

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MENTOURI DE CONSTANTINE
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

N° d'ordre :

Série :

THESE

Présentée pour obtenir le diplôme de Doctorat d'Etat
En Electronique

OPTION

Contrôle

PAR

AHMIDA Zahir

THEME

**Contribution à la Commande Prédictive :
Stabilisation Prédictive Non Linéaire d'une Commande en Poursuite Basée
sur une Classe de Réseaux Neuroniques**

SOUTENUE LE 15/05/2006

Devant le Jury :

Président	: Dr BENNIA Abdelhak	Professeur, U.M. de Constantine
Rapporteur	: Dr CHAREF Abdelfatah	Professeur, U.M. de Constantine
Examineurs	: Dr MANSOURI Nora	Professeur, U.M. de Constantine
	Dr ABBASSI Hadj Ahmed	Professeur, U.B.M. de Annaba
	Dr FERDI Youcef	Maître de Conférences, U. 20 Août 55 de Skikda

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MENTOURI DE CONSTANTINE
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

N° d'ordre :

Série :

THESE

Présentée pour obtenir le diplôme de Doctorat d'Etat
En Electronique

OPTION

Contrôle

PAR

AHMIDA Zahir

THEME

**Contribution à la Commande Prédictive :
Stabilisation Prédictive Non Linéaire d'une Commande en Poursuite Basée
sur une Classe de Réseaux Neuroniques**

SOUTENUE LE 15/05/2006

Devant le Jury :

Président	: Dr BENNIA Abdelhak	Professeur, U.M. de Constantine
Rapporteur	: Dr CHAREF Abdelfatah	Professeur, U.M. de Constantine
Examineurs	: Dr MANSOURI Nora	Professeur, U.M. de Constantine
	Dr ABBASSI Hadj Ahmed	Professeur, U.B.M. de Annaba
	Dr FERDI Youcef	Maître de Conférences, U. 20 Août 55 de Skikda

Dédicaces

A mes parents , pour tout ce qu'ils ont fait pour moi .

A mon épouse , pour ses sacrifices , ses encouragements et son support indéfectible et sans lesquels ce travail n'aurait pas vu le jour.

A mes enfants : Mérième , Yasser et la petite Maroua.

A mon beau-père ; Ammi Ahmed , pour son héroïsme et pour les batailles qu'il a mené au nom des siens .

Zahir

Remerciements

En premier, je tiens à remercier Le Docteur **CHAREF Abdelfatah**, Professeur à l'Université Mentouri de Constantine, et lui exprimer toute ma gratitude pour avoir encadré et dirigé ce travail. Ses conseils, ses orientations et le soutien que j'ai toujours trouvé auprès de lui et qu'il n'a jamais cessé de me témoigner durant des années, ont permis de terminer ce travail.

Je remercie également le Docteur **BECERRA Victor M.**, Professeur au 'Cybernetics Department' de l'Université de Reading (UK), pour son engagement et, pour avoir mis à ma disposition toute la documentation dont j'avais besoin pour accomplir ce travail.

Je ne saurais remercier assez mon Ami et Collègue, Docteur **GROUCHE Lakhdar** pour son soutien moral et logistique.

J'exprime ma gratitude envers le Docteur **BENNIA Abdelhak**, Professeur à l'Université Mentouri de Constantine de m'avoir fait l'honneur d'accepter de présider le Jury.

J'adresse également mes vifs remerciements aux : Docteur **MANSOURI Nora** ; Professeur à l'Université Mentouri de Constantine, au Docteur **ABBASSI Hadj Ahmed** ; Professeur à l'Université Badji Mokhtar de Annaba et, au Docteur **FERDI Youcef** Maître de Conférences à l'Université 20 Août 55 de Skikda pour avoir accepté de juger mes travaux et de faire partie du Jury. Qu'ils trouvent, ici, l'expression de toute ma gratitude.

Enfin, je remercie tous mes amis et collègues et en particulier, Dr. **GOUTAS Ahcène**, Dr. **DJEZZAR Salah** et Dr. **SELMi Salah** pour leurs encouragements et leur support.

Sommaire

Chapitre 1 : Introduction

- 1.1. La commande prédictive : un aperçu historique
- 1.2. Les problématiques de la commande prédictive
 - 1.2.1 La modélisation et l'identification
 - 1.2.2 La stabilité
 - 1.2.3 La poursuite de trajectoires
 - 1.2.4 La robustesse
 - 1.2.5 L'optimisation
- 1.3. Contributions de la thèse
- 1.4. Organisation du manuscrit

Chapitre 2 : De la régulation linéaire quadratique à la commande prédictive

- 2.1 Stabilité asymptotique des systèmes dynamiques
- 2.2 Le principe d'optimalité et la programmation dynamique
- 2.3 La Régulation Linéaire Quadratique (RLQ)
 - 2.3.1 Position du problème
 - 2.3.2 Solution du problème RLQ
 - 2.3.3 Stabilité de la commande RLQ
- 2.4 La commande prédictive linéaire
 - 2.4.1 Formulation du problème de la commande prédictive linéaire à horizon infini
 - 2.4.2 Solution du problème de la commande prédictive à horizon infini par la méthode de Rawlings-Muske

Chapitre 3 : Stabilisation non linéaire par la commande prédictive

- 3.1 Principe et formulation mathématique de la commande prédictive non linéaire

3.2 Résultats sur la stabilité de la commande prédictive non linéaire

- 3.2.1 Commande prédictive à horizon infini
- 3.2.2 Commande prédictive à horizon fini avec contrainte d'égalité terminale
- 3.2.3 Commande prédictive à horizon fini avec contrainte d'inégalité terminale
- 3.2.4 Commande prédictive à horizon fini avec fonction de coût terminale

Chapitre 4 : Les réseaux de neurones non bouclés et la théorie de l'approximation

- 4.1 Les réseaux de neurones : histoire et concept
- 4.2 Du neurone biologique au neurone artificiel
 - 4.2.1 Morphologie et physiologie du neurone biologique
 - 4.2.2 Modèle du neurone artificiel
- 4.3 Architectures des réseaux de neurones et techniques d'apprentissage
 - 4.3.1 Les perceptrons multicouches
 - 4.3.2 Réseaux à fonction à base radiale (RBF)
- 4.4 L'approximation des fonctions non-linéaires par les réseaux de neurones
 - 4.4.1 Préliminaires mathématiques
 - 4.4.2 Les perceptrons multicouches sont des approximateurs universels
 - 4.4.3 L'approximation des fonctions par les réseaux à fonctions à base radiale

Chapitre 5 : Identification des systèmes dynamiques non linéaires par les réseaux de neurones

- 5.1 Modélisation et identification
 - 5.1.1 Identification des systèmes statiques et dynamiques
 - 5.1.2 Modèle fondamental et modèle empirique
 - 5.1.3 Etapes pratiques de l'identification
- 5.2 Structures des modèles empiriques
 - 5.2.1 Modèles linéaires
 - 5.2.2 Modèles non linéaires

- 5.3 Approche pratique de l'identification non linéaire par les réseaux de neurones RBF
 - 5.3.1 Stratégies d'apprentissage des réseaux GRBF
 - 5.3.2 Algorithme RANEKF modifié à apprentissage aléatoire
 - 5.3.3 Emulation d'un modèle non linéaire dans l'espace d'état par une structure de réseaux RBF bouclée
- 5.4 Identification non linéaire inverse

Chapitre 6 : Commande prédictive stabilisante basée sur une classe de modèles neuroniques à fonctions gaussiennes

- 6.1 Les modèles neuroniques dans la commande prédictive non linéaire
- 6.2 Problématique d'une commande prédictive non linéaire en poursuite de trajectoire dans l'espace d'état
 - 6.2.1 Description du système
 - 6.2.2 Problème optimal de poursuite de trajectoire dans l'espace d'état
- 6.3 Une architecture générale pour une commande prédictive stabilisante référencée modèle GRBF
 - 6.3.1 Génération des déviations des variables d'état pour une classe de systèmes non linéaires commandés en poursuite de trajectoire
 - 6.3.2 Propriétés dynamiques du modèle de l'erreur d'état et reformulation du problème optimal de poursuite de trajectoire dans l'espace d'état
 - 6.3.3 Synthèse d'une loi de commande prédictive stabilisante référencée modèle GRBF
- 6.4 Simulations et résultats

Conclusion

Bibliographie

Chapitre 1

Introduction

1.5. La commande prédictive : un aperçu historique

Le concept de la commande prédictive référencée modèle, connue aussi sous l'appellation de commande à horizon fuyant (Receding Horizon Control or Moving Horizon Control) est apparu au début de la décennie 70 et son application a été, d'abord, initiée et développée par la firme *Shell Oil* [1,2]. Dans l'étude effectuée par Qin et Badgwell en 1997 [2], plus de deux mille (2000) applications de la commande prédictive ont été recensées et couvrent un large champ de l'activité industrielle allant de la chimie vers l'aérospatial en passant par la robotique. Plusieurs entreprises proposent, aujourd'hui, des logiciels de commande prédictive configurables selon les besoins de l'application (MDC, Invensys, AspenTech, Honeywell...).

La stratégie de la commande prédictive repose sur un principe relativement simple pour la détermination d'une loi de commande optimale pour les systèmes linéaires et non-linéaires. C'est un schéma dans lequel le contrôleur détermine un profil de la variable de commande en résolvant à chaque instant, en boucle ouverte, un problème de commande optimale à horizon fini tout en considérant l'état courant du système comme état initial. Cette optimisation résulte en une séquence de commandes optimales dont le premier terme est effectivement appliqué à l'entrée du système et la sortie correspondante est aussitôt mesurée [3]. L'effet de rétroaction (feedback) est incorporé dans ce schéma par l'exploitation de ces mesures dans la mise-à-jour du problème d'optimisation correspondant à l'instant d'échantillonnage suivant.

Ainsi, le principe fondamental d'une stratégie de commande prédictive repose sur une optimisation à répétition et en boucle ouverte d'une fonction de coût à horizon fini s'étendant de l'instant courant vers le futur.

