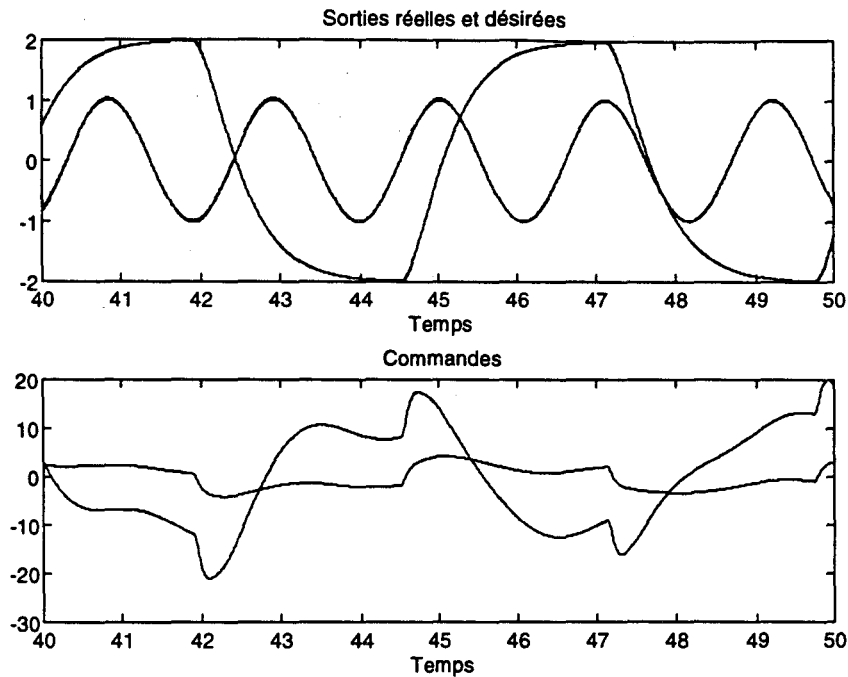


Figure 4.5 : Commande d'un processus multivariable instable

Les simulations avec ce système instable sont satisfaisantes. Pour mettre alors en évidence la rapidité d'adaptation du modèle, le passage du système stable (4.24) vers le système instable est effectué en cours de simulation à l'instant $t = 44s$. Les courbes obtenues sont représentées sur la figure 4.6.

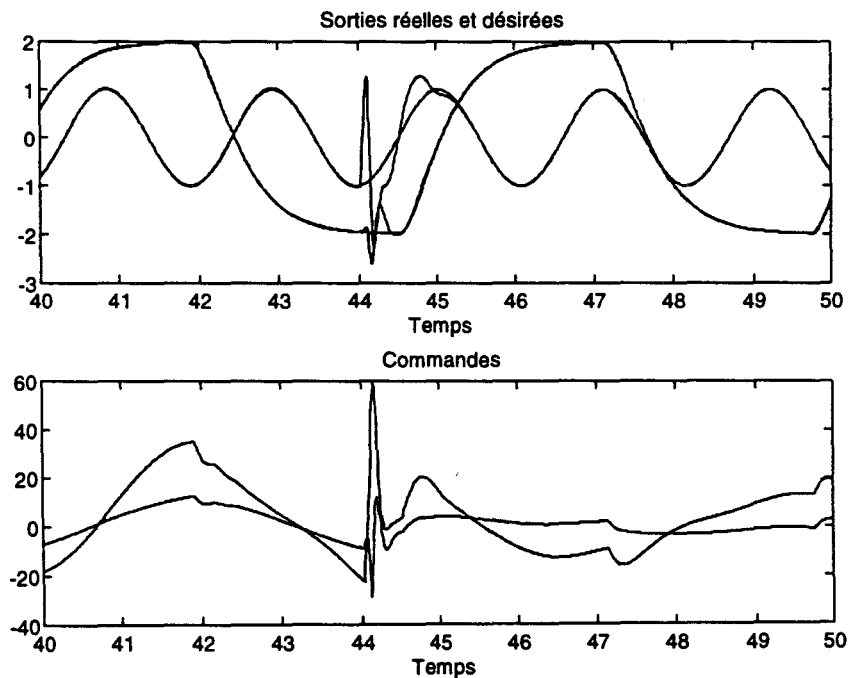


Figure 4.6 : Commande d'un processus multivariable variant à un instant donné

Le critère aboutissant à cette loi de commande adaptative et prédictive donne donc des résultats probants. Cependant d'autres approches sont à envisager pour contrôler d'autres types de processus.

5. Commande avec réglage du taux d'annulation de l'erreur

5.1. Définition du critère

La formulation du critère peut être établie de manière à imposer le taux de décroissance de l'erreur. L'objectif est de contraindre les erreurs entre la consigne et les valeurs futures à évoluer suivant la loi :

$$\dot{e} = A_e e \quad (4.26)$$

avec e l'erreur et A_e une matrice strictement négative fixant la décroissance des erreurs.

En discret avec un pas d'échantillonnage h_e , les erreurs doivent alors obéir à la loi suivante :

$$e_{k+1} = (I + h_e A_e) e_k$$

ou encore

$$e_{k+1} = A_{ed} e_k \text{ avec } A_{ed} = I + h_e A_e \quad (4.27)$$

Le critère est défini tel que les nouvelles erreurs $\varepsilon_i = e_{k+i+1} - A_{ed} e_{k+i}$ sur l'ensemble des prédictions soient minimisées. Le vecteur des erreurs ε_i sur l'horizon de prédiction s'écrit alors :

$$\varepsilon = \begin{bmatrix} I & -A_{ed} & & & & \\ & I & -A_{ed} & & & \\ & & \ddots & \ddots & & \\ & & & I & -A_{ed} & \\ & & & & & \end{bmatrix} \begin{bmatrix} e_{fut,k+h} \\ e_{fut,k+h-1} \\ e_{fut,k+h-2} \\ \vdots \\ e_{fut,k+1} \\ e_k \end{bmatrix} \quad (4.28)$$

avec $e_{fut} = Y_{d,fut} - Y_{fut}$ l'erreur entre les sorties désirées et prédites.

Le vecteur des erreurs ε peut s'écrire également sous la forme:

$$\varepsilon = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} e_{fut} \\ e_k \end{bmatrix} \quad (4.29)$$

avec les matrices P_1 et P_2 définies ci-dessous :

$$P_1 = \begin{bmatrix} I & -A_{ed} & & & & \\ & I & -A_{ed} & & & \\ & & \ddots & \ddots & & \\ & & & I & -A_{ed} & \\ & & & & & I \end{bmatrix}, P_2 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -A_{ed} \end{bmatrix}$$

Les différences entre les nouvelles matrices P_1 et P_2 par rapport aux matrices de l'équation (4.17) proviennent de la substitution du terme $-I$ par la matrice $-A_{ed}$. Le réglage de la dynamique d'annulation des erreurs s'effectue alors en fonction de la matrice A_{ed} et non plus en fonction du coefficient q du critère (4.18).

Chapitre IV : Détermination de la loi de commande

Le critère est également défini pour limiter les variations de la commande δu et pour éviter les problèmes numériques lors de la résolution du système. L'allure du vecteur des commandes futures est choisie en rampe ($H_{fut} = H_{fut,1}$). Le critère à minimiser est donc le suivant :

$$H = \frac{1}{2} (\varepsilon^T \varepsilon + \delta u^T R_u \delta u) \quad (4.30)$$

ce qui donne après développement des calculs :

$$H = \frac{1}{2} (e_{fut}^T P_1^T P_1 e_{fut} + e_k^T P_2^T P_2 e_k + e_{fut}^T P_1^T P_2 e_k + e_k^T P_2^T P_1 e_{fut} + \delta u^T R_u \delta u) \quad (4.31)$$

La dérivée du critère par rapport à δu est alors égal à :

$$\frac{\partial H}{\partial \delta u} = -H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_1 (V - K_{x,fut} J_{u,fut} H_{fut} \delta u) - H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_2 e_k + R_u \delta u \quad (4.32)$$

avec

$$H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_2 e_k = 0 \text{ pour les mêmes raisons que celles évoquées à la section IV.4.1.}$$

La loi de commande découle donc du système ci-dessous dont la résolution donne la variation de la commande à appliquer :

$$\begin{aligned} & \left(H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_1 K_{x,fut} J_{u,fut} H_{fut} + R_u \right) \delta u = \\ & H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_1 \left(Y_{d,fut} - K_{x,fut} V_{pas} - K_{x,fut} J_{u,fut} \begin{bmatrix} u_k \\ \vdots \\ u_k \end{bmatrix} \right) \end{aligned} \quad (4.33)$$

En suivant le même principe pour déterminer la loi de commande dans le repère local, le système à résoudre devient :

$$\begin{aligned} & \left(H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_1 K_{x,fut} J_{u,fut} H_{fut} + R_u \right) \delta u = \\ & H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_1 \left(\Delta Y_{d,fut} - K_{x,fut} \Delta V_{pas} \right) \end{aligned} \quad (4.34)$$

Le calcul du terme $P_1 K_{x,fut}$ peut être optimisé en reprenant à la base le calcul de la matrice $K_{x,fut}$.

Nous posons $L_{x,fut} = P_1 K_{x,fut}$ qui est une matrice dont la structure est similaire à celle de $K_{x,fut}$:

$$K_{x,fut} = \begin{bmatrix} I & K_1 & K_2 & \cdots & K_n \\ & I & K_1 & \cdots & K_{n-1} \\ & & \ddots & & \vdots \\ & & & I & K_1 \\ & & & & I \end{bmatrix} \text{ et } L_{x,fut} = \begin{bmatrix} I & L_1 & L_2 & \cdots & L_n \\ & I & L_1 & \cdots & L_{n-1} \\ & & \ddots & & \vdots \\ & & & I & L_1 \\ & & & & I \end{bmatrix} \quad (4.35)$$

avec $L_i = K_i - A_{ed} K_{i-1}$ pour $i = 2$ à n .

ou bien $V = \Delta Y_{d,fut} - K_{x,fut} \Delta V_{pas}$ dans le repère local.

5.3. Exemple d'un système comportant un retard pur

Soit un système du second ordre avec un retard pur d'une seconde défini par la fonction de transfert suivante :

$$\frac{Y}{U} = \frac{26e^{-s}}{s^2 + 2s + 26} \quad (4.42)$$

Les pôles du système sont complexes conjugués et valent $-1 \pm 5j$ ce qui donne un caractère oscillant au système. Pour identifier le processus et le retard, le modèle choisi est d'ordre 6 ($nbry = 6$ et $nbru = 6$). La période d'acquisition des mesures est de 0.05s pour obtenir une adaptation rapide du modèle et une commande calculée à une fréquence suffisamment élevée. Un facteur de sous échantillonnage égal à 4 est cependant nécessaire pour que les régressions du modèle couvrent un intervalle de temps supérieur au retard. Le nombre de points dans la fenêtre d'observation est égal à 100 et le seuil σ pour l'insertion d'un nouveau point vaut 0.1.