Dans un problème de commande prédictive la fonction de coût est définie selon les besoins et les contraintes opérationnelles de l'application, ce qui représente un avantage certain par rapport aux autres commandes optimales où la fonction de coût standard est définie par l'intégrale (Somme dans le cas des systèmes discrets) du carré de l'erreur entre la sortie du système et la consigne (exemple: Commande Quadratique Linéaire).

Cette flexibilité dans la définition de la fonction de coût ajoutée à la possibilité offerte pour résoudre le problème d'optimisation en tenant compte des contraintes en entrée/sortie ont grandement contribué à la popularité de la commande prédictive à horizon fuyant.

Historiquement, les premières applications de la commande prédictive étaient basées sur des modèles temporels linéaires de type entrée/sortie construits à partir des réponses impulsionnelles ou à des excitations en échelon du procédé/système. C'est ainsi que fut proposée la commande algorithmique (Model Algorithmic Control—MAC) par Richalet et al. [4], qui repose sur la minimisation d'une fonction de coût quadratique en utilisant le modèle impulsionnel d'un système stable en boucle-ouverte. Vint ensuite, la commande à matrice dynamique (Dynamic Matrix Control – DMC) développée par Cutler et Ramaker [5] en 1980 citée dans [6], qui ont adopté une structure similaire à la commande algorithmique mais en utilisant un modèle obtenu par une excitation en échelon.

Parallèlement, une autre méthodologie étroitement liée à la commande prédictive à horizon fuyant fut proposée par des chercheurs dans le domaine de la commande auto-adaptative, il s'agit de la commande auto-adaptative à prédiction étendue (extended prediction self-adaptive control—EPSAC) proposée par De Keyser et Van Cauwenberge [7], et de la commande prédictive généralisée présentée par Clarck et al. [8, 9]. La commande GPC développée autour d'un modèle CARIMA (Controlled Auto-Regressive Integrated Moving Average) était, en fait, une extension de la commande à variance minimale généralisée (generalised minimum variance—GMV) initiée par Aström [10] et était dans sa formulation déterministe très proche de la commande prédictive à horizon fuyant. L'introduction de la commande GPC a inspiré l'apparition de quelques algorithmes de commande optimale tel le LQGPC (Linear Quadratic Gaussian Predictive Control) qui propose une solution polynômiale [11-13], généralement

adoptée pour la commande LQG, à un problème optimal à horizon fini. Tous les schémas de commande sus-cités se caractérisent, dans leur versions originales, par la linéarité et l'empirisme de leur modèles ainsi que par le manque des outils théoriques nécessaires à l'analyse et à la garantie de la stabilité. La stabilité n'était, en fait, obtenue que par un ajustage des paramètres de la fonction de coût et de l'horizon de commande [8, 9]. Ceci représentait un handicap majeur, car la stabilité en boucle-fermée n'était pas imposée par modification et redéfinition du problème de commande optimale. Dans leur travaux sur la commande optimale, Bitmead et al. [14] ont critiqué cette approche en l'assimilant à du 'faire joujou' (playing games), au lieu d'imposer la stabilité en modifiant le problème de la commande optimale. En conséquence, les automaticiens engagés dans la recherche sur la commande prédictive étaient ramenés à redéfinir leur problématique et imposer la stabilité en introduisant des contraintes sur les variables de sortie et de commande en boucle ouverte [15-17]. Une approche qui a grandement influencé l'évolution de la commande prédictive fut sa reformulation dans un cadre d'espace d'état [18-21], ce qui avait permis de bénéficier directement de toute la richesse de la théorie des systèmes linéaires—théorie de la commande quadratique linéaire, stabilité au sens de Lyapunov, filtrage de Kalman, etc.— dans l'analyse des propriétés en boucle-fermée de la commande à horizon fuyant et son extension à l'étude des systèmes multivariables.

Notons, à la fin de cet aperçu historique, l'importante contribution de Becerra et al [22,23] sur l'application industrielle de la commande prédictive augmentée par l'intégration des objectifs et des contraintes économiques dans la formulation du problème d'optimisation et, ceci dans le cadre d'une stratégie globale de commande prédictive.

1.6. Les problématiques de la commande prédictive

Aujourd'hui, la commande prédictive référencée modèle est une stratégie largement admise et acceptée dans les milieux industriels et particulièrement dans les industries chimiques et pétrochimiques. Un développement significatif de la commande prédictive est aussi enregistré dans d'autres domaines tels que les industries agroalimentaires, l'industrie du papier, l'aérospatial et l'automobile [2]. Les principaux avantages de cette stratégie sont :

1. Applicabilité à toute sorte de processus même ceux présentant des caractéristiques d'instabilités en boucle ouverte ou, des retards temporels
2. S'accommode avec les processus/systèmes multi-variables ou multi entrées multi sorties.
3. Prise en charge efficace des contraintes d'opération du processus sous commande de manière optimale. Ceci permet aussi d'assurer une exploitation fiable et sécurisée des processus.
4. Possibilité d'intégrer les facteurs économiques (coût, profit, ...) dans la formulation du problème d'optimisation.
5. La commande prédictive est une technologie ouverte pour des additions et des améliorations futures.

Cependant, le succès, l'extension des applications et les avantages qu'offre la commande prédictive reposent sur une prise en charge adéquate des problématiques suivantes [24] :

1. La modélisation et l'identification
2. La stabilité
3. La poursuite de trajectoires (tracking)
4. la robustesse
5. L'optimisation

1.2.6 La modélisation et l'identification

Les performances d'une commande prédictive dépendent énormément de la disponibilité dans la structure de commande d'un modèle fiable et fidèle du processus sous commande pour permettre des prédictions précises des variables d'état/sortie tout le long de l'horizon de prédiction.

Plusieurs techniques sont proposées dans la littérature de l'identification des systèmes pour l'estimation directe ou indirecte des paramètres d'un modèle d'état, en se basant sur des mesures entrée/sortie obtenues lors d'une excitation du système soit par une impulsion, un signal échelon ou encore un signal synthétique tel un PRBS (Pseudo-Random binary sequence). L'identification indirecte se fait par transformation d'un modèle empirique canonique préalablement construit (ARX, ARMAX, CARIMA, ...) en un modèle d'état linéaire dans lequel le vecteur d'état comprend les échantillons courants et passés des

sorties mesurables du système [19-21, 23]. L'autre alternative, l'identification directe du modèle d'état, est une approche plus avancée connue sous l'appellation : identification par sous-espace [25].

Jusqu'à une date récente, les applications industrielles de la commande prédictive se suffisaient de l'emploi de modèles linéaires, bien que la majorité des procédés ou des systèmes contrôlés sont de nature non linéaire. Cette situation est acceptable tant que le système commandé est restreint à fonctionner autour d'un état local, stable et stationnaire. Dans ce cas, et quoique la dynamique du système est globalement non linéaire, le modèle linéaire reste une approximation valable au voisinage de cet état d'équilibre. Cependant, et dès que le système s'éloigne de son état de fonctionnement stationnaire le modèle linéaire devient inadéquat pour l'estimation des états/sorties futures entraînant une dégradation dans la performance de la commande prédictive. Dans ce cas, l'emploi d'un modèle non linéaire est nécessaire pour compenser les insuffisances de la commande prédictive induites par l'incapacité du modèle linéaire à exprimer correctement la dynamique non linéaire du système. Cette nécessité d'introduire les modèles non linéaires dans la formulation de la problématique de la commande prédictive est aujourd'hui bien reconnue, et est traduite par l'apparition d'une nouvelle approche : La commande prédictive non-linéaire ou encore, la commande non-linéaire à horizon fuyant (nonlinear receding horizon control) [26].

C'est dans ce contexte que les réseaux de neurones ont émergés comme un outil indispensable pour l'identification des processus non linéaires [27-29] et, cette puissance de l'outil neuronique s'est renforcée à cause de leurs aptitudes prouvées à l'approximation des fonctionnelles non linéaires avec une précision choisie *a priori*.

1.2.7 La stabilité

La stabilité est une propriété et condition importante pour la mesure de la performance d'une stratégie de commande. Pour un système/processus décrit par un modèle d'état on définit, généralement, un état d'équilibre (souvent nul) comme étant la solution constante de l'équation d'état pour un vecteur de commande nul. Ceci étant dit, l'équilibre d'un système peut aussi correspondre à un état non nul.

Dans la théorie générale de la stabilité de Lyapunov : Un état d'équilibre est dit asymptotiquement stable si, toute solution (trajectoire dans l'espace d'état) de l'équation d'état initiée au voisinage de cet état d'équilibre converge vers cet état et, fini par l'atteindre.

Concernant la commande prédictive et comme déjà mentionné dans la section précédente (1.1), la stabilité était obtenue en ajustant certains paramètres de la fonction de coût en choisissant, par exemple, des valeurs larges pour l'horizon de prédiction [8, 9, 30 et 31]. Ceci était considéré comme une faiblesse majeure de cette stratégie de commande [14]. La percée dans l'analyse et l'imposition de la stabilité pour la commande prédictive vint, lorsque le problème optimal de la commande fût redéfini dans un cadre de modèle d'espace d'état et en adoptant la théorie de stabilité de Lyapunov comme outil d'analyse de la stabilité [21,32-33] sachant que, l'analyse de la stabilité pour des modèles empiriques présente des difficultés théoriques majeures.

Pour les systèmes linéaires la théorie de la commande quadratique et la programmation dynamique étaient d'un grand apport pour la définition des conditions de stabilité d'une commande prédictive linéaire. Dans le cas non linéaire, cependant, c'est la théorie de Lyapunov qui va jouer le premier rôle dans l'établissement des conditions de stabilité pour des systèmes continus ou discrets décrits par des équations d'état [24]. Comme pour le cas linéaire, l'analyse de la stabilité pour une commande prédictive basée sur des modèles empiriques non linéaires de type entrées/sorties est, encore plus, rarement évoquée [34-36].

1.2.8 La poursuite de trajectoires

L'objectif d'une commande en poursuite est de forcer les états et les sorties d'un système à suivre une trajectoire de référence. Si les sorties sont des fonctions des états, il suffirait alors de forcer les états à suivre une trajectoire dans l'espace d'état. Pour les systèmes linéaires la commande en poursuite optimale est, généralement, conçue dans le cadre de la régulation quadratique linéaire ou de la commande prédictive linéaire si les états et les variables de commandes sont soumis à des contraintes [21,37]. Cependant, et comme noté par Magni et al. [38] '*les algorithmes stabilisants de la commande prédictive non linéaire orientés*

vers des applications de poursuites restent rares' et, 'seul De Nicolao et al. [35] proposent un algorithme stabilisant pour des systèmes décrits par une classe restreinte de modèles NARX'. Dans [39], Magni dérive une loi de commande prédictive non linéaire stabilisante pour résoudre le problème de poursuite pour des références constantes en effectuant une transformation des équations d'état et en évitant la tâche difficile de calculer les déviations des variables d'état à partir de la valeur de consigne de la sortie.

1.2.9 La robustesse

La question de la robustesse se pose dès que les incertitudes sur la représentation du processus sont admises. Il convient, alors, de définir les conditions pour maintenir certaines propriétés telles que la stabilité et la performance de la commande en présence de ces incertitudes. Les causes principales des incertitudes sont :

1. Erreurs d'estimations des états et, cette situation arrive quand les états du système ne sont pas directement accessibles mais sont estimés par un observateur sur la base des mesures faites sur les sorties.
2. Erreurs de modélisation qui sont des incertitudes causées durant l'identification du système ou, dues à des changements dans la dynamique du système.
3. Perturbations externes.

L'objectif de la commande prédictive robuste est, donc, de prendre en considération ces facteurs dans la conception de l'algorithme de commande [24], [35] et [40-43].

1.2.10 L'optimisation

Comme la stratégie de commande prédictive repose sur le principe d'une optimisation à répétition d'une fonction coût en présence de certaines contraintes sur les variables d'état et de commande, la disponibilité d'un algorithme efficace et rapide dans la structure de commande pour l'optimisation en-ligne de la fonction de coût et la détermination d'un profil de commande optimale devient impérative. Lorsque le système est linéaire, la fonction coût est linéaire ou quadratique par rapport à ses arguments et les contraintes sont, eux aussi, exprimées par des

inégalités linéaires, alors le problème d'optimisation est lui-même linéaire et peut être traité dans le cadre d'une programmation quadratique (QP) qui désigne une méthode numérique itérative pour résoudre les problèmes d'optimisation linéaires avec prise en charge des contraintes [21,43-46]. Dans le cas général d'un système dynamique non linéaire et d'un problème optimal exprimé par une fonction coût non-linéaire et non quadratique, il devient nécessaire d'utiliser des techniques de programmation non linéaires (NLP). Partant avec un objectif d'efficacité et de rapidité, Becerra et al. [23,47] développèrent un algorithme d'optimisation pour les systèmes non linéaires discrets représentés par des modèles empiriques qu'ils ont appelés : Dynamic Integrated System Identification and Parameter Estimation ou DISOPE. Enfin, l'article de Biegler [48], donne une synthèse complète sur les algorithmes disponibles et les questions ouvertes de l'optimisation dans la commande prédictive.

1.7. Contributions de la thèse

Dans ce travail nous abordons les problématiques de : *la modélisation et l'identification, de la stabilité et de poursuite de trajectoires* dans la commande prédictive non linéaire basée sur des modèles neuroniques. Notre motivation est double :

1. Montrer que les réseaux de neurones peuvent, effectivement, s'intégrer dans des schémas de commande prédictive non linéaire tout en garantissant les propriétés régulatrices et stabilisatrices.
2. Dériver une loi de commande stabilisante, basée sur une classe de réseaux neuroniques, pour traiter la problématique de la poursuite optimale de trajectoire dans l'espace d'état dans un cadre générale de commande prédictive non linéaire.

Prenant en considération ces objectifs, nous pouvons résumer la contribution de cette thèse dans les axes suivants :

- i) **Modélisation et identification** : Dans une première phase nous avons orienté notre travail sur la création de modèles type NARX en utilisant des réseaux neuroniques GRBF (Gaussian radial Basis Function). Au cours de cette étape, nous avons exploité et modifié l'algorithme d'apprentissage RANEKF (Resource Allocating Network with Extended

Kalman Filter) pour l'identification des systèmes dynamiques non linéaires. Cette contribution était sanctionnée par une communication internationale [49]. L'objectif de ce travail était de pouvoir intégrer le modèle neuronique NARX dans une structure de commande prédictive stabilisante. Cependant, et bien que les modèles identifiés donnent de bons résultats en simulation, les propriétés de stabilité n'étaient pas prouvées.

ii) Régulation prédictive stabilisante basée sur une structure de réseaux de neurones : La situation décrite plus haut, nous a conduit à construire une structure neuronique basée sur des modèles GRBF et pouvant émuler une représentation non linéaire d'état. Notre objectif était de pouvoir intégrer directement cette structure neuronique dans un schéma de commande prédictive non linéaire stabilisant. En effet, et en appliquant l'algorithme RANEKF modifié, il était possible de construire un modèle GRBF à rétroaction externe qui est, non seulement capable d'approximer le système non linéaire identifié mais aussi, d'apprendre ces propriétés dynamiques et de stabilité. Cette structure est ensuite utilisée dans un schéma stabilisant de commande prédictive non linéaire à horizon fini avec fonction de coût terminale. Cette contribution constitue une partie de nos travaux publiés [50].

iii) Solution stabilisante pour une classe de problèmes de poursuite optimale basée sur des réseaux neuroniques :

Le succès de notre démarche pour intégrer les réseaux neuroniques dans une stratégie de commande prédictive stabilisante, nous a encouragé à aborder le problème de poursuite de trajectoire pour cette classe particulière de modèles. Nous avons exploité une propriété analytique intéressante des réseaux GRBF pour transformer le problème de poursuite en un problème de régulation. Ce dernier est, ensuite, formulé dans le cadre d'une régulation prédictive stabilisante. Ceci, nous a permis de proposer une nouvelle structure de commande en poursuite pour les systèmes non linéaires modélisés par une classe de réseaux neuroniques GRBF. Cette structure comporte un régulateur prédictif stabilisant basé sur un modèle non linéaire explicite des

déviations des variables d'état, et d'un contrôleur neuronique pour maintenir l'état à sa valeur de référence. C'est donc, un contrôleur non linéaire hybride prédictif-neuronique avec stabilité imposée [50].

1.8. Organisation du manuscrit

Après cette introduction, le reste de ce travail est organisé comme suit :

- Le chapitre 2, introduit les concepts de stabilité asymptotique, d'optimalité et de programmation dynamique pour des systèmes décrits par des modèles d'état discrets. Ensuite, la problématique de la régulation linéaire quadratique est formulée et, les principaux résultats de stabilité présentés. Le reste du chapitre est consacré à la commande prédictive linéaire.
- Le chapitre 3, contient une description de la problématique générale de la commande prédictive non linéaire. Le problème est défini pour les cas de l'horizon fini et infini. Le chapitre présente, aussi, les différentes approches et algorithmes qui assurent la stabilité en boucle fermée pour un système non linéaire entraîné par une commande prédictive.
- Le chapitre 4, présente les réseaux de neurones artificiels non bouclés. Après une introduction historique sur le concept du neurone artificiel, les perceptrons multicouches et les réseaux RBF sont introduits. Le reste du chapitre reprend les résultats théoriques qui consacrent les perceptrons et les réseaux RBF comme des 'Approximateurs Universels' pour les fonctionnelles non linéaires.
- Dans le chapitre 5, nous abordons le sujet de la modélisation et de l'identification des systèmes non linéaire. Après un rappel sur les différentes structures des modèles empiriques, nous présentons l'algorithme *RANEKF modifié à apprentissage aléatoire* avec des exemples d'identification de modèles NARX. Nous présentons aussi notre structure de réseaux RBF pour l'émulation d'un modèle d'état discret non

linéaire. Le chapitre se termine par un traitement de l'identification non linéaire inverse.

- Le chapitre 6, présente les principaux résultats concernant la stabilisation nominale d'un schéma de commande prédictive basée sur une classe de réseaux de neurones GRBF. La problématique de la commande prédictive non linéaire en poursuite de trajectoire dans l'espace d'état est formulée et, l'architecture générale de notre structure de commande stabilisante en poursuite basée sur une transformation du problème de poursuite en un problème de régulation, est présentée. Des résultats de simulations démontrant les performances de la commande sont aussi avancés.
- La conclusion donne un aperçu général sur les travaux accomplis et discute des orientations futures de notre recherche.

Chapitre 2

De la régulation linéaire quadratique à la commande prédictive

Dans ce chapitre nous passerons en revue la commande prédictive linéaire et ses propriétés de stabilité en se basant sur un modèle d'état discret, linéaire et invariant. Les concepts de base concernant la stabilité asymptotique des systèmes discrets sont présentés dans la première section du chapitre. La deuxième section est consacrée au principe d'optimalité et au concept important de la programmation dynamique. La problématique de la régulation linéaire quadratique déterministe est introduite dans la troisième section et, la commande prédictive linéaire est analysée à la fin du chapitre.

2.5 Stabilité asymptotique des systèmes dynamiques

Considérons un système discret décrit par l'équation suivante :

$$x(t+1) = f(t, x(t)) \quad t=0,1,2,\dots \quad (2.1)$$

avec la condition initiale $x(0) = x_0$ et l'état du système $x \in \mathbb{R}^n$. Le système défini par (2.1) est dit invariant si la fonction-vecteur f est indépendante du temps t , et il est dit linéaire si $f(t, x(t)) = A(t)x(t)$; avec la matrice $A(t) \in \mathbb{R}^{n \times n}$.