Le calcul de la commande est alors défini pour minimiser les erreurs futures à partir du 6^{ème} pas jusqu'au 15^{ème} avec un coefficient de décroissance des erreurs fixé à 0.9. Le contrôleur ainsi défini permet d'aboutir au résultat suivant :

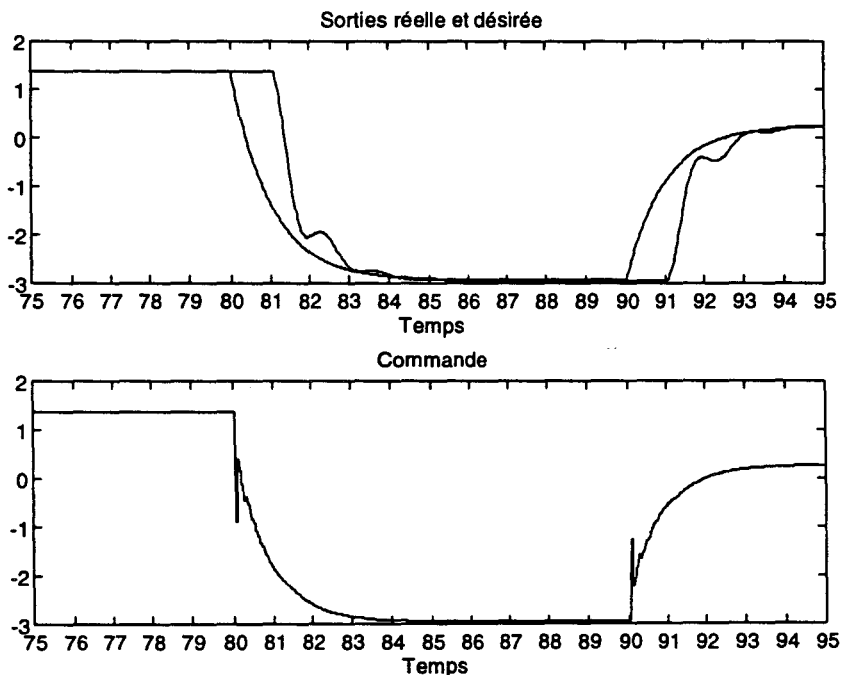


Figure 4.7 : Commande d'un processus avec un retard pur d'une seconde

Grâce à la modélisation jusqu'à l'ordre 6 qui permet d'inclure le retard dans le modèle, et avec le calcul des prédictions sur un horizon de 15 pas, la loi de commande obtenue est stable et la précision du suivi de trajectoire est satisfaisante au retard près.

5.4. Etude d'un système d'ordre élevé

L'étude suivante porte sur un système d'ordre élevé avec cinq pôles se trouvant en -0.5, -2, -3, -4 et -6, avec un zéro en -1 et avec un gain statique égal à 1/72. La fonction de transfert de ce système vaut donc :

$$\frac{Y}{U} = \frac{s+1}{(s+0.5)(s+2)(s+3)(s+4)(s+6)} \quad (4.43)$$

L'identification précise de ce type de systèmes est assez difficile à cause des nombreux pôles et des dynamiques assez voisines que l'on ne peut donc pas négliger ou simplifier. Mais l'approche adaptative et prédictive développée se prête bien à la commande de tels processus. En utilisant les mêmes paramètres de réglage que pour l'exemple précédent, à l'exception du coefficient de décroissance des erreurs qui vaut 0.8, les résultats obtenus sont les suivants :

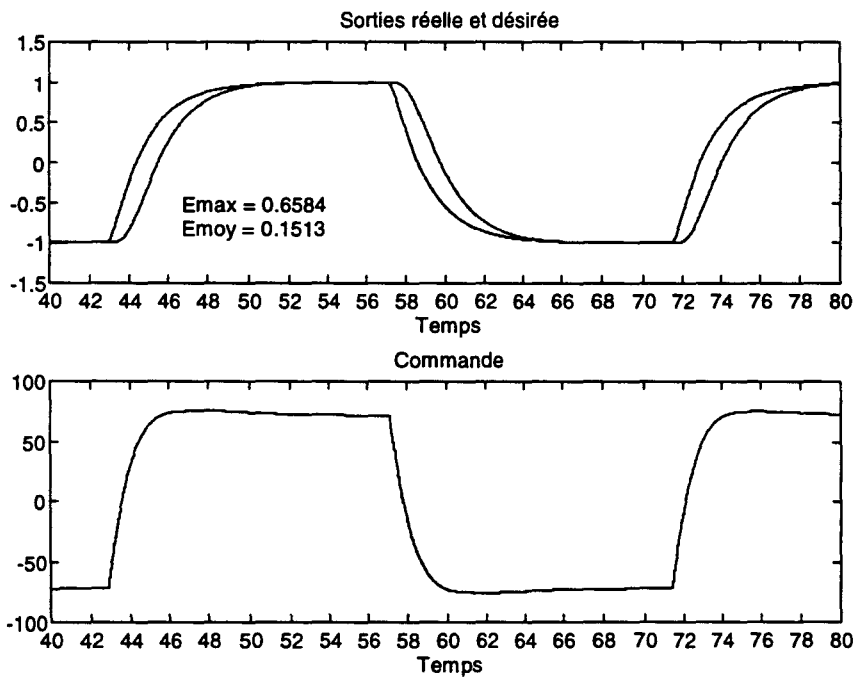


Figure 4.8 : Commande de système d'ordre élevé (cas n°1)

La commande obtenue est régulière et l'erreur en régime permanent tend vers zéro. Des erreurs importantes sont toutefois commises lors des régimes transitoires, puisque l'erreur maximale en valeur absolue vaut 0.65. La moyenne de la norme des erreurs vaut 0.15 ce qui est également assez élevé. La diminution de ces erreurs est possible en prenant une fréquence d'adaptation du modèle et de calcul de la commande plus rapide. Avec une période d'acquisition égale à 0.025s au lieu de 0.05s, et un facteur de sous échantillonnage égal à 2, les résultats sont plus précis mais l'identification et la commande sont beaucoup plus sensibles aux bruits de mesure :

Chapitre IV : Détermination de la loi de commande

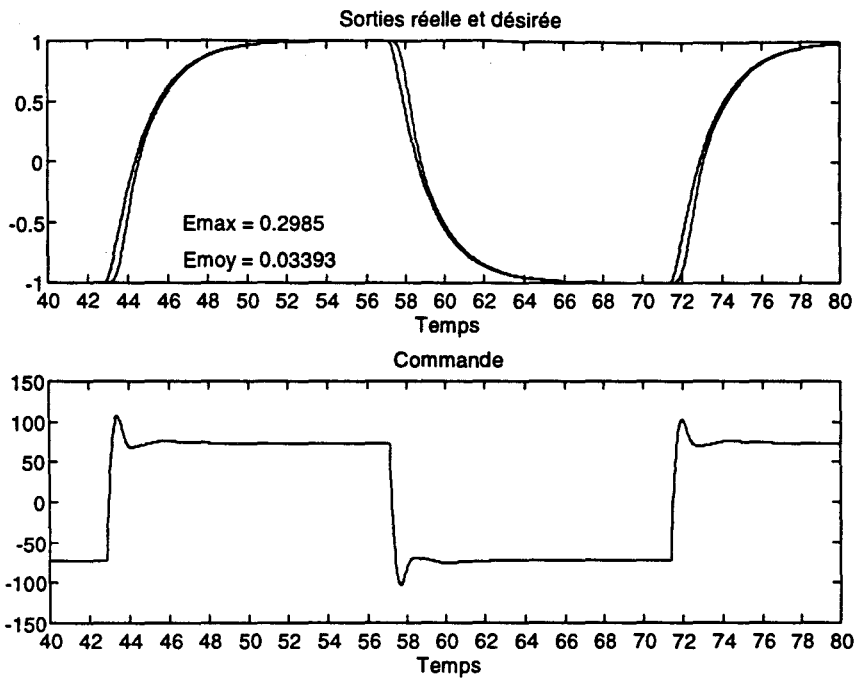


Figure 4.9 : Commande de système d'ordre élevé (cas n°2)

Les essais ci-dessous sont réalisés avec les mêmes réglages que dans les deux cas précédents mais en ajoutant un bruit de moyenne nulle et d'écart type égal à 0.01.

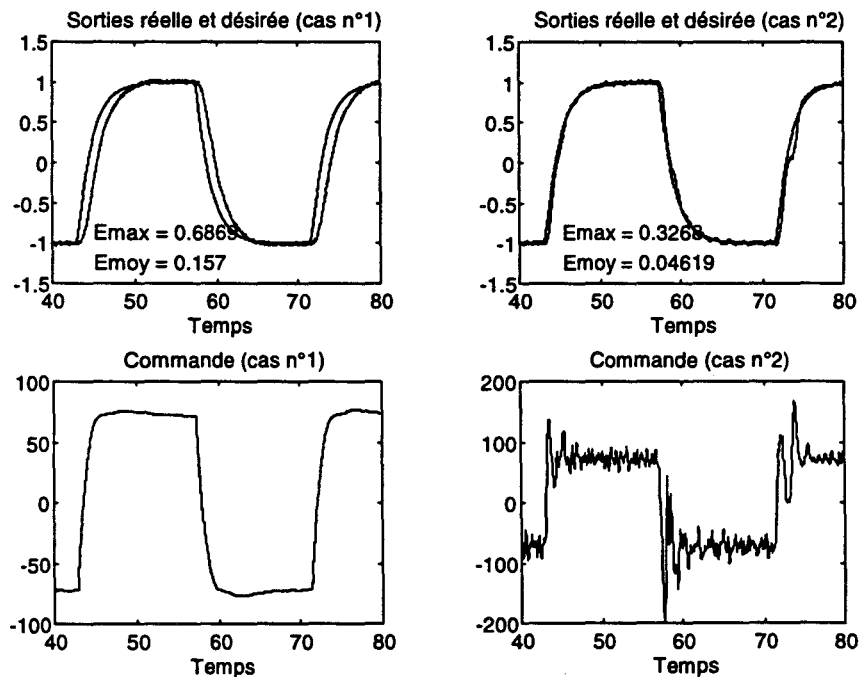


Figure 4.10 : Commande de système d'ordre élevé avec un bruit sur les mesures

Cet exemple montre le compromis entre la précision du suivi de trajectoire et la robustesse vis à vis des bruits. Le critère établi dans le chapitre suivant vise alors à améliorer ce compromis et à gagner à la fois en précision et en robustesse.

6. Commande avec filtrage des erreurs

Le critère est modifié de manière à calculer une loi de commande sur laquelle les bruits ont une influence réduite. Cette méthode dégrade légèrement les performances dynamiques et la précision lors des régimes transitoires, mais améliore les propriétés de robustesse de la loi de commande par rapport aux bruits de mesure. Ces derniers sont en effet atténués par un filtre qui fixe la dynamique d'annulation des erreurs.

6.1. Détermination des coefficients du filtre

La loi d'évolution des erreurs est définie par le filtre d'ordre $h-1$, avec h le nombre de prédictions. Les coefficients de ce filtre sont déterminés de manière à ce que les erreurs diminuent mais également de manière à pouvoir formuler certains calculs en récursif afin de limiter le nombre d'opérations nécessaires même quand l'ordre du filtre est élevé.