Définition 2.1 : Un point x_e est dit point d'équilibre du système décrit par (2.1) si, $f(t, x_e) = 0$ pour tout $t \geq 0$.

Si l'origine $x=0$ est un point d'équilibre, alors la fonction f satisfait $f(0) = 0$.

Définition 2.2 : Le point x_e est dit point d'équilibre stable du système décrit par (2.1) si, quelque soit $\delta > 0$, il existe un nombre $\varepsilon \geq 0$, tel que si, $\|x(t) - x_e\| < \varepsilon$ on ait $\|x(t+l) - x_e\| < \delta$ pour tout $l \geq 0$.

Définition 2.3 : Le point x_e est dit point d'équilibre asymptotiquement stable du système décrit par (2.1) si,

1. x_e est point d'équilibre stable.
2. Il existe un nombre $\varepsilon \geq 0$ tel que : $\|x(t) - x_e\| < \varepsilon$ entraîne

$$\lim_{l \rightarrow \infty} \|x(t+l) - x_e\| = 0 .$$

Définition 2.4 : Une fonction $\alpha(\varepsilon): \mathfrak{R}^+ \rightarrow \mathfrak{R}^+$ est dite fonction de classe K (on écrit $\alpha(\varepsilon) \in K$) si , elle est continue, monotone non-décroissante est nulle à l'origine ($\alpha(0)=0$).

A partir de ces définitions nous allons énoncer le théorème de stabilité asymptotique de Lyapunov [51,52].

Théorème 2.1 : S'il existe une fonction $V: \mathfrak{R}^n \rightarrow \mathfrak{R}^+$ qui vérifie les conditions suivantes :

1. elle existe une fonction α de classe K , non-négative et non- décroissante tel que :

$$0 < \alpha(\|x(t)\|) \leq V(x(t)) \quad , \quad \text{pour tout } x(t) \neq 0. \quad (2.2)$$

2. elle existe une fonction γ de classe K, strictement croissante tel que

$$V(x(t)) - V(x(t+l)) \geq \gamma(\|x(t)\|) \quad (2.3)$$

3. V est continue et $V(0)=0$.

Alors le système décrit par (2.1) est asymptotiquement stable.

Démonstration : Commençons d'abord par prouver la stabilité et choisisant un nombre $\delta > 0$. Etant donné que la fonction de Lyapunov $V(\cdot)$ est continue ; il existe alors un nombre $\rho > 0$, tel que $V(x(t)) < \alpha(\delta)$, pour $\|x(t)\| < \rho$.

Si nous supposons le système (2.1) instable, il existe alors un nombre $l \geq 0$ pour lequel $\|x(t+l)\| \geq \delta$, et on aurait :

$$V(x(t)) < \alpha(\rho) \leq \alpha(\|x(t+l)\|) \leq V(x(t+l)) \quad (2.4)$$

Ce résultat contredit l'hypothèse $V(x(t)) - V(x(t+l)) \geq 0$, et on déduit que le système est stable.

Pour prouver la convergence, on prend l'hypothèse

$$V(x(t)) - V(x(t+l)) \geq \gamma(\|x(t)\|) \quad (2.5)$$

Tenant compte de la propriété de non-négativité de la fonction de Lyapunov $V(\cdot)$, effectuant la sommation des inégalités obtenues à partir de (2.5) pour obtenir :

$$V(\mathbf{x}(t)) \geq V(\mathbf{x}(t)) - V(\mathbf{x}(t+l)) \geq \sum_{j=t}^{t+l-1} \gamma(\|\mathbf{x}(j)\|) \quad (2.6)$$

Sachant que $\gamma(\|\mathbf{x}(j)\|)$ est positive et que la somme partielle $\sum_{j=t}^{t+l-1} \gamma(\|\mathbf{x}(j)\|)$ est majorée par $V(\mathbf{x}(t))$ qui est indépendante de l . On déduit alors,

$$\lim_{l \rightarrow \infty} \gamma(\|\mathbf{x}(t+l)\|) = 0 \quad (2.7)$$

Sachant aussi que $\gamma(0) = 0$ et que la fonction γ est continue, ceci implique $\|\mathbf{x}(t+l)\| \rightarrow 0$. Le système est asymptotiquement stable. CQFD.

Ce théorème de Lyapunov établit des conditions suffisantes pour garantir la stabilité asymptotique du système (2.1). Il est remarquable de savoir que l'inverse du théorème 1 est aussi vrai : *Si un point d'équilibre est asymptotiquement stable alors, elle existe une fonction de Lyapunov $V(\cdot)$ Telle que $V(\mathbf{x}(t)) - V(\mathbf{x}(t+l)) \geq 0$.* Notons enfin, qu'il n'existe aucune prescription particulière quant à la manière de choisir la fonction de Lyapunov.

2.6 Le principe d'optimalité et la programmation dynamique [53-55]

Considérons un système dynamique décrit par un vecteur- fonction f qui peut être non-linéaire ;

$$\begin{aligned} \mathbf{x}(t+l) &= \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) & t=0,1,2,\dots \\ \mathbf{y}(t) &= \mathbf{g}(t, \mathbf{x}(t)) \end{aligned} \quad (2.8)$$

Dans cette formulation, $\mathbf{y} \in \mathcal{R}^m$ représente le vecteur de sortie, $\mathbf{u} \in \mathcal{R}^p$ le vecteur d'entrée et, $\mathbf{x} \in \mathcal{R}^n$ le vecteur des variables d'état. Le vecteur-fonction $f(\cdot, \cdot)$ représente la *fonction de transition d'état locale*, qui détermine la règle de transition d'un événement $(t, \mathbf{x}(t))$ vers l'événement suivant $(t+l, \mathbf{x}(t+l))$, sous l'action d'une commande $\mathbf{u}(t)$ appliquée à l'instant discret t .

Par itérations successives de l'équation (2.8), on peut arriver à définir une *fonction de transition d'état globale* ;

$$\mathbf{x}(t+j) = \varphi(t+j, t, \mathbf{x}(t), \mathbf{u}_{t,t+j-1}) \quad j \geq 0 \quad (2.9)$$

où $\mathbf{u}_{t,t+j-1} = \{\mathbf{u}(t), \mathbf{u}(t+1), \dots, \mathbf{u}(t+j-1)\}$ est une séquence de commande. La fonction $\varphi(\cdot)$ permet de calculer un état final $(t+j, \mathbf{x}(t+j))$ à partir d'un état initial $(t, \mathbf{x}(t))$ et d'une séquence de commande $\mathbf{u}_{t,t+j-1}$, pour $j \geq 0$. Pour illustrer, prenons l'exemple $j=2$:

$$\begin{aligned} \mathbf{x}(t+2) &= \mathbf{f}(t+1, \mathbf{x}(t+1), \mathbf{u}(t+1)) \\ &= \mathbf{f}(t+1, \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \mathbf{u}(t+1)) \\ &= \varphi(t+2, t, \mathbf{x}(t), \mathbf{u}_{t,t+1}) \end{aligned} \quad (2.10)$$

Pour le cas particulier $j=0$, la séquence $\mathbf{u}_{t,t}$ est un ensemble vide et le système n'évolue pas et reste dans son état $(t, \mathbf{x}(t))$. Ceci impose la condition suivante à la fonction $\varphi(\cdot)$:

$$\mathbf{x}(t) = \varphi(t, t, \mathbf{x}(t), \mathbf{u}_{t,t}) \quad (2.11)$$

Définissons maintenant une fonction de coût, pas nécessairement quadratique:

$$\Phi(t, \mathbf{x}(t), \mathbf{u}_{t,t+N-1}) = \sum_{k=t}^{t+N-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \psi(\mathbf{x}(t+N)) \quad (2.12)$$

où $l(k, \mathbf{x}(k), \mathbf{u}(k))$ et $\psi(\mathbf{x}(t+N))$ sont des fonctions scalaires non-négatives, et où l'intervalle $[t, t+N]$ représente l'horizon de commande. Le problème de commande optimale est celui de déterminer la séquence de commande optimale $\mathbf{u}_{t,t+N-1}^o = \{\mathbf{u}^o(t), \mathbf{u}^o(t+1), \dots, \mathbf{u}^o(t+N-1)\}$ pour le système (2.8), et qui minimise la fonction de coût (2.12).

Pour ce problème optimal introduisons la fonction suivante ;

$$\begin{aligned} \Phi^0(t, \mathbf{x}(t)) &= \min_{\mathbf{u}_{t,t+N-1}} \{ \Phi(t, \mathbf{x}(t), \mathbf{u}_{t,t+N-1}) \} \\ &= \min_{\mathbf{u}_{t,t+N-1}} \left\{ \sum_{k=t}^{t+N-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \psi(\mathbf{x}(t+N)) \right\} \end{aligned} \quad (2.13)$$

Ils existent deux méthodes pour le traitement du problème (2.13) : L'optimisation séquentielle (batch optimisation), et la programmation dynamique. La technique de l'optimisation séquentielle repose sur des algorithmes de recherche de la valeur optimale par des procédures numériques alors que la programmation dynamique est une méthode analytique. La programmation dynamique (PD) est préférée quant- il s'agit d'analyser les propriétés en boucle fermée du système commandé. Pour continuer le traitement du problème

optimale à N -étapes par la PD , considérons une étape j entre 0 et l'horizon de commande N : $j \in \{1, 2, \dots, N\}$.

On peut écrire (2.13) comme suit :

$$\begin{aligned}\Phi^0(t, \mathbf{x}(t)) &= \min_{\mathbf{u}_{t:t+N-1}} \left\{ \sum_{k=t}^{t+j-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \sum_{k=t+j}^{t+N-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \psi(\mathbf{x}(t+N)) \right\} \\ &= \min_{\mathbf{u}_{t:t+N-1}} \left\{ \sum_{k=t}^{t+j-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \Phi(t+j, \mathbf{x}(t+j), \mathbf{u}_{t+j:t+N-1}) \right\}\end{aligned}\quad (2.14)$$

En utilisant la fonction de transition d'état globale pour exprimer $\mathbf{x}(t+j)$, l'équation (2.14) devient ;

$$\begin{aligned}\Phi^0(t, \mathbf{x}(t)) &= \min_{\mathbf{u}_{t:t+N-1}} \left\{ \sum_{k=t}^{t+j-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \Phi(t+j, \varphi(t+j, t, \mathbf{x}(t), \mathbf{u}_{t:t+j-1}), \mathbf{u}_{t+j:t+N-1}) \right\} \\ &= \min_{\mathbf{u}_{t:t+j-1}} \left\{ \min_{\mathbf{u}_{t+j:t+N-1}} \left[\sum_{k=t}^{t+j-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \Phi(t+j, \varphi(t+j, t, \mathbf{x}(t), \mathbf{u}_{t:t+j-1}), \mathbf{u}_{t+j:t+N-1}) \right] \right\}\end{aligned}\quad (2.15)$$

Sachant que la somme sur $l(k, \mathbf{x}(k), \mathbf{u}(k))$ ne dépend pas de la séquence de commande restreinte $\mathbf{u}_{t+j:t+N-1}$, et en substituant par la fonction (2.13) dans l'équation précédente, on obtient

$$\begin{aligned}\Phi^0(t, \mathbf{x}(t)) &= \min_{\mathbf{u}_{t:t+j-1}} \left\{ \sum_{k=t}^{t+j-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \min_{\mathbf{u}_{t+j:t+N-1}} \left[\Phi(t+j, \varphi(t+j, t, \mathbf{x}(t), \mathbf{u}_{t:t+j-1}), \mathbf{u}_{t+j:t+N-1}) \right] \right\} \\ &= \min_{\mathbf{u}_{t:t+j-1}} \left\{ \sum_{k=t}^{t+j-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \Phi^0(t+j, \varphi(t+j, t, \mathbf{x}(t), \mathbf{u}_{t:t+j-1})) \right\}\end{aligned}\quad (2.16)$$

Posons $t_1 = t+j$ pour désigner un instant postérieur à t dans l'horizon de commande, et substituant par t_1 dans l'équation (2.16) :

$$\begin{aligned}\Phi^0(t, \mathbf{x}(t)) &= \min_{\mathbf{u}_{t:t_1-1}} \left\{ \sum_{k=t}^{t_1-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \min_{\mathbf{u}_{t_1:t+N-1}} \left[\Phi(t_1, \varphi(t_1, t, \mathbf{x}(t), \mathbf{u}_{t:t_1-1}), \mathbf{u}_{t_1:t+N-1}) \right] \right\} \\ &= \min_{\mathbf{u}_{t:t_1-1}} \left\{ \sum_{k=t}^{t_1-1} l(k, \mathbf{x}(k), \mathbf{u}(k)) + \Phi^0(t_1, \varphi(t_1, t, \mathbf{x}(t), \mathbf{u}_{t:t_1-1})) \right\}\end{aligned}\quad (2.17)$$

Où $\mathbf{u}_{t:t_1-1}$ représente la séquence de commande restreinte à un horizon $[t, t_1]$, avec $t_1 \leq t+N$.

En supposant l'existence d'une séquence optimale $\mathbf{u}_{t,t+N-1}^o$ solution du problème optimal défini par (2.12) sur un horizon de commande $[t, t+N]$, à partir d'un état initial $(t, \mathbf{x}(t))$, on peut écrire pour toute séquence de commande $\mathbf{u}_{t,t+N-1}$ ($\mathbf{u}_{t,t+N-1}$ étant non-optimale) :

$$\begin{aligned}\Phi^0(t, \mathbf{x}(t)) &= \Phi(t, \mathbf{x}(t), \mathbf{u}_{t,t+N-1}^o) \\ &\leq \Phi(t, \mathbf{x}(t), \mathbf{u}_{t,t+N-1})\end{aligned}\quad (2.18)$$

De même, la séquence $\mathbf{u}_{t_1,t+N-1}^o$; restriction de $\mathbf{u}_{t,t+N-1}^o$ sur l'horizon $[t_1, t+N]$, est optimale pour le système (2.8) initialisé à l'état $(t_1, \mathbf{x}(t_1))$, avec $\mathbf{x}(t_1) = \varphi(t_1, t, \mathbf{x}(t), \mathbf{u}_{t,t-1})$ et l'on peut écrire en vertu du résultat (2.18) :

$$\begin{aligned}\Phi^0(t_1, \mathbf{x}(t_1)) &= \Phi(t_1, \mathbf{x}(t_1), \mathbf{u}_{t_1,t+N-1}^o) \\ &\leq \Phi(t_1, \mathbf{x}(t_1), \mathbf{u}_{t_1,t+N-1})\end{aligned}\quad (2.19)$$

Et l'on déduit le principe d'optimalité suivant :

Pour une séquence optimale $\mathbf{u}_{t_0,t_0+N-1}^o$ et, pour tout événement intermédiaire $(t_1, \mathbf{x}(t_1))$ le long de la trajectoire optimale correspondante avec $\mathbf{x}(t_1) = \varphi(t_1, t_0, \mathbf{x}(t_0), \mathbf{u}_{t_0,t_1-1}^o)$, la séquence de commande restreinte $\mathbf{u}_{t_1,t_0+N-1}^o$ est aussi optimale pour le reste de l'horizon de commande $[t_1, t_0+N]$.

En posant $j=1$ dans l'équation (2.16), on déduit la fonctionnelle récurrente suivante (Equation de Bellman) :

$$\begin{aligned}\Phi^0(t, \mathbf{x}(t)) &= \min_{\mathbf{u}_{t,t}} \{l(t, \mathbf{x}(t), \mathbf{u}(t)) + \Phi^0(t+1, \varphi(t+1, t, \mathbf{x}(t), \mathbf{u}_{t,t}))\} \\ &= \min_{\mathbf{u}(t)} \{l(t, \mathbf{x}(t), \mathbf{u}(t)) + \Phi^0(t+1, \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)))\}\end{aligned}\quad (2.20)$$

avec la condition sur la valeur finale de $\Phi^0(.,.)$ donnée par :

$$\Phi^0(t+N, \mathbf{x}(t+N)) = \psi(\mathbf{x}(t+N))\quad (2.21)$$

En exploitant judicieusement La fonctionnelle (2.20), le problème d'optimisation à N -étapes peut-être divisée en N sous-problèmes d'optimisation. A cet fin, introduisons un indice i dans l'équation (2.20) pour désigner le $i^{\text{ème}}$ sous-problème d'optimisation :

$$\Phi_i^0(t+N-i, \mathbf{x}(t+N-i)) = \min_{\mathbf{u}(t+N-i)} \{l(t+N-i, \mathbf{x}(t+N-i), \mathbf{u}(t+N-i)) + \Phi_{i-1}^0(t+N-i+1, \mathbf{x}(t+N-i+1))\} \quad (2.22)$$

avec $i \in \{1, 2, \dots, N\}$. L'équation (2.22) est récursive par rapport à l'indice i dans le sens décroissant de l'horizon de prédiction N , avec une condition initiale pour $i=1$ qui correspond à l'étape finale $\Phi_0^0(t+N, \mathbf{x}(t+N)) = \psi(\mathbf{x}(t+N))$ du problème de commande optimale. La solution optimale $\mathbf{u}^0(t+N-i)$ du $i^{\text{ème}}$ sous-problème tel que décrit par (2.22), lorsqu'elle existe, prends la forme d'un retour d'état (state-feedback) :

$$\mathbf{u}^0(t+N-i) = \mathbf{u}^0(t+N-i, \mathbf{x}(t+N-i)) \quad (2.23)$$

Cette technique d'optimisation basée sur la programmation dynamique prends l'état du système $\mathbf{x}(\cdot)$, et fourni une commande optimale par retour d'état.

2.7 La Régulation Linéaire Quadratique (RLQ)

Cette section est consacrée au problème de la régulation linéaire quadratique (RLQ) qui représente, d'un point de vue historique, le point de départ de la technique de commande prédictive et ses variantes linéaire et nonlinéaire [14].

2.3.1 Position du problème

Soit un système discret, invariant et linéaire décrit dans l'espace d'état par l'équation

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (2.24)$$

tel que $\mathbf{x} \in \mathfrak{R}^n$ représente le vecteur d'état, $\mathbf{u} \in \mathfrak{R}^p$ est le vecteur de commande donnés à l'instant discret t . Les matrices \mathbf{A} et \mathbf{B} étant de dimensions appropriées. Le problème de commande consiste à 'déplacer', d'une manière optimale, l'état $\mathbf{x}(t)$ du système (2.24) vers l'origine à partir d'un état initial \mathbf{x}_0 . L'objectif de commande est, alors, défini comme suit [56]:

Maintenir l'état du système le plus proche possible de l'origine de l'espace d'état, tout en fournissant le minimum d'effort en terme d'action de commande.

A chaque instant discret t , la distance par rapport à l'origine est mesurée par le carré pondéré de la norme euclidienne du vecteur d'état : $\|\mathbf{x}(t)\|_Q^2 = \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)$. L'action de commande étant évaluée par la quantité $\|\mathbf{u}(t)\|_R^2 = \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)$.

La commande optimale du problème de la régulation quadratique linéaire s'obtient, alors, en minimisant une fonction de coût quadratique définie sur un horizon N par

$$\Phi_N(t, \mathbf{x}(t)) = \min_{\mathbf{u}(t+N-i)} \left\{ \sum_{k=t}^{t+N-1} \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k) + \mathbf{x}^T(t+N)\mathbf{S}\mathbf{x}(t+N) \right\} \quad (2.25)$$

avec $\mathbf{Q} \geq 0$, $\mathbf{R} > 0$ et $\mathbf{S} \geq 0$ des matrices symétriques de pondération. Le problème ainsi posé est dit à horizon fini. Dans le cas limite $N \rightarrow \infty$, la fonction de coût s'écrit

$$\Phi_\infty(t, \mathbf{x}(t)) = \min_{\mathbf{u}} \left\{ \sum_{k=t}^{\infty} \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k) \right\} \quad (2.26)$$

et le problème optimal est dit à horizon infini.