La structure du filtre est alors la suivante :

$$e_{k+1} = \frac{\alpha}{\beta} e_k + \frac{\alpha^2}{\beta} e_{k-1} + \frac{\alpha^3}{\beta} e_{k-2} + \dots + \frac{\alpha^{h-1}}{\beta} e_{k-h+2} \quad (4.44)$$

avec

α un coefficient réglant la dynamique d'annulation des erreurs ($|\alpha| < 1$)

et β est choisi tel que $\sum_{i=1}^{h-1} \frac{\alpha^i}{\beta} = \alpha$ pour garantir la stabilité du filtre, d'où $\beta = \frac{\alpha}{\sum_{i=1}^{h-1} \alpha^i}$

6.2. Définition du critère et résolution

De ce filtre est déduite la nouvelle erreur ε à minimiser :

$$\varepsilon = \begin{bmatrix} I & -\frac{\alpha}{\beta} I & -\frac{\alpha^2}{\beta} I & \dots & -\frac{\alpha^{h-1}}{\beta} I \\ & I & -\frac{\alpha}{\beta} I & -\frac{\alpha^2}{\beta} I & \dots \\ & & & \ddots & \ddots \\ & & & & I \end{bmatrix} \begin{array}{c} \vdots \\ -\frac{\alpha^{h-1}}{\beta} I \\ \vdots \\ -\frac{\alpha}{\beta} I \end{array} \begin{bmatrix} e_{fut,k+h} \\ e_{fut,k+h-1} \\ \vdots \\ e_{fut,k+1} \\ e_k \\ e_{k-1} \\ \vdots \\ e_{k-h+2} \end{bmatrix} \quad (4.45)$$

La définition adéquate du vecteur des erreurs passées e_{pas} et des matrices P_1 et P_2 permet d'aboutir à l'expression suivante :

$$\varepsilon = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} e_{fut} \\ e_{pas} \end{bmatrix} \quad (4.46)$$

Le critère H à minimiser et son écriture développée sont respectivement :

$$H = \frac{1}{2} (\varepsilon^T \varepsilon + \delta u^T R_u \delta u) \quad (4.47)$$

d'où

$$H = \frac{1}{2} (e_{fut}^T P_1^T P_1 e_{fut} + e_{pas}^T P_2^T P_2 e_{pas} + e_{fut}^T P_1^T P_2 e_{pas} + e_{pas}^T P_2^T P_1 e_{fut} + \delta u^T R_u \delta u) \quad (4.48)$$

L'optimisation de la commande δu par rapport au critère H donne en effectuant les calculs dans le repère global le système suivant :

$$\begin{aligned} & (H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_1 K_{x,fut} J_{u,fut} H_{fut} + R_u) \delta u = \\ & H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T \left(\begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} Y_{d,fut} \\ e_{pas} \end{bmatrix} - P_1 K_{x,fut} V_{pas} - P_1 K_{x,fut} J_{u,fut} \begin{bmatrix} u_k \\ \vdots \\ u_k \end{bmatrix} \right) \end{aligned} \quad (4.49)$$

Dans le repère autour du point de fonctionnement, le système à résoudre est le suivant :

$$\begin{aligned} & (H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T P_1 K_{x,fut} J_{u,fut} H_{fut} + R_u) \delta u = \\ & H_{fut}^T J_{u,fut}^T K_{x,fut}^T P_1^T \left(\begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} \Delta Y_{d,fut} \\ e_{pas} \end{bmatrix} - P_1 K_{x,fut} \Delta V_{pas} \right) \end{aligned} \quad (4.50)$$

L'avantage de ce critère apparaît clairement car les erreurs passées e_{pas} et les variations de la consigne $\Delta Y_{d,fut}$ sont filtrées par les matrices P_1 et P_2 . L'influence du bruit et des perturbations est donc réduite. Toutefois la consigne est suivie avec une moins grande précision.

6.3. Formulation récursive des calculs

Le produit entre les matrices $\begin{bmatrix} P_1 & P_2 \end{bmatrix}$ et $\begin{bmatrix} \Delta Y_{d,fut} \\ e_{pas} \end{bmatrix}$ consiste à appliquer le filtre au vecteur considéré

ce qui peut s'effectuer en calculant un premier élément entièrement pour l'initialisation puis par récurrence pour réduire le nombre de calculs. Soient Y et Y_f les vecteurs avant et après filtrage, et y_{k+i} et $y_{f,k+i}$ leurs éléments respectifs aux instants $k+i$, l'application de la matrice $\begin{bmatrix} P_1 & P_2 \end{bmatrix}$ de filtrage s'exprime alors en définissant une suite intermédiaire s_{k+i} de la manière suivante :

$$\begin{cases} s_{k+1} = \sum_{j=1}^l \frac{\alpha^j}{\beta} y_{k+1-j} \\ y_{f,k+1} = y_{k+1} - s_{k+1} \end{cases} \quad (4.51)$$

puis par récurrence pour $i = 1$ à $h-1$:

$$\begin{cases} s_{k+i+1} = \frac{\alpha}{\beta} y_{k+i} + \alpha \left(s_{k+i} - \frac{\alpha^i}{\beta} y_{k+i-i} \right) \\ y_{f,k+i+1} = y_{k+i+1} - s_{k+i+1} \end{cases} \quad (4.52)$$

Le filtrage des erreurs avec le filtre défini en (4.44) se réalise donc aisément.

Les termes du produit des matrices $P_1 K_{x,fut}$ s'obtiennent en développant les calculs et en comparant chacun des termes ce qui conduit aux expressions ci-dessous. La structure des deux matrices permet d'aboutir à une autre matrice triangulaire du type :

$$L_{x,fut} = P_1 K_{x,fut} = \begin{bmatrix} I & L_1 & L_2 & \cdots & L_n \\ & I & L_1 & \cdots & L_{n-1} \\ & & \ddots & & \vdots \\ & & & I & L_1 \\ & & & & I \end{bmatrix} \quad (4.53)$$

Les termes L_i et L_{i+1} valent alors:

$$L_i = K_i - \sum_{j=1}^i \frac{\alpha^j}{\beta} K_{i-j} \text{ et } L_{i+1} = K_{i+1} - \sum_{j=1}^{i+1} \frac{\alpha^j}{\beta} K_{i+1-j} \quad (4.54)$$

La comparaison des deux équations permet de déduire la relation :

$$L_{i+1} - \alpha L_i = K_{i+1} - \left(\alpha + \frac{\alpha}{\beta}\right) K_i \quad (4.55)$$

En posant $\gamma = \alpha + \frac{\alpha}{\beta}$ et $S_i = K_i - \gamma K_{i-1}$, une nouvelle suite S_i est à déterminer à partir de

l'expression des K_i . On aboutit alors aux récurrences suivantes :

$$S_1 = J_{x,1} - \gamma I \quad (4.56)$$

$$S_i = \sum_{k=1}^{i-1} S_{i-k} J_{x,k} + J_{x,i} \text{ pour } i \leq \text{nby}, \quad (4.57)$$

$$S_i = \sum_{k=1}^{\text{nby}} S_{i-k} J_{x,k} \text{ pour } i > \text{nby}. \quad (4.58)$$

Les éléments L_i à calculer se déduisent ensuite de la relation :

$$L_i = S_i + \alpha L_{i-1} \quad (4.59)$$

L'utilisation dans le critère d'un filtre d'ordre $h-1$ choisi judicieusement selon l'équation (4.44) réduit la sensibilité au bruit de la loi de commande sans pour autant alourdir démesurément les calculs.

6.4. Exemple de commande d'un système bruité

L'exemple ci-dessous a pour objectif de montrer les avantages de ce critère en présence de bruit. La commande est alors plus régulière et le bruit n'est pas amplifié. Toutefois l'identification du processus et l'adaptation du modèle restent toujours sensibles au bruit si aucun filtrage n'est réalisé sur les mesures servant à la détermination du modèle. La commande avec filtrage des erreurs ne

Chapitre IV : Détermination de la loi de commande

présente donc de véritable intérêt que si le modèle est obtenu avec une précision suffisante. La méthode de filtrage proposée au chapitre II.6.1 est donc requise lors de l'identification.

Le système étudié est défini par la fonction de transfert suivante :

$$\frac{Y}{U} = \frac{12}{s^2 + 8s + 12} \quad (4.60)$$

Le modèle à adapter est du second ordre ($nbry = 2$ et $nbru = 2$) dont le pas d'échantillonnage est choisi égal à 0.02s. L'adaptation est alors réalisée à partir d'un ensemble de 100 mesures. L'ajout d'un nouveau point dans la fenêtre d'observation a lieu seulement quand l'écart entre la mesure actuelle et la dernière mesure mémorisée est supérieur à 0.2. De plus les mesures servant à adapter le modèle sont filtrées en calculant leur moyenne sur 10 points ($nbf = 10$).

La loi de commande est établie en calculant 10 prédictions, et les paramètres du filtre inséré dans le critère sont $\alpha = 0.8$ et $\beta = 4.3289$.

La simulation du système auquel un bruit de moyenne nulle et d'écart type égal à 0.1 a été ajouté donne les résultats présentés sur la figure 4.11.

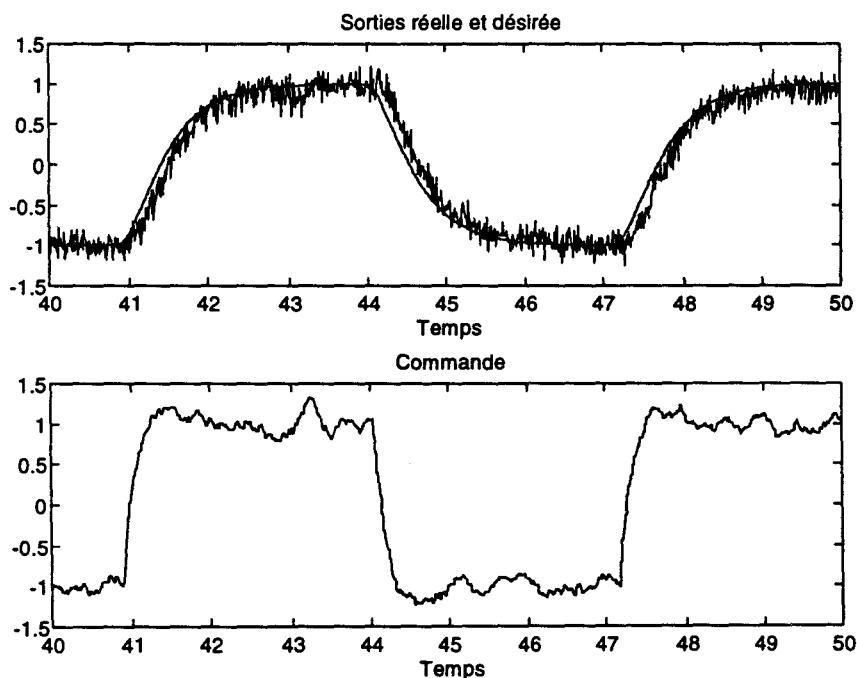


Figure 4.11 : Influence du bruit de mesure sur la commande

La présence d'un bruit important sur les mesures n'entraîne pas avec ce critère de dégradations majeures de la précision et de la loi de commande. Les objectifs concernant la robustesse des algorithmes vis à vis du bruit sont donc atteints.

6.5. Etude d'un système à non minimum de phase

Ces systèmes ont la particularité de réagir différemment à la commande et aux variations de la commande. L'influence sur la sortie du processus de la commande et de sa dérivée sont opposés ce qui se traduit sur les fonctions de transfert par la présence d'un zéro dont la partie réelle est positive. [DOR 95].

Un exemple aussi simple que celui défini ci-dessous montre la particularité d'un tel système et la difficulté pour déterminer la loi de commande qui ne doit annuler les erreurs qu'à long terme pour rester stable. En effet la réduction des erreurs à court terme obligerait le contrôleur à compenser le zéro dont la partie réelle est positive. Cette compensation rendrait évidemment la loi de commande instable. Cette divergence est évitée en augmentant l'horizon de prédiction, et en accordant une plus grande importance aux erreurs à long terme. On peut alors modifier le critère comme au chapitre IV.5.2. pour ne minimiser que les erreurs à long terme.

L'analyse de cette loi commande et la comparaison avec un contrôleur Proportionnel Intégral dont le réglage est optimisé par une méthode d'analyse du lieu des racines, permet de valider le principe de la commande adaptative et prédictive et montre les avantages de cette approche.

Le système à non minimum de phase illustrant la méthode est défini par la fonction de transfert ci-dessous qui possède deux pôles réels en -2 et -6 et un zéro en 1. Le régulateur PI peut également se représenter par une fonction de transfert entre la commande et l'erreur.

$$\frac{Y}{U} = \frac{-12s+12}{s^2+8s+12} \text{ et } \frac{U}{E} = \frac{K_p s + K_i}{s} \quad (4.61)$$

Le réglage des gains K_p et K_i du contrôleur PI est effectué de manière à garantir la stabilité et d'aboutir à un temps de réponse le plus court possible avec un faible dépassement. La stabilité est obtenue quand tous les pôles du système bouclé ont leur partie réelle négative. Pour obtenir un amortissement suffisant et éviter les dépassements importants et les oscillations, la partie imaginaire des pôles doit être inférieure en valeur absolue à la partie réelle. Pour réduire le temps de réponse, il faut que la plus petite norme des racines soit la plus grande possible. Le résultat de cette optimisation en tenant compte de toutes les contraintes donne les valeurs de gains $K_p = 0.29$ et $K_i = 0.55$. Les trois pôles en boucle fermée sont en -1.53 et $-1.49 \pm 1.43i$. Le zéro du système n'est pas compensé et reste en -1, mais le contrôleur introduit un second zéro en -1.9.

La réponse du processus régulé et la commande appliquée sont présentées ci-dessous et montre clairement l'influence du zéro.

Chapitre IV : Détermination de la loi de commande

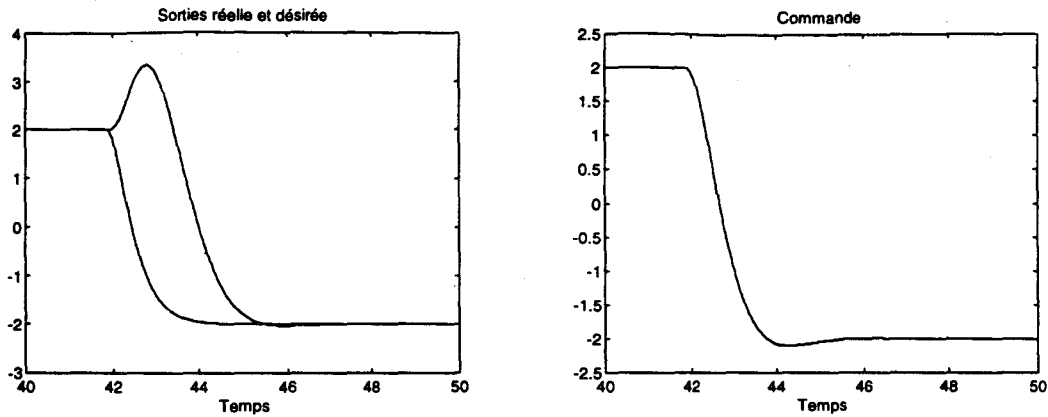


Figure 4.12 : Système à non minimum de phase contrôlé par un PI

La consigne qui passe de la valeur 2 à -2 est rejointe à 2% près par la sortie du processus au bout de 4 secondes.

Le même système est ensuite contrôlé par la méthode adaptative et prédictive basée sur le critère avec filtrage des erreurs. Le nombre de régressions utilisées pour modéliser le système est fixé à 3 ($nbry = 3$ et $nbru = 3$) et leur période est de 0.05s. Un facteur de sous échantillonnage égal à 2 est défini entre la période d'acquisition et le période du modèle. 300 points sont mémorisés dans la fenêtre d'observation avec un seuil de mise à jour σ égal à 0.1. Pour le contrôleur, les paramètres de filtrage sont $\alpha = 0.8$ et $\beta = 4.9279$. Le calcul de la commande est réalisé en minimisant les erreurs de la quinzième à la trentième prédiction. En effet les premières erreurs prédites ne peuvent pas être annulées à cause du zéro à valeur réelle positive. Les résultats obtenus avec ces paramètres de réglage sont les suivants :

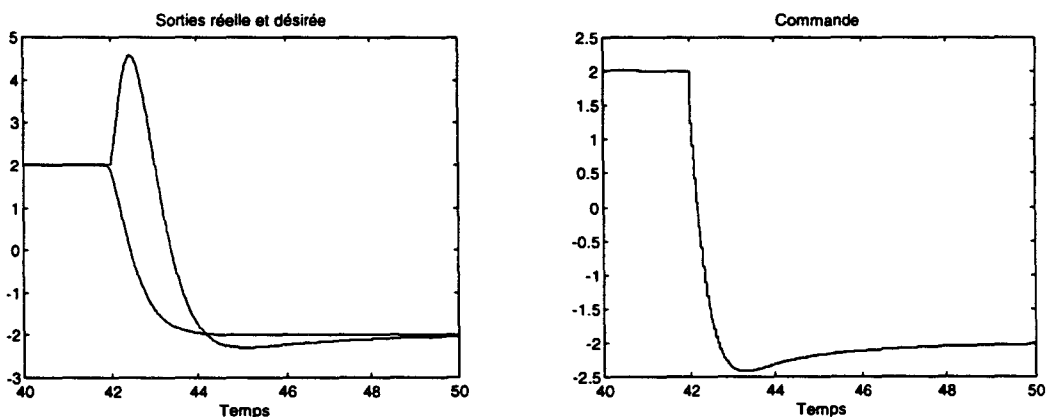


Figure 4.13 : Système à non minimum de phase contrôlé par la commande adaptative

Les dépassements avec cette loi de commande sont plus importants qu'avec le contrôleur Proportionnel Intégral dont le réglage a été optimisé. Le temps de réponse avoisine les 8 secondes et est donc plus long. Le principal avantage de la commande adaptative par rapport un régulateur PI bien réglé est alors la robustesse vis à vis des changements des paramètres.

Chapitre IV : Détermination de la loi de commande

Les simulations suivantes ont alors pour but de tester l'influence d'une modification du système dont la nouvelle fonction de transfert est supposé valoir :

$$\frac{Y}{U} = \frac{-18s + 12}{s^2 + 8s + 12} \quad (4.62)$$

Le zéro du système après changement est donc placé en 1.5 dans le plan des pôles et zéros. Les contrôleurs dont le réglage reste identique donnent alors les résultats suivants :

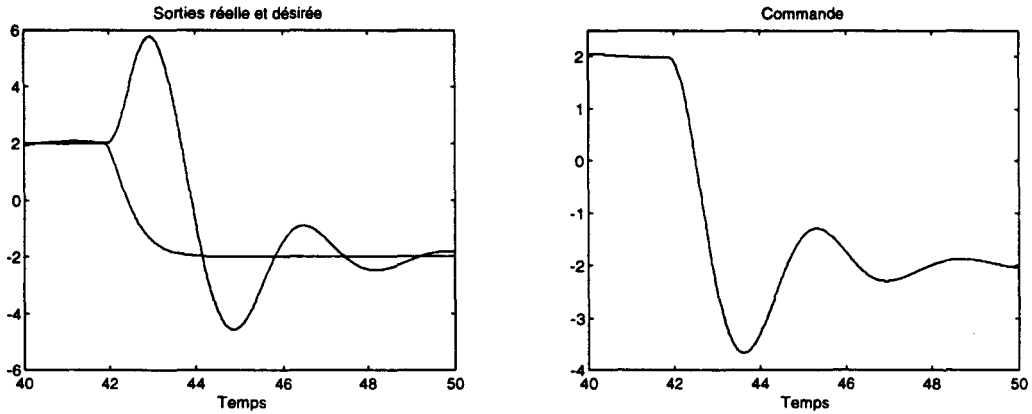


Figure 4.14 : Contrôleur PI avec le système modifié

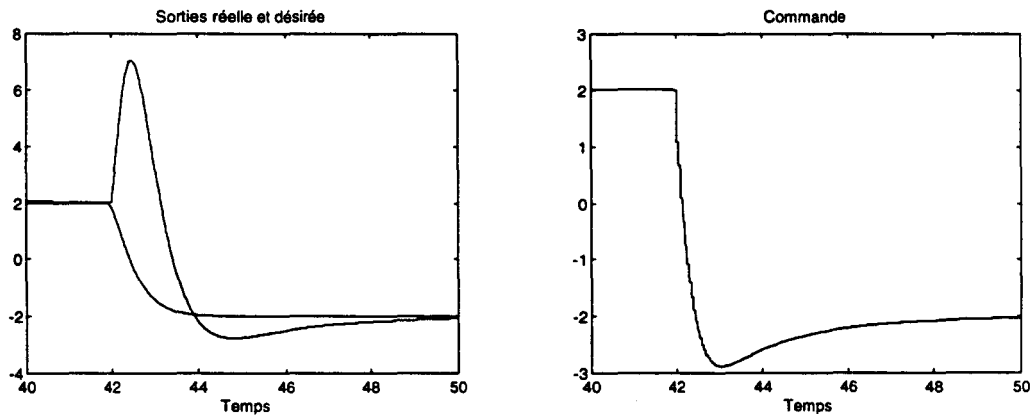


Figure 4.15 : Contrôleur adaptatif avec le système modifié

Ces simulations montrent que la commande adaptative et prédictive permet de contrôler des systèmes complexes variant dans le temps avec des performances qui restent satisfaisantes malgré des changements importants de la dynamique du système.

7. Conclusion

Les différents critères et lois de commande étudiés au cours de ce chapitre montrent à travers de multiples exemples les avantages et la robustesse de l'approche adaptative et prédictive. Les bruits sur les mesures, les retards purs, les non linéarités des processus, les changements de paramètres et les perturbations sont pris en compte lors de l'adaptation du modèle. Le calcul des prédictions et la minimisation des erreurs futures selon les quatre critères proposés permettent alors d'optimiser les valeurs de la commande à appliquer au processus, de manière à aboutir à un compromis satisfaisant entre la précision du suivi de trajectoire et la robustesse de la commande.

Chapitre V :

Mise en oeuvre sur la machine TALC

1. Description du processus

La démarche suivie pour la mise en oeuvre de la loi de commande adaptative sur cette maquette de test a consistée dans un premier temps à modéliser et à simuler le processus, puis à valider la loi de commande en effectuant des essais réels [CAN 97b].

La machine TALC dont une vue d'ensemble simplifiée figure ci-dessous est constituée d'une plate forme à 5 axes de rotation agissant sur l'orientation d'un miroir, sur lequel se réfléchit un rayon laser, qui est alors projeté sur un écran. L'instrumentation composée entre autres de la caméra, des calculateurs et des capteurs fournit les coordonnées du spot lumineux, qui doit suivre une trajectoire de référence, ou atteindre un point particulier de l'écran.

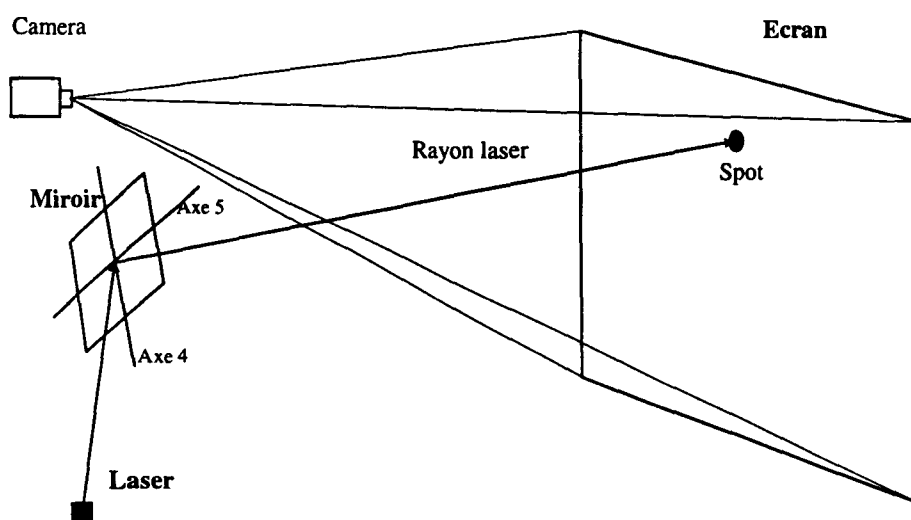


Figure 5.1 : Vue d'ensemble de la machine TALC

Le contrôle s'effectue en agissant sur les moteurs des axes de commande numérotés 4 et 5. Les trois premiers axes sont utilisés pour générer des perturbations. Les valeurs des angles autour des cinq axes sont supposées inconnues pour le contrôleur, qui doit donc calculer les commandes à appliquer

aux moteurs 4 et 5 seulement à partir des coordonnées du spot et de sorte que celui-ci suive la trajectoire imposée, en dépit des perturbations.