Notons, que le problème classique de la régulation linéaire quadratique (à horizon fini ou infini) ne pose aucune condition ou contrainte sur l'évolution des vecteurs d'état et de commande dans leur espaces respectifs.

2.3.2 Solution du problème RLQ

Pour rester dans la même logique établie dans la section (2.2), on adopte la technique de la programmation dynamique telle que formulée par l'équation (2.22) comme point de départ vers la solution du problème RLQ et la synthèse d'une séquence de commande optimale $\mathbf{u}_{t,t+N-1}^0 = \{\mathbf{u}^0(t), \mathbf{u}^0(t+1), \dots, \mathbf{u}^0(t+N-1)\}$, où l'élément $\mathbf{u}^0(t)$ prend la forme d'un retour d'état (state-feedback) :

$$\mathbf{u}^0(t) = \mathbf{u}^0(t, \mathbf{x}(t)) \quad (2.27)$$

La solution du problème optimal décrit par (2.24) et (2.25), pour un horizon de prédiction N , est une loi de commande linéaire par retour d'état [14] :

$$\mathbf{u}(t) = \mathbf{K}_N \mathbf{x}(t) \quad (2.28)$$

où \mathbf{K}_N est la matrice-gain du régulateur LQ :

$$\mathbf{K}_N = -(\mathbf{R} + \mathbf{B}^T \mathbf{P}_N \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P}_N \mathbf{A} \quad (2.29)$$

et où P_N est une matrice symétrique non-négative ($P_N > 0$) solution d'une équation aux différences de Riccati (EDR)

$$P_N = A^T P_{N-1} A - A^T P_{N-1} B (R + B^T P_{N-1} B)^{-1} B^T P_{N-1} A + Q \quad (2.30)$$

avec la condition initiale $P_0 = S$.

Ce résultat intéressant s'obtient en commençant, d'abord, par réécrire l'équation (2.22) pour tout Indice $i \in \{1, 2, \dots, N\}$:

$$\Phi_i^0(t + N - i, \mathbf{x}(t + N - i)) = \min_{\mathbf{u}(t + N - i)} \{l(t + N - i, \mathbf{x}(t + N - i), \mathbf{u}(t + N - i)) + \Phi_{i-1}^0(t + N - i + 1, \mathbf{x}(t + N - i + 1))\} \quad (2.31)$$

ou encore, en rendant le temps discret implicite dans l'équation précédente, on obtient

$$\Phi_i^0(\mathbf{x}) = \min_{\mathbf{u}} \{l(\mathbf{x}, \mathbf{u}) + \Phi_{i-1}^0(f(\mathbf{x}, \mathbf{u}))\} \quad (2.32)$$

sachant que $l(\mathbf{x}, \mathbf{u}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}$, $f(\mathbf{x}, \mathbf{u}) = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}$, et $\Phi_0^0(\mathbf{x}) = \mathbf{x}^T \mathbf{S} \mathbf{x}$, l'équation de programmation dynamique à étape unique pour $i=1$ s'écrit

$$\Phi_1^0(\mathbf{x}) = \min_{\mathbf{u}} \{\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} + (\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u})^T \mathbf{S} (\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u})\} \quad (2.33)$$

sachant que la variable de commande \mathbf{u} est non contrainte, la valeur de la commande optimale \mathbf{u}^0 s'obtient facilement par résolution de l'équation :

$$\frac{\partial \Phi^0}{\partial \mathbf{u}} = 2\mathbf{R} \mathbf{u} + 2\mathbf{B}^T \mathbf{S} (\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}) = 0 \quad (2.34)$$

pour obtenir

$$\mathbf{u}^0 = -(\mathbf{R} + \mathbf{B}^T \mathbf{S} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{S} \mathbf{A} \mathbf{x} \quad (2.35)$$

Par substitution de l'équation (2.35) dans (2.36), on obtient pour $i = 1$

$$\Phi_1^0(\mathbf{x}) = \mathbf{x}^T [\mathbf{A}^T \mathbf{S} \mathbf{A} - \mathbf{A}^T \mathbf{S} \mathbf{B} (\mathbf{R} + \mathbf{B}^T \mathbf{S} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{S} \mathbf{A} + \mathbf{Q}] \mathbf{x} \quad (2.36)$$

Et en répétant les étapes précédentes à partir de la récurrence (2.32) pour les valeurs $i = 2, 3, \dots, N$, on constate que $\Phi_i(\mathbf{x})$ est une fonction quadratique de son argument \mathbf{x} :

$$\Phi_i^0(\mathbf{x}) = \mathbf{x}^T \mathbf{P}_i \mathbf{x} \quad (2.37)$$

en substituant par $\Phi_N^0(\mathbf{x}) = \mathbf{x}^T \mathbf{P}_N \mathbf{x}$ l'équation (2.32) s'écrit

$$\Phi_{N+1}^0(\mathbf{x}) = \min_u \{ \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{Q} \mathbf{u} + \Phi_N^0(\mathbf{f}(\mathbf{x}, \mathbf{u})) \} \quad (2.38)$$

ce qui résulte en une loi de commande

$$\mathbf{u}^0 = -(\mathbf{R} + \mathbf{B}^T \mathbf{P}_N \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P}_N \mathbf{A} \mathbf{x}$$

(2.39)

où \mathbf{P}_N est solution d'une équation au différence de Riccati (Riccati Difference Equation-RDE) :

$$\mathbf{P}_{N+1} = \mathbf{A}^T \mathbf{P}_N \mathbf{A} - \mathbf{A}^T \mathbf{P}_N \mathbf{B} (\mathbf{R} + \mathbf{B}^T \mathbf{P}_N \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P}_N \mathbf{A} + \mathbf{Q} \quad (2.40)$$

Avec la condition initiale $\mathbf{P}_0 = \mathbf{S}$. La symétrie de la solution de l'équation de Riccati peut se vérifier par transposition de l'équation (2.40) ce qui conduit à une équation identique en fonction de \mathbf{P}_{N+1}^T et \mathbf{P}_N^T . Sachant que \mathbf{R} et \mathbf{Q} sont symétriques, que pour une matrice carrée non singulière on a $(\mathbf{G}^{-1})^T = (\mathbf{G}^T)^{-1}$, alors, l'équation (2.40) et sa transposée ont nécessairement la même solution, on a donc :

$$\mathbf{P}_N^T = \mathbf{P}_N \quad (2.41)$$

C'est-à-dire que la solution de l'équation de Riccati est symétrique. La fonction de coût (2.25) étant quadratique; elle est nécessairement non-négative et l'on déduit, à partir de (2.33) et (2.37) que \mathbf{P}_i doit satisfaire la condition de non-négativité $\mathbf{P}_i > 0$.

2.3.3 Stabilité de la commande RLQ

L'étude de la stabilité du système (2.24) sous l'action de loi de commande optimale (2.39) revient à l'étude de la stabilité du système

$$\mathbf{x}(t+1) = (\mathbf{A} + \mathbf{B} \mathbf{K}_N) \mathbf{x}(t) \quad (2.42)$$

où $\mathbf{K}_N = -(\mathbf{R} + \mathbf{B}^T \mathbf{P}_N \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P}_N \mathbf{A}$.

Le système linéaire (2.42) est stable si les valeurs propres λ_i de la matrice $(\mathbf{A} + \mathbf{B} \mathbf{K}_N)$ ont des modules strictement inférieurs à l'unité ($|\lambda_i| < 1$). Cette condition pour la stabilité de la commande RLQ à horizon fini N , n'est pas satisfaite pour toute les valeurs positives de R et Q ou de l'horizon N . Pour approcher ce problème Bitmead et al. [14], commencent par analyser les conditions de stabilité du problème optimal à horizon infini (2.26) et, en particulier

les conditions de convergence de l'équation algébrique de Riccati (2.40) pour le cas $N \rightarrow \infty$.

Les propriétés et conditions de convergence de l'équation de Riccati ont été amplement étudiées dans la littérature [57-60] qu'on peut résumer par les résultats suivants :

Résultat 1. Lorsque $N \rightarrow \infty$ l'équation (2.40) converge vers une solution unique $P_\infty = P$, symétrique et non-négative et s'écrit :

$$P = A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A + Q \quad (2.43)$$

si les conditions suivantes sont satisfaites :

1. Le système (A, B) est stabilisable.
2. Le système $(A, Q^{1/2})$ est détectable¹ ($Q = (Q^{1/2})(Q^{1/2})^T$)
3. $Q \geq 0$ et $R > 0$

Dans ces conditions la solution unique P de l'équation de Riccati pour $N = \infty$ est une solution stabilisante pour le système (2.24) et les valeurs propres de la matrice $A - B(R + B^T P B)^{-1} B^T P A$ se situent toutes à l'intérieur du cercle unité. Ceci peut aussi se vérifier en définissant la fonction de Liapunov :

$$V(x(t)) = \frac{1}{2} x^T(t) P x(t) \quad (2.44)$$

qui vérifie les conditions (1) et (3) du théorème (2.1). Alors :

$$\begin{aligned} \Delta V(x(t)) &= V(x(t+1)) - V(x(t)) \\ &= \frac{1}{2} x^T(t+1) P x(t+1) - \frac{1}{2} x^T(t) P x(t) \end{aligned} \quad (2.45)$$

En utilisant (2.41), avec $K_\infty = K = -(R + B^T P B)^{-1} B^T P A$ et en substituant dans (2.44):

$$\begin{aligned} \Delta V(x(t)) &= \frac{1}{2} x^T(t) [A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A + \\ &\quad A^T P B (R + B^T P B)^{-1} B^T P B (R + B^T P B)^{-1} B^T P A - \\ &\quad A^T P B (R + B^T P B)^{-1} B^T P A - P] x(t) \end{aligned} \quad (2.46)$$

En utilisant (2.42), cette dernière équation se simplifie en :

¹ Le système est dit détectable si les modes non observables appartiennent au sous-espace stable de la matrice A

$$\begin{aligned}
\Delta V(x(t)) &= \frac{1}{2} x^T(t) [-Q - A^T P B (R + B^T P B)^{-1} B^T P [A - B (R + B^T P B)^{-1} B^T P A]] x(t) \\
&= \frac{1}{2} x^T(t) \left\{ -Q - A^T \left(P B (R + B^T P B)^{-1} B^T P [I - B (R + B^T P B)^{-1} B^T P] A \right) \right\} x(t)
\end{aligned}
\tag{2.47}$$

Sachant que $Q \geq 0$, $R > 0$ et $P > 0$, et en déduisant à partir de (2.42) que la quantité $P[I - B(R + B^T P B)^{-1} B^T P]$ est aussi positive, donc l'expression entre accolades est strictement négative. Par conséquent, $\Delta V(x(t))$ est négative, le système (2.41) est asymptotiquement stable selon le théorème 2.1 de stabilité de Lyapunov.