1.1. Modélisation de la partie optique

1.1.1. Calcul de l'orientation du miroir

L'orientation du miroir dépend des 5 angles de rotation par rapport aux 5 axes. Ces derniers concourent vers le point O qui est choisi comme l'origine du repère absolu R de base orthonormale (O,x,y,z) définie telle que le miroir dans sa position initiale (angles nuls) soit dans le plan vertical (Oz) et sur la première bissectrice entre les vecteurs Ox et Oy . Le repère pour cette position du miroir est le repère R_0 de base (O,x_0,y_0,z_0) représenté sur la figure 5.2.

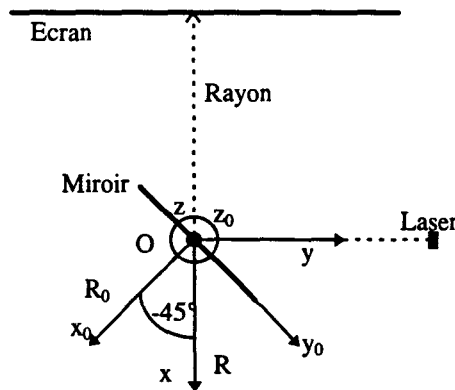


Figure 5.2 : Définition des repères R et R_0

En se basant sur les définitions des différents repères et d'après la conception de la machine, dont une vue détaillée est montrée sur la figure 5.3, la première rotation θ_1 s'effectue autour de l'axe z_0 . Le miroir subit donc une rotation de θ_1 , qui se traduit par l'application dans le repère R_0 d'une matrice de rotation R_{axe1} de la forme :

$$R_{axe1} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}_{R_0} \quad (5.1)$$

L'indice R_0 signifie que la rotation a lieu par rapport au repère R_0 . On définit également un nouveau repère R_1 de base (O,x_1,y_1,z_1) où z_1 est confondu avec l'axe de rotation z_0 . Les axes x_1 et y_1 correspondent à l'image des vecteurs x_0 et y_0 après la rotation θ_1 . Les changements de base du repère R_0 vers R_1 , et de R_1 vers R_0 s'obtiennent en appliquant respectivement les matrices R_{axe1}^{-1} ou R_{axe1} aux vecteurs voulus.

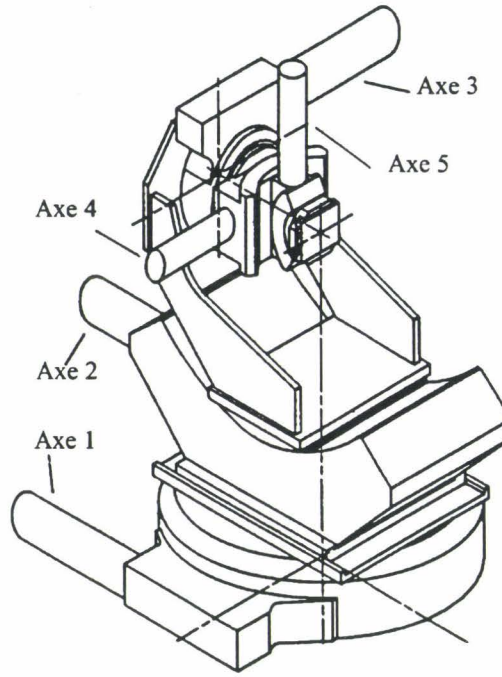


Figure 5.3 : Vue détaillée de la plate-forme TALC.

La deuxième rotation θ_2 s'effectue par rapport à l'axe y_1 ce qui donne dans le repère R_1 la matrice de rotation R_{axe2} :

$$R_{axe2} = \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) \\ 0 & 1 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) \end{bmatrix}_{R_1} \quad (5.2)$$

La base du repère R_2 est alors (O, x_2, y_2, z_2) où z_1 et z_2 sont confondus. Les matrices de changement de base entre les repères R_1 et R_2 ou R_2 et R_1 sont respectivement R_{axe2}^{-1} ou R_{axe2} . En procédant de manière similaire pour les rotations 3, 4 et 5 dont les axes de rotation sont respectivement les axes x_2, z_3, y_4 dans les repères R_2, R_3 et R_4 , on aboutit aux matrices de rotation R_{axe3} , R_{axe4} et R_{axe5} :

$$R_{axe3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_3) & -\sin(\theta_3) \\ 0 & \sin(\theta_3) & \cos(\theta_3) \end{bmatrix}_{R_2} \quad (5.3)$$

$$R_{axe4} = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 \\ 0 & 0 & 1 \end{bmatrix}_{R_3} \quad (5.4)$$

$$R_{axe5} = \begin{bmatrix} \cos(\theta_5) & 0 & \sin(\theta_5) \\ 0 & 1 & 0 \\ -\sin(\theta_5) & 0 & \cos(\theta_5) \end{bmatrix}_{R_4} \quad (5.5)$$

Dans l'état initial avec les cinq angles nuls, les repères R_0, R_1, R_2, R_3, R_4 et R_5 sont confondus, et le miroir est défini à l'aide de deux vecteurs directeurs, le premier suivant l'axe y_0 et le second sur l'axe z_0 , formant une matrice M_{ini} qui vaut :

$$M_{ini} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.6)$$

La composition des 5 rotations et changements de base permet d'obtenir les vecteurs directeurs M_0 du miroir dans le repère R_0 en fonction des angles $\theta_1, \theta_2, \theta_3, \theta_4$ et θ_5 .

$$M_0 = R_{axe1} R_{axe2} R_{axe3} R_{axe4} R_{axe5} M_{ini} \quad (5.7)$$

Pour obtenir la position du miroir dans le repère absolu R , un dernier changement de repère est nécessaire. Le passage de R_0 à R correspond à une rotation de -45° ($-\pi/4$) par rapport à l'axe z_0 , définit par la matrice R_{axe0} suivante :

$$R_{axe0} = \begin{bmatrix} \cos\left(\frac{-\pi}{4}\right) & -\sin\left(\frac{-\pi}{4}\right) & 0 \\ \sin\left(\frac{-\pi}{4}\right) & \cos\left(\frac{-\pi}{4}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.8)$$

La matrice M des vecteurs directeurs du miroir dans le repère R vaut alors :

$$M = R_{axe0} R_{axe1} R_{axe2} R_{axe3} R_{axe4} R_{axe5} M_{ini} \quad (5.9)$$

Connaissant l'orientation du miroir, le calcul de la position du spot peut s'effectuer.

1.1.2. Calcul de la position du spot

Le rayon incident provenant de la source laser est réfléchi au point O, et a comme vecteur directeur dans le repère absolu le vecteur V_i :

$$V_i = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad (5.10)$$

La réflexion sur un miroir se déduit à partir du calcul de la matrice de réflexion H ou matrice de Householder [GOL 89] qui est telle que :

$$H = 2M(M^T M)^{-1} M^T - I \quad (5.11)$$

d'où le vecteur réfléchi V_r :

$$V_r = (2M(M^T M)^{-1} M^T - I)V_i \quad (5.12)$$

avec M la matrice des vecteurs directeurs du miroir dans le repère R , et I la matrice identité.

Ce rayon réfléchi est projeté sur l'écran qui est un plan avec comme vecteurs directeurs (Oy, Oz) formant une matrice notée M_E . De plus ce plan passe par le point P_E de l'écran situé sur l'axe x à une distance d du point O . Le plan de l'écran est donc défini par le point P_E et les deux vecteurs M_E .

$$M_E = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ et } P_E = \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} \quad (5.13)$$

Par définition, le point de projection P_P appartient à la fois au plan de l'écran et à la droite engendré par le rayon réfléchi, ce qui conduit au système ci-dessous à résoudre en fonction des inconnues α , β et γ :

$$\begin{cases} P_P = V_r \alpha \\ P_P = P_E + M_E \begin{bmatrix} \beta \\ \gamma \end{bmatrix} \end{cases} \text{ d'où } \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = [V_r \quad -M_E]^{-1} P_E \quad (5.14)$$

Dans le cas général, après détermination de α , β et γ , les coordonnées se déduisent à partir de la deuxième équation. Avec la disposition particulière de l'écran, les coordonnées du spot se déduisent plus simplement à partir des relations suivantes :

$$\alpha = \frac{d}{V_{r,x}}, P_P = \begin{bmatrix} d \\ \alpha V_{r,y} \\ \alpha V_{r,z} \end{bmatrix} \text{ avec } V_r = \begin{bmatrix} V_{r,x} \\ V_{r,y} \\ V_{r,z} \end{bmatrix} \quad (5.15)$$

Le modèle mathématique est obtenu en supposant que le rayon incident est réfléchi exactement au point d'origine O et en négligeant les éventuels défauts ou imprécisions sur les vecteurs directeurs du rayon et de l'écran de projection. La comparaison entre le modèle et les mesures effectuées sur la machine corroborent les résultats et les hypothèses posées.

Pour parachever la modélisation de la partie optique, la numérisation de l'écran par la caméra doit être prise en compte. La caméra filme l'écran de face et couvre un champ de $v \times v \text{ cm}^2$ avec une résolution de 256 points sur 256. L'image résultante est centrée autour du point P_E et est obtenue dans un repère noté $(O_{ima}, x_{ima}, y_{ima})$ défini par rapport au repère absolu de la manière suivante :

$$x_{ima} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \text{ et } y_{ima} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}. \quad (5.16)$$

Les coordonnées du spot P_{ima} dans le repère de l'image deviennent :

$$P_{ima} = \frac{256}{v} \begin{bmatrix} P_{P,y} + v/2 \\ -P_{P,z} + v/2 \end{bmatrix} \quad (5.17)$$

avec $P_{P,y}$ et $P_{P,z}$ les coordonnées du spot dans le repère absolu.

1.2. Modélisation de la partie mécanique

Pour obtenir un modèle de la partie mécanique, l'approche expérimentale est préférable. Les caractéristiques des moteurs, les couples fournis, et l'inertie des réducteurs et des pièces mécaniques complexes ne sont pas connues précisément. De plus, un niveau de description aussi détaillé apparaît superflu par rapport à nos objectifs.

1.2.1. Détermination expérimentale des zones inactives

Lors des essais d'identification, d'importantes zones inactives sur la commande des moteurs sont observées. Pour vaincre les frottements, la commande à appliquer sur chacun des moteurs doit être supérieure à un certain seuil que l'on détermine à partir de relevés expérimentaux. Chacun des axes est étudié séparément en lui appliquant une commande sinusoïdale pour détecter les seuils ou en créneaux pour déterminer la dynamique des moteurs.

Pour une commande du moteur 4 comprise entre -0.041 V et 0.041 V, la rotation de l'axe 4 cesse et les coordonnées du spot restent constantes (Figure 5.4).

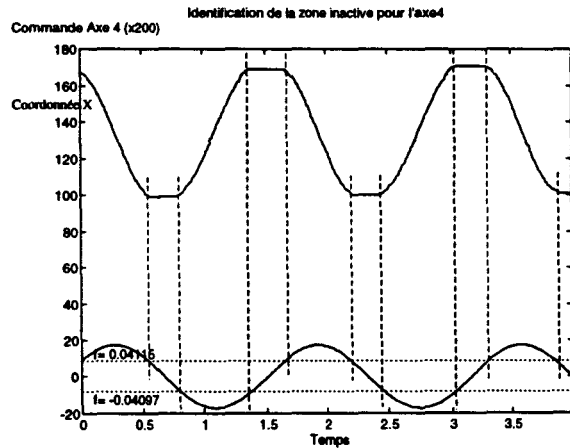


Figure 5.4 : Identification de la zone inactive pour l'axe 4

Le seuil f_4 de la zone inactive est donc de 0.041 V.

Pour le moteur de l'axe 5, la zone inactive représentée sur la figure 5.5 est comprise entre -0.10 V et 0.12 V. Pour cet axe la zone inactive n'est donc pas symétrique et explique la décroissance de

l'ordonnée Y. Le seuil f_5 peut cependant être choisi égal à la moyenne en valeur absolue des seuils négatif et positif d'où : $f_5 = 0.11$ V.

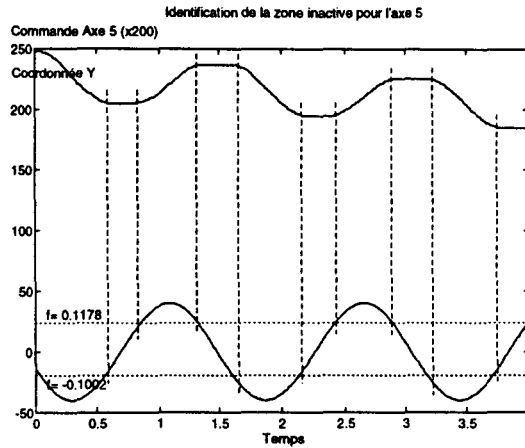


Figure 5.5 : Identification de la zone inactive pour l'axe 5

Les simulations et essais réels présentés ci-après avec la méthode adaptative d'identification de ces seuils donne des résultats concordant avec ceux définis graphiquement. Le modèle utilisé (1.66) et l'algorithme d'adaptation défini pour commander ce processus se révèlent alors satisfaisants pour résoudre les difficultés causées par les zones inactives.

1.2.2. Détermination expérimentale de la dynamique des moteurs

L'ensemble moteur et réducteur pour chaque axe peut être modélisé par une fonction de transfert du deuxième ordre avec un pôle à l'origine qui correspond à l'intégrateur entre la vitesse de rotation du moteur et sa position. Il ne reste alors à déterminer qu'une constante de temps et le gain statique. De plus l'identification n'est réalisée que pour les axes 4 et 5 puisque ce sont ces axes qui importent le plus et qui sont utilisés pour contrôler la position du miroir et du spot.

A partir de relevés de mesures et en comparant les réponses obtenues avec le modèle et par l'expérimentation, il est possible de trouver une approximation du gain statique et des caractéristiques dynamiques des axes 4 et 5. Les estimations des fonctions de transfert pour ces axes sont respectivement :

$$\frac{Y_4}{U_4} = \frac{3}{s(0.01s + 1)} \text{ et } \frac{Y_5}{U_5} = \frac{-1.4}{s(0.01s + 1)} \quad (5.18)$$

La modélisation est alors très simple mais permet néanmoins d'obtenir un modèle satisfaisant de la partie mécanique.

1.2.3. Réduction de l'influence du bruit de mesure

Les mesures sont arrondies à cause de la résolution de la caméra, et de la numérisation des images. De plus la forme du spot n'est pas parfaitement circulaire et forme après numérisation un groupe de quelques pixels qui peut devenir légèrement ovale suivant les angles de réflexion et de projection, et suivant la position du point sur l'écran. Les mesures sont également sensibles aux conditions d'utilisation et à la luminosité ambiante. Les coordonnées du spot sont donc précises seulement à un ou deux pixels près. Pour limiter l'influence du bruit de mesure sur la commande et éviter des oscillations non désirées, il est possible quand l'erreur de position du spot est inférieure à quelques pixels, et quand la consigne est constante, de limiter les variations de la commande en les multipliant par un coefficient α inférieur à 1 qui est défini ci-dessous.

Soient Y_p la précision désirée, $\|\Delta Y_d\|$ la norme des variations de la consigne, Y_d et Y les vecteurs de sortie désirée et réelle, alors α est défini par :

$$\alpha = \left(\frac{\|Y_d - Y\| + \|\Delta Y_d\|}{Y_p} \right)^{\frac{1}{2}} \text{ pour } \|Y_d - Y\| + \|\Delta Y_d\| \leq Y_p \quad (5.19)$$

Quand le spot se situe dans un rayon inférieur à Y_p pixels par rapport à la consigne, les variations de commande appliquées au système sont alors égales à $\alpha \delta u$ au lieu de δu . La précision Y_p désirée est pour la suite des essais fixée à 4 pixels.

2. Etude du contrôleur et du processus en simulation

Les objectifs de la simulation sont d'étudier le comportement du processus puis de valider la loi de commande. Ces simulations permettent également d'optimiser les paramètres du contrôleur et d'analyser leur influence sur la précision et la robustesse. Lors des simulations, les calculs supplémentaires de validation peuvent être réalisés alors que les contraintes liées au temps réel ne permettent pas d'effectuer complètement certains tests en ligne.

2.1. Définition du modèle et estimation des zones inactives

Les premiers essais portent sur l'estimation des zones inactives afin de vérifier que les seuils déduits lors de l'adaptation du modèle convergent vers les valeurs désirées f_4 et f_5 obtenues expérimentalement et utilisées pour la simulation. L'adaptation du modèle dont le nombre de régressions de chaque variable est égal à 2 ($nbry = 2$ et $nbru = 2$), s'effectue à une période de 0.01s avec 200 points mémorisés dans la fenêtre d'observation. Le seuil d'insertion d'un nouveau point σ est de 4 pixels. De plus, un filtrage est appliqué aux données pour limiter les problèmes engendrés

par le bruit et les erreurs d'arrondi sur les mesures des coordonnées du spot. Le filtre est alors défini pour calculer la moyenne sur 4 mesures ($nbf = 4$).

Avec ces paramètres, les valeurs des seuils des zones inactives représentées sur la figure 5.6 varient autour des seuils réels 0.11 et 0.041 avec des erreurs de l'ordre de 20% qui sont dues aux non-linéarités de la partie optique du processus.

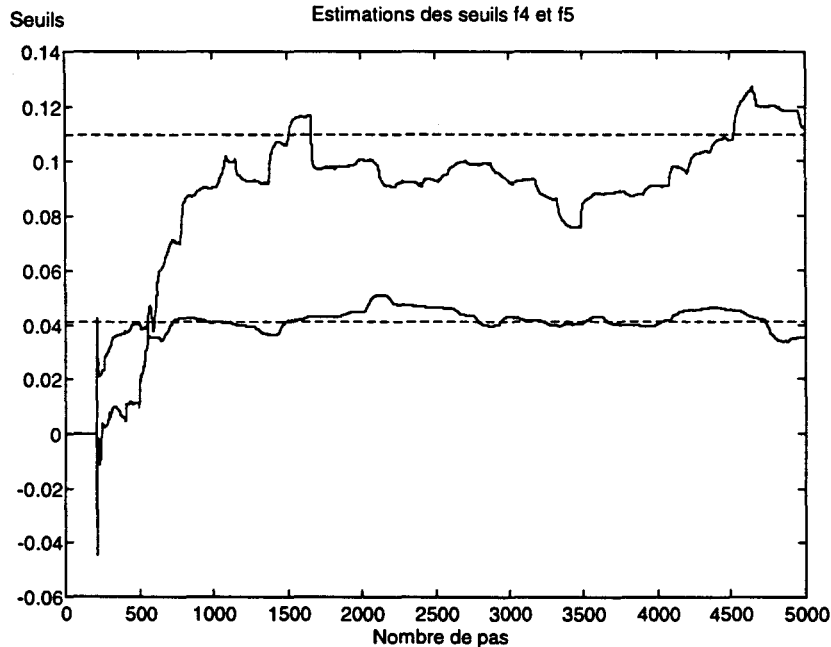


Figure 5.6 : Estimations des seuils des zones inactives

La précision de ces estimations s'avère suffisante pour le contrôle du processus même si ces erreurs dégradent légèrement les performances du contrôleur et ont pour conséquence d'accroître l'amplitude des variations de la commande et d'augmenter le nombre de franchissement de la zone inactive.

2.2. Définition du contrôleur et simulation sans perturbation

La loi de commande est établie pour minimiser les erreurs sur 8 prédictions avec le critère (4.30) imposant le taux de décroissance des erreurs (Chapitre IV.5.). La matrice d'annulation des erreurs est choisie telle que :

$$e_{k+1} = \begin{bmatrix} 0.9 & \\ & 0.9 \end{bmatrix} e_k \quad (5.20)$$

La trajectoire imposée au spot est cyclique et de période 6s. Elle est définie de la manière suivante :

$$\begin{aligned} x_d &= 80 \cos\left(\frac{2\pi}{3}t\right) \\ y_d &= 80 \cos\left(\frac{2\pi}{6}t\right) \end{aligned} \quad (5.21)$$

Le contrôleur permet alors d'obtenir un suivi de trajectoire précis. La moyenne de la norme des erreurs est inférieure à 2 pixels et l'erreur maximale ne dépasse pas 4 pixels. La trajectoire du spot et les commandes appliquées au processus sont représentées sur la figure 5.7.

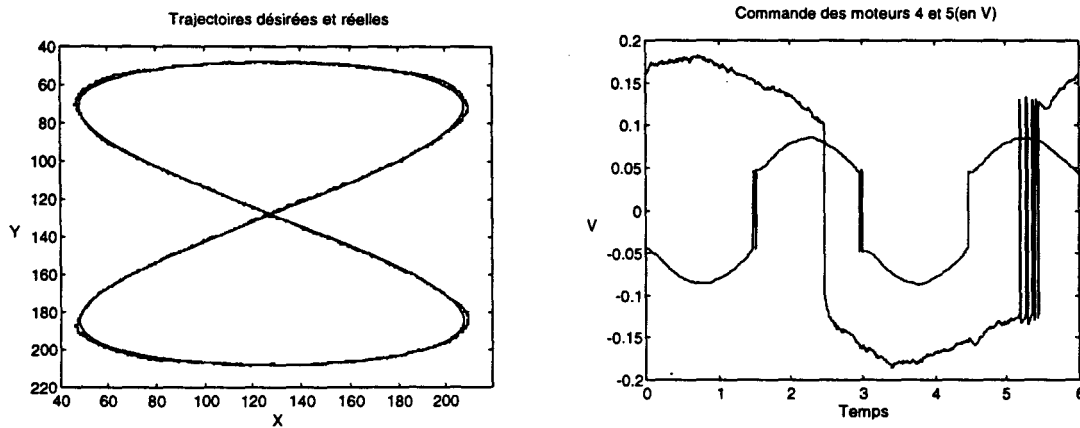


Figure 5.7 : Trajectoires désirées et réelles du spot et commande des axes 4 et 5

On observe sur ces courbes entre la cinquième et sixième seconde, une succession de franchissements de la zone inactive qui sont dus à une surévaluation du seuil f_5 . A partir des nouvelles mesures obtenues, ce seuil est alors réévalué et corrigé.

2.3. Influence des perturbations

Des perturbations sinusoïdales sont générées et appliquées au processus directement en modifiant les angles des axes 1, 2 et 3. Pour la simulation réalisée ci-dessous, les angles des perturbations évoluent suivant les lois :

$$\begin{aligned}
 \theta_1 &= 0.1 \cos(0.8t) \\
 \theta_2 &= 0.15 \cos(0.7t) \\
 \theta_3 &= 0.2 \cos(0.9t)
 \end{aligned}
 \tag{5.22}$$

Ces perturbations provoquent des rotations maximales de 6 à 12 degrés du miroir suivant les trois premiers axes. En désactivant le contrôleur, les variations de ces trois angles déplacent le spot sur la totalité de l'écran. La trajectoire décrite par le spot sur 20 secondes en appliquant seulement les perturbations est montrée sur la figure 5.8. La trajectoire désirée est également tracée pour montrer l'influence des perturbations par rapport à la consigne.