Résultat 2. Dans le cas d'une optimisation sur un horizon fini N , une condition supplémentaire est ajoutée pour la convergence de l'équation (2.40) ; c'est : $P_0 = S \geq 0$.

Une technique utilisée par Bitmead et al. [14] pour l'analyse de la stabilité du RLQ à horizon fini consiste à exploiter les résultats précédents et définir une équation appelée *Fake Algebraic Riccati Equation* (FARE)

$$P_N = A^T P_N A - A^T P_N B (R + B^T P_N B)^{-1} B^T P_N A + \bar{Q}_N \tag{2.48}$$

obtenue par substitution dans (2.40) de

$$\bar{Q}_N = Q - (P_{N+1} - P_N) \tag{2.49}$$

L'avantage de cette écriture c'est qu'elle permet d'exploiter directement le '*Résultat 1*' cité plus haut et, de poser ainsi les conditions de la stabilité du régulateur à horizon fini.

Résultat 3. En considérant l'équation FARE (2.47) et l'équation (2.49) définissant la matrice \bar{Q}_N . Si $\bar{Q}_N \geq 0$, $R > 0$, (A, B) est stabilisable et $(A, \bar{Q}_N^{1/2})$ est détectable alors ; La solution P_N est stabilisante, c'est-à-dire que les valeurs propres de la matrice $A - B(R + B^T P_N B)^{-1} B^T P_N A$ se situent toutes à l'intérieur du cercle unité. Ce résultat est conditionné par la monotonie non croissante de la suite $\{P_j\}$, c'est-à-dire $P_{j+1} \leq P_j$ pour tout $j \geq 0$ à partir d'une valeur initiale $P_0 = P_0^T \geq 0$.

La manière de choisir la matrice P_0 pour garantir la monotonie non croissante de la suite $\{P_j\}$ a été proposée par J. B. Rawlings et al. [61] pour le cas d'une

commande linéaire contrainte à horizon fuyant et qui peut s'appliquer au régulateur quadratique linéaire selon [14].

2.8 La commande prédictive linéaire

Une extension logique du concept de la régulation quadratique linéaire à horizon fini était l'introduction des contraintes sur les vecteurs d'états et de commande et, la reformulation du problème sous la nouvelle appellation de *commande prédictive linéaire* ou *commande linéaire à horizon fuyant* [4]. Cependant, le développement théorique et l'analyse de la stabilité de ce nouveau concept connaîtrons des difficultés durant les années 80's dues essentiellement à la présence des contraintes dans la formulation du problème optimal [14] et un retour vers une formulation à horizon infini était nécessaire pour produire une loi de commande prédictive stabilisante.

2.8.1 Formulation du problème de la commande prédictive linéaire à horizon infini

Considérons un système linéaire discret décrit dans l'espace d'état par l'équation :

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (2.50)$$

tel que $\mathbf{x} \in \mathbb{R}^n$ représente le vecteur d'état , $\mathbf{u} \in \mathbb{R}^p$ est le vecteur de commande donnés à l'instant discret t . Les matrices \mathbf{A} et \mathbf{B} étant de dimensions appropriées, et le système (\mathbf{A}, \mathbf{B}) est considéré stabilisable. L'objectif est de réguler l'état du système de manière optimale vers l'origine. Dans ce cas l'optimalité est définie en termes d'une fonction de coût quadratique et d'un ensemble de contraintes imposées sur les vecteurs d'état et de commande. La fonction de coût à horizon infini est donnée par :

$$\Phi(t, \mathbf{x}(t)) = \sum_{k=t}^{\infty} \mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R} \mathbf{u}(k) \quad (2.51)$$

avec $\mathbf{Q} \geq 0$ et $\mathbf{R} > 0$ des matrices de pondération symétriques, tel que $(\mathbf{Q}^{1/2}, \mathbf{A})$ soit détectable. Les contraintes imposées sur l'état et la commande sont exprimées comme suit :

$$\mathbf{x}(t) \in \mathbf{X} = \{ \mathbf{x} \in \mathfrak{R}^n / \mathbf{H}\mathbf{x} \leq \mathbf{h} \} \quad (2.52)$$

$$\mathbf{u}(t) \in \mathbf{U} = \{ \mathbf{u} \in \mathfrak{R}^p / \mathbf{D}\mathbf{u} \leq \mathbf{d} \} \quad (2.53)$$

avec $\mathbf{h} \in \mathfrak{R}^{n_h}$ et $\mathbf{d} \in \mathfrak{R}^{n_d}$ les vecteurs définissant les niveaux des contraintes, n_h et n_d étant le nombre des contraintes et, \mathbf{H} et \mathbf{D} sont des matrices de distribution des contraintes sur l'état et la commande, respectivement.

Ce problème d'optimisation à horizon infini présente des difficultés numériques car, la présence des contraintes additionnelles (2.52) et (2.53) empêche la dérivation d'une série d'équations algébriques de Riccati indépendantes de l'état du système et, d'une loi de commande correspondante. Pour approcher cette situation plusieurs contributions furent avancées parmi lesquelles nous citons [61,62] qui proposent une paramétrisation de la commande $u_{t+j} = 0$ pour $j \geq N$ et ouvrir ainsi la possibilité de transformer le problème optimale à horizon infini en un problème sub-optimal à horizon fini. Sckaert et Rawlings [63] étendent le travail de Sznaier et Damborg [64] et démontrent la possibilité de résoudre le problème optimal contraint en un temps fini en se basant sur une paramétrisation $u_{t+j} = -Kx_{t+j}$ pour $j \geq N$ où, K représente le gain calculé pour le problème non contraint d'horizon N tel que décrit dans la section précédente.

La procédure numérique pour la résolution d'un problème d'optimisation quadratique à contraintes linéaires est connue sous l'appellation de *programmation quadratique* (quadratic programming algorithm – QP) [44-46]

2.8.2 Solution du problème de la commande prédictive à horizon infini par la méthode de Rawlings-Muske

Rawlings et Muske [61] traitèrent le problème d'optimisation contraint à horizon infini en paramétrisant l'entrée par $\{u_t, u_{t+1}, u_{t+2}, \dots, u_{t+N-1}, 0, 0, \dots\}$, en prenant une action de commande nulle après la N ème-étape. En adoptant cette paramétrisation, La fonction de coût (2.50) de la commande prédictive à horizon infini devient

$$\Phi(t, \mathbf{x}(t)) = \sum_{k=t}^{t+N-1} \mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R} \mathbf{u}(k) + \sum_{k=t+N}^{\infty} \mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) \quad (2.54)$$

Dans le cas où le système (2.50) est stable, c'est-à-dire que les valeurs propres de la matrice \mathbf{A} ont des modules strictement inférieurs à l'unité et en tenant compte que $\mathbf{u}_{t+j} = 0$ pour $j \geq N$, la sommation infinie dans (2.54) s'écrit :

$$\sum_{k=t+N}^{\infty} \mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) = \mathbf{x}^T(t+N) \left(\sum_{i=0}^{\infty} (\mathbf{A}^i)^T \mathbf{Q} \mathbf{A}^i \right) \mathbf{x}(t+N) \quad (2.55)$$

En définissant \mathbf{S} par :

$$\mathbf{S} = \sum_{i=0}^{\infty} (\mathbf{A}^i)^T \mathbf{Q} \mathbf{A}^i \quad (2.56)$$

Qu'on peut aussi écrire

$$\mathbf{S} = \mathbf{Q} + \sum_{i=1}^{\infty} (\mathbf{A}^i)^T \mathbf{Q} \mathbf{A}^i = \mathbf{Q} + \mathbf{A}^T \left(\sum_{i=0}^{\infty} (\mathbf{A}^i)^T \mathbf{Q} \mathbf{A}^i \right) \mathbf{A} \quad (2.57)$$

L'équation matricielle suivante est, alors, vérifiée

$$\mathbf{S} - \mathbf{A}^T \mathbf{S} \mathbf{A} = \mathbf{Q} \quad (2.58)$$

Des techniques sont disponibles pour la résolution de cette équation telle que celle proposée par Laub [65] citée par Muske dans [66].

Ainsi la somme infinie dans (2.54) est transformée en un coût sur l'état terminal et on écrit

$$\Phi(t, \mathbf{x}(t)) = \sum_{k=t}^{t+N-1} \mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R} \mathbf{u}(k) + \mathbf{x}^T(t+N) \mathbf{S} \mathbf{x}(t+N) \quad (2.59)$$

Maintenant, le problème de commande optimale définie par (2.50), (2.52), (2.53) et (2.59) peut-être formalisé sous forme d'un problème de programmation quadratique (PQ) et, traité en utilisant les procédures numériques disponibles pour la résolution de tels problèmes d'optimisation.

Chapitre 3

Stabilisation non linéaire par la commande prédictive

Dans le deuxième chapitre nous avons présenté la commande linéaire quadratique et la commande prédictive linéaire dans lesquelles un modèle d'état linéaire est utilisé pour prédire la dynamique du système sur un horizon fini ou infini. Nous avons aussi revu quelques méthodes théoriques pour l'analyse de la stabilité de ces stratégies de commande linéaire. La commande prédictive linéaire, en particulier, était devenue très populaire à cause de sa formulation qui impose de trouver une commande optimale en satisfaisant certaines contraintes sur l'état (la sortie) et la commande [2,4 et 67].