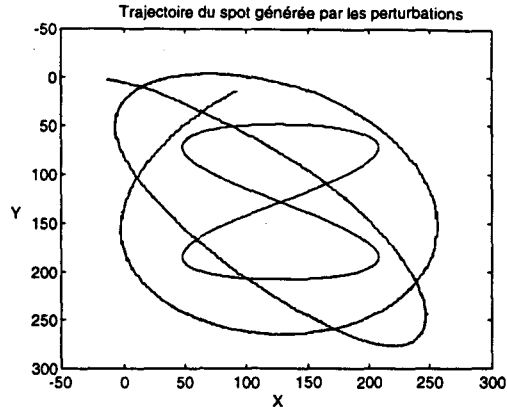


Figure 5.8 : Influence des perturbations sur la trajectoire du spot

Avec le contrôleur activé (Figure 5.9), la précision moyenne du suivi de trajectoire est alors de 3 pixels, et l'erreur maximale est inférieure à 10 pixels.

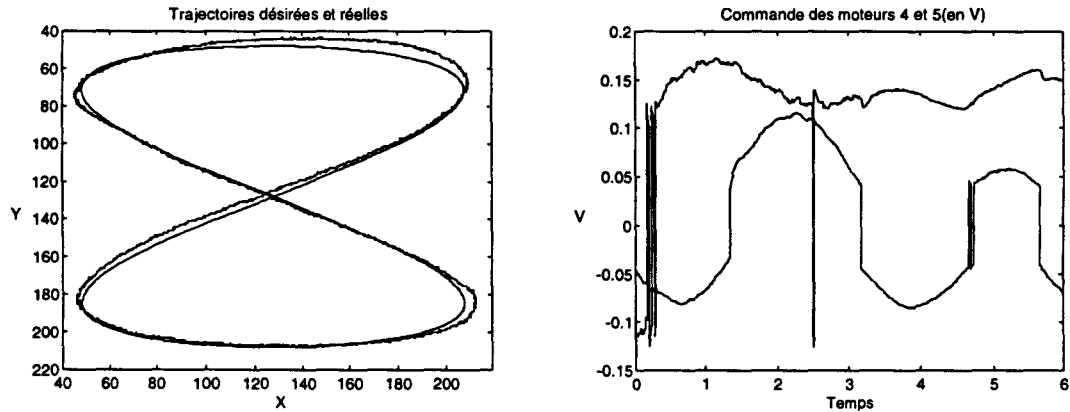


Figure 5.9 : Trajectoires du spot et commande avec perturbations des axes 1, 2 et 3

Les résultats obtenus en simulation sont concluants. Il est alors possible d'entreprendre des essais réels sur la machine TALC en utilisant les paramètres et réglages du contrôleur défini pour le modèle du processus.

3. Etude expérimentale

3.1. Essai sans perturbation

La trajectoire désirée pour cet essai est définie par les fonctions :

$$\begin{aligned} x_d &= 60 \cos\left(\frac{2\pi}{3}t\right) \\ y_d &= 60 \cos\left(\frac{2\pi}{6}t\right) \end{aligned} \tag{5.23}$$

Le suivi de trajectoire et les commandes obtenues sans perturbation pour un cycle de 6 secondes sont alors les suivants :

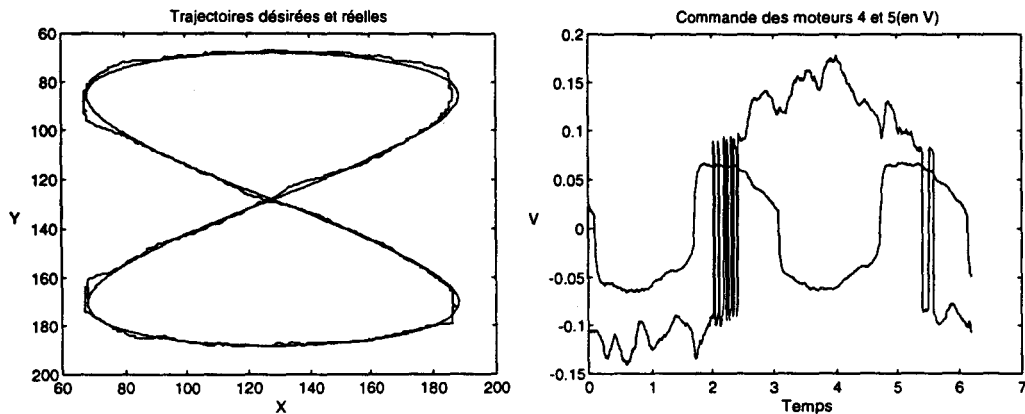


Figure 5.10 : Commande du processus sans perturbation.

La moyenne et la valeur maximale des normes des erreurs valent respectivement 1.8 et 8.7 pixels. Le suivi de trajectoire est donc moins précis qu'en simulation, mais est acceptable par rapport à la précision des mesures et à la complexité du processus. Deux autres essais sont réalisés en imposant respectivement les coordonnées x puis y du spot à suivre des consignes en créneaux.

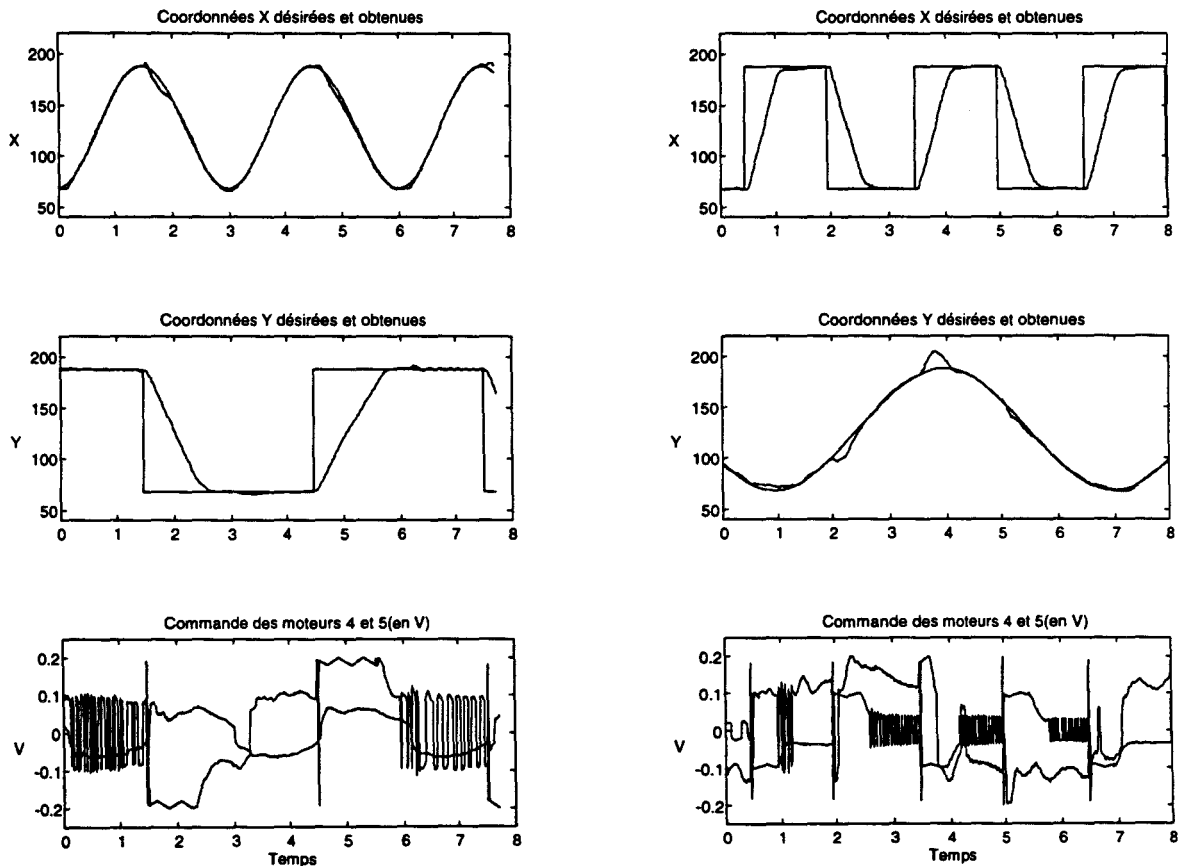


Figure 5.11 : Commande du processus pour une consigne x et y évoluant en créneaux.

Ces essais mettent en évidence les problèmes survenant sur la commande quand une des consignes est constante. En effet, pour corriger les erreurs, la commande de l'axe 5 dans le premier cas et 4 dans le second cas est obligée de franchir la zone inactive à une fréquence élevée, ce qui entraîne des sollicitations importantes des actionneurs et une dégradation de la précision du suivi de trajectoire.

3.2. Influence des perturbations

Des perturbations sont générées par l'intermédiaire des axes 1, 2 et 3 en agissant sur la commande de ces moteurs. Trois essais sont alors effectués pour observer l'influence de perturbations introduites en modifiant les angles de rotation pendant quelques secondes entre -5 et 5 degrés.

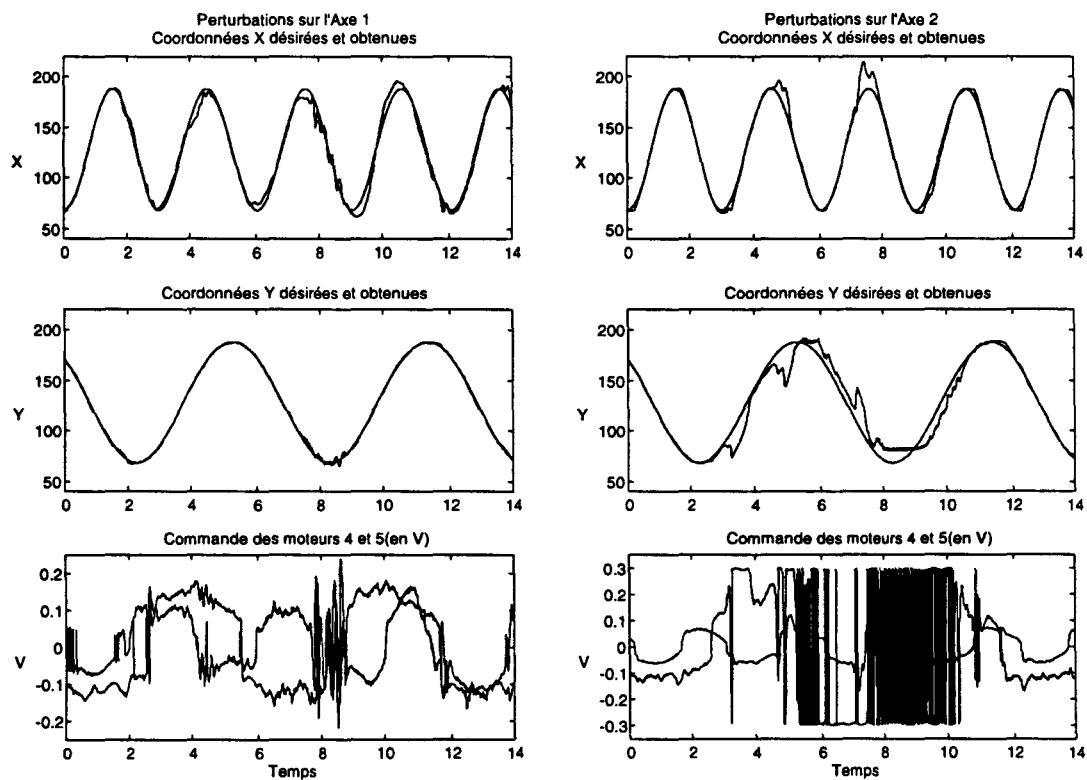


Figure 5.12 : Perturbations générées en agissant sur les axes 1 et 2

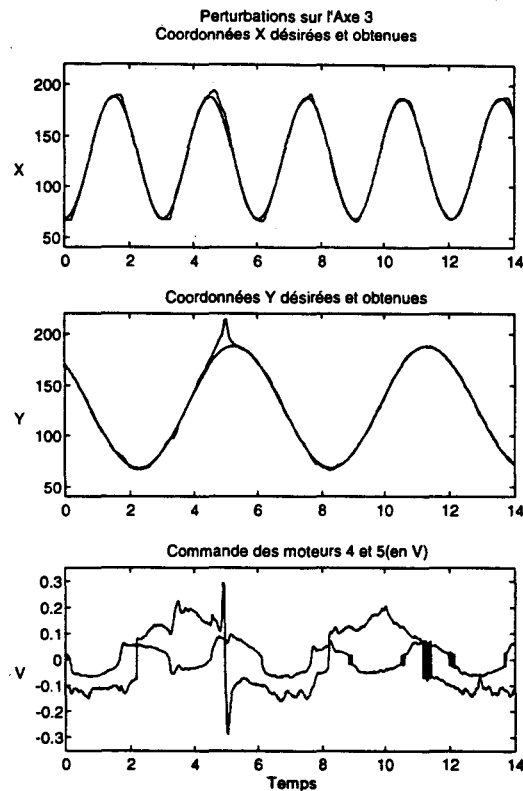


Figure 5.13 : Perturbations générées en agissant sur l'axe 3

Les perturbations sur l'axe 2 sont les plus gênantes à cause de l'importante zone inactive de l'axe 5 dont l'évaluation du seuil se trouve faussée. La commande calculée présente en conséquence des oscillations et variations de fortes amplitudes ce qui dégrade la précision. Le spot peut également lors de brusques perturbations sortir du champ de vision de la caméra et bloquer tout le système de commande. Quand cette situation extrême est évitée, et lorsque les perturbations cessent, le modèle et le contrôleur se réadaptent correctement et donnent à nouveau une commande satisfaisante.

4. Conclusion

L'expérimentation d'une loi de commande sur un processus tel que la machine TALC, où les perturbations sont importantes, où les non linéarités ne peuvent pas être négligées, et où les bruits de mesure ont une amplitude relativement élevée, constitue une validation des algorithmes d'identification et une preuve expérimentale de la robustesse du contrôleur. Le modèle qui est défini pour identifier les seuils de la zone inactive des actionneurs donne une bonne approximation de la partie mécanique du processus. La loi de commande adaptative et prédictive basée sur ce modèle est donc pertinente et efficace pour le contrôle de processus complexes.

Conclusion générale

Les objectifs de ces travaux de recherche ont été d'établir des lois de commande pour des processus complexes perturbés, variant dans le temps et présentant des non-linéarités. L'approche retenue qui est basée sur l'identification en ligne du processus et l'adaptation permanente d'un modèle permet de prendre en considération les variations du comportement et des caractéristiques du processus, mais aussi de limiter l'influence des perturbations et des non-linéarités. En plus de l'identification du processus, des algorithmes de filtrage et de surveillance sont proposés pour améliorer la précision du modèle lorsque du bruit est présent sur les mesures et pour valider le modèle obtenu et vérifier la cohérence des résultats.

Les algorithmes d'adaptation, de calcul des prédictions et de détermination de la commande ont été établis pour réduire dans la mesure du possible les temps de calcul, et pour offrir une grande robustesse numérique vis-à-vis des bruits et des éventuels problèmes émanant de matrices singulières ou mal conditionnées, qui peuvent provoquer une divergence irrémédiable de la loi de commande. Le choix approprié des algorithmes contribue donc à la robustesse du contrôleur et de la méthode d'identification. La mise en oeuvre et la validation expérimentale de la loi de commande sur la machine de test TALC montrent que les problèmes concrets liés au calcul numérique ou dus à la précision des mesures sont résolus et confirment le bien-fondé de l'approche adaptative et prédictive proposée.

Le compromis entre la précision du suivi de trajectoire, la rapidité d'annulation des erreurs et la stabilité de la loi de commande est résolu à l'aide de critères minimisant les erreurs sur un horizon de prédiction de plusieurs pas avec une dynamique définie en fonction de la précision et des performances désirées. Ce principe de commande est très général et utilisable pour de nombreux processus multivariables, comportant des pôles instables, à non minimum de phase ou encore pour des systèmes à retards purs.

Les perspectives sont d'étendre cette approche à des systèmes plus complexes requérant l'utilisation de modèles dont la structure est établie spécialement en fonction du processus à contrôler, afin d'obtenir un modèle plus proche du processus réel et d'améliorer le calcul des prédictions pour aboutir à une loi de commande plus précise et plus robuste.

Bibliographie

[AMI 97] J. Amin, B. Friedland, A. Harnoy "Implementation of a Friction Estimation and Compensation Technique", IEEE Control Systems, pp 71-75, August 1997.

[AST 89] K. J. Astrom, B. Wittenmark, "Adaptive Control", Addison-Wesley, Reading, Massachusetts 1989.

[AUS 95] A. Aussem, "Théorie et Applications des Réseaux de Neurones Récurents et Dynamiques à la prédiction, à la modélisation et au contrôle adaptatif des Processus Dynamiques", Université R. Descartes Paris V, Paris 1995.

[BIS 92] R. H. Bishop, "Adaptive Control of Space Station with Control Moment Gyros", IEEE Control Systems, Vol. 12, No 5, October 1992.

[BIT 90] S. Bittanti, P. Bolzern, M. Campi, "Recursive Least-Squares Identification Algorithms with Incomplete Excitation : Convergence Analysis and Application to Adaptive Control", IEEE Transactions on Automatic Control, Vol. 35, No 12, December 1990.

[BLA 91] G. Blanchet, J. Prado, "Méthodes numériques pour le traitement du signal", Masson - Paris, 1991.

[BOL 92] F. Bolourchi and R. A. Hess, "Nonlinear Model Reference Adaptive Control using Tap-Delay Filters", IEEE Transactions on Systems Man and Cybernetics, Vol. 22, No. 2, March/April 1992.

[BOR 90] P. Borne, G. Dauphin-Tanguy, J.P. Richard, F. Rotella, I. Zambettakis, "Commande et optimisation des processus", Editions Technip - Paris, 1990.

[BOR 92] P. Borne, G. Dauphin-Tanguy, J.P. Richard, F. Rotella, I. Zambettakis, "Modélisation et identification des processus" (Tome 2), Editions Technip - Paris, 1992.

[BOUANI 96a] F. Bouani, R Ben Abdennour, M. Ksouri, G. Favier, "A matrix formulation for the multivariable predictor with simple recursivity", Proceedings of the World Automation Congress - WAC 96, Vol 4, pp 81-88, Montpellier France, May 28-29, 1996.

[BOUANI 96b] F. Bouani, R. Ben Abdennour, M. Ksouri, "La commande prédictive généralisée : validation d'une base de règles sur un procédé réel", Conférence Maghrébine d'Automatique, d'Electrotechnique et d'Electronique Industrielle (COMAEI 96), Vol 1 pp 304-309 ,Tiemcen, 3-5 Décembre 1996

[BOU 96] P. Boucher, D. Dumur, "La commande prédictive", Editions Technip - Paris, 1996.

[CAN 97a] R. Canart, P. Borne, "Implementation of Adaptive Hyperplanes for the Determination of Robust Control", Journal of Intelligent and Robotic Systems, Vol 18, No 3, March 1997.

[CAN 97b] R. Canart, P. Borne, "Design of Robust Adaptive Controller", Studies in Informatics and Control, Vol 6, No 3, September 1997.

[COR 96] P.I. Corke, "A Robotics Toolbox for MATLAB" , IEEE Robotics and Automation Magazine, Vol.3 No.1 , pp 24-32, March 1996.

[DOR 95] R. C. Dorf and R. H. Bishop, "Modern Control Systems", Addison-Wesley Publishing, 1995.

[GOL 89] Gene H. Golub Charles F. Van Loan. "Matrix Computations", Johns Hopkins University Press, Baltimore, 1989

[KOS 95] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou and P. A. Ioannou, "High-Order Neural Network Structures for Identification of Dynamical Systems", IEEE Transactions on Neural Networks, Vol. 6, No 2, March 1995.

[LAN 93] I.D. Landau, "Identification et commande des systèmes", Hermès, Paris, 1993

[LON 95] R. Longchamp, "Commande numérique de systèmes dynamiques", Presses polytechniques et universitaires romandes, 1995.

[MEY 93] P. Meyn, J. Brown, "Model Reference Adaptive Control of Time Varying and Stochastic Systems", IEEE Transactions on Automatic Control, Vol. 38, No 12, December 1993.

[MSA 92] F. Msahli, R. Ben Abdennour, M. Ksouri, "Identification structurelle et paramétrique d'un refroidisseur à grilles de clinker d'une cimenterie", proceedings ICEA 92 vol 3, pp 58 -68 , Tizi Ouzou - Algérie, May 04-06, 1992.

[NAR 90] K.S. Narendra, K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transactions on Neural Networks Vol.1. No. 1. March 90.

[OHB 96] M. Ohbayashi, K. Hirasawa, M. Hashimoto and J. Murata, "Robust using Second Order Derivative of Universal Learning Network", IEEE International Conference on Systems Man and Cybernetics, vol 2, pp1184-1189, October 1996.

[ORT 89] R.Ortega, Yu Tang, "Robustness of Adaptive Controllers - a survey", Automatica, Vol 25, No 5, pp. 651-677, 1989.

[PAT 95] D.W. Patterson, "Artificial neural networks : theory and applications", Prentice Hall, New York, 1995.

[POP 94] M. R. Popovic, A. A. Goldenberg, K. B. Shimoga, R. C. Hui, "Model-Based Compensation of Friction in Direct-Drive Robotic Arms", Studies in Informatics and Control, Vol 3, No 1, March 1994.

[POP 96] T.D. Popescu, "Comparative Study of some Sequential Detection Methods for Digital Signals", Studies in Informatics and Control, Vol 5, No 4, December 1996.

[POV 75] L. Povy, "Identification des processus", Dunod université, 1975.

[RIC 91] J. Richalet, "Pratique de l'identification", Hermès, Paris, 1991.

[ROV 94] G.A. Rovithakis, M.A. Christodoulou, "Adaptive Control of Unknown Plants Using Dynamical Neural Networks", IEEE Transactions on Systems Man and Cybernetics, Vol.24 No.3 March 1994.

[ROV 95] G.A. Rovithakis and M.A. Christodoulou, "Direct Adaptive Regulation of Unknown Nonlinear Systems via Dynamic Neural Networks", IEEE Transactions on Systems Man and Cybernetics, Vol. 25, No.12, pp1578-1594, Dec.1995.

[SAD 95] N. Sadegh, "A Nodal Link Perceptron Network with Applications to Control of a Nonholonomic System", IEEE Transactions on neural networks, Vol.6, No.6, November 1995

[SAN 92] R.M. Sanner and J.J. E. Slotine, "Gaussian Networks for Direct Adaptive Control", IEEE Transactions on neural networks, Vol.3, No.6, November 1992

[SUD 96] W. Sudarno, "Amélioration des méthodes d'identification de type moindres carrés appliquées à la commande adaptative et à la reconnaissance de formes", Thèse, Université Paul Sabatier de Toulouse, 1996.