Cependant, plusieurs systèmes et processus industriels sont de nature non linéaire, ce qui a conduit à une reformulation de la commande prédictive dans un cadre non linéaire dans le but d'améliorer les prédictions et pour mieux atteindre les objectifs de la commande.

Dans ce chapitre nous présentons donc, la problématique de la commande prédictive non linéaire pour des modèles d'état discrets et nous passons en revue les méthodes théoriques permettant de garantir la stabilité de cette commande.

3.1 Principe et formulation mathématique de la commande prédictive non linéaire

En générale, le problème de la commande prédictive (linéaire et non linéaire) référencée modèle (model based predictive control – MPC) est formulé comme étant la résolution en temps réel (en ligne) d'un problème de commande optimale à boucle ouverte, en tenant compte de la dynamique du système et de certaines contraintes imposées sur les états et les variables de commande. La figure 3.1 illustre le principe de la commande prédictive pour un système discret.

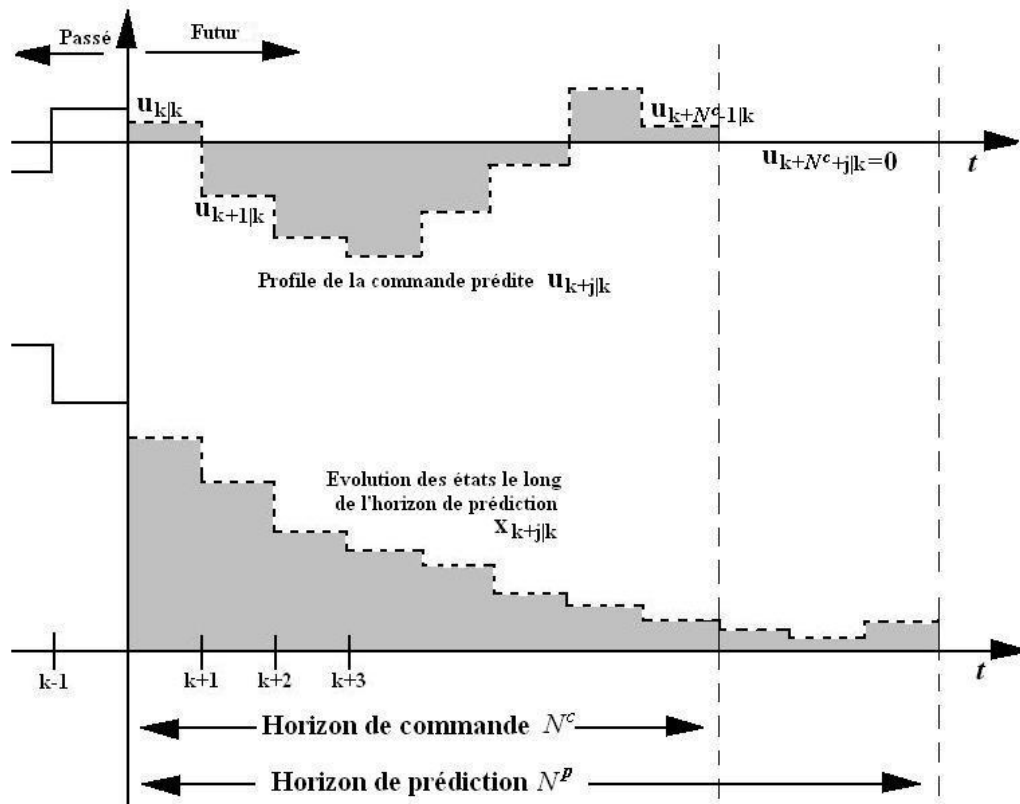


Figure 3.1 : Principe de la commande prédictive
Pour un système discret

En se basant sur les mesures des états obtenues à l'instant discret k , le système de commande fournit des prédictions sur le comportement futur du processus le long d'un horizon de prédiction N^p et détermine un profil de commande sur un horizon N^c ($N^p \geq N^c$) de telle sorte qu'une fonction de coût, préalablement spécifiée, soit optimisée en boucle ouverte en respectant les contraintes imposées au fonctionnement du système. En l'absence des perturbations et des erreurs de modélisation (modèle parfait), la résolution du problème d'optimisation pour des horizons infinis à l'instant discret k est suffisante pour commander le système pour tout instant futur. En pratique, ceci n'est pas possible à cause des perturbations et des différences entre le système réel et son modèle. Pour introduire une forme de rétroaction (feedback), la commande calculée en boucle ouverte à l'instant k est appliquée jusqu'à disponibilité des mesures des états à l'instant $k+1$. Les nouvelles mesures sont, alors, exploitées pour répéter la procédure – prédiction et optimisation – et

calculer une nouvelle commande pour l'instant $k+1$. La structure de base d'un système de commande prédictive est montrée dans la figure 3.2.

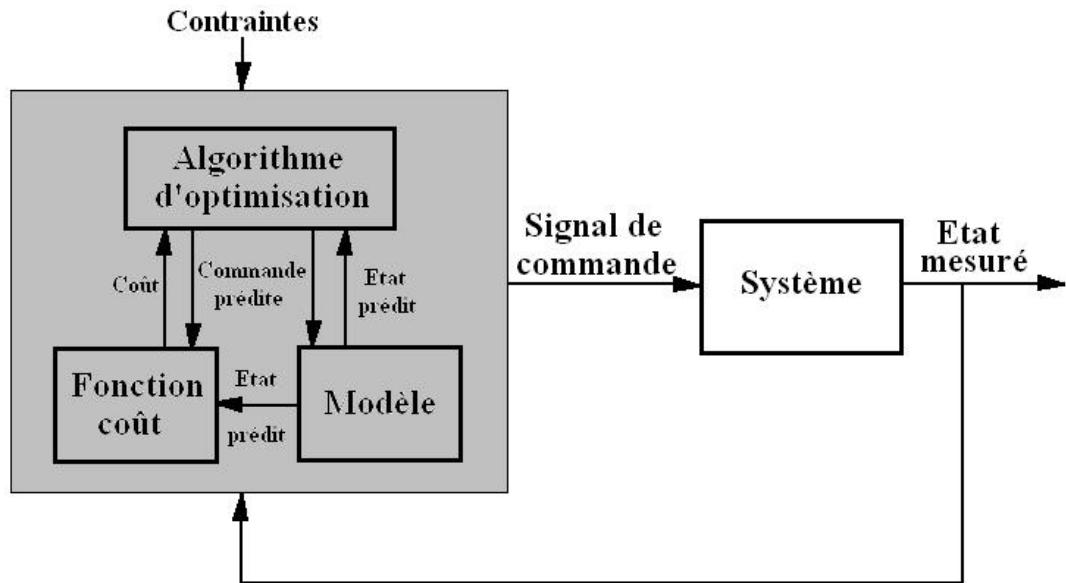


Figure 3.2 : Structure de base d'une commande prédictive

Dans la formulation mathématique de la commande prédictive non linéaire, nous considérons que le système dynamique objet de la commande est décrit par les équations discrètes

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (3.1)$$

Avec t l'indice du temps discret, $\mathbf{x}(t) \in \mathbf{X} \subset \mathfrak{R}^n$ représente le vecteur d'état, $\mathbf{u}(t) \in \mathbf{U} \subset \mathfrak{R}^p$ est le vecteur de commande. Le vecteur-fonction $\mathbf{f}(\cdot, \cdot)$ représente la fonction nonlinéaire de transition d'état. Dans ce qui suit, nous admettons les hypothèses suivantes :

Hypothèse 1. Les ensembles \mathbf{X} et \mathbf{U} appartiennent à la classe \mathbf{C} des ensembles compacts² $\mathbf{A} \subset \mathfrak{R}^q$ contenant l'origine $(0,0) \in \mathbf{X} \times \mathbf{U}$.

Les ensembles \mathbf{X} et \mathbf{U} définissent les contraintes sur l'état et la commande et sont définis par :

$$\mathbf{X} = \left\{ \mathbf{x} \in \mathfrak{R}^n / \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \right\} \quad (3.2)$$

de $\mathbf{S}(\delta)$ est dit compact s'il contient une sphère fermée $\mathbf{A} \subset \mathfrak{R}^{q^2}$ Par définition, un ensemble

$$\mathbf{S}(\delta) = \left\{ \mathbf{x} \in \mathfrak{R}^q : \|\mathbf{x}\| \leq \delta \right\} \text{ et centrée à l'origine ; } \delta \text{ rayon}$$

$$U = \left\{ \mathbf{u} \in \mathbb{R}^p / \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} \right\} \quad (3.3)$$

Avec $\mathbf{x}_{\min}, \mathbf{x}_{\max}$ et $\mathbf{u}_{\min}, \mathbf{u}_{\max}$ des vecteurs constants donnés.

Hypothèse 2. La fonction $f : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ est continue et deux fois dérivable, avec $f(0,0) = 0$.

Hypothèse 3. Nous assumons que les horizons de prédiction N^p et de commande N^c coïncident ; c'est-à-dire $N^p = N^c = N$.

L'objectif de commande est de déplacer l'état du système vers l'origine à partir d'un état initial $\mathbf{x}_0 = \mathbf{x}(t_0)$ moyennant une commande optimale $\mathbf{u}^0(t)$ calculée pour tout $t \geq t_0$ en minimisant une fonction de coût et, tel que les contraintes sur l'état et la commande définies par (3.2) et (3.3) soient respectées. Une manière possible pour stabiliser le système (3.1) serait de le linéariser autour de l'origine, et d'appliquer une loi de commande linéaire $\mathbf{u}(t) = \mathbf{K}\mathbf{x}(t)$ qui garantit la stabilité dans un voisinage de l'origine. Ce voisinage est, cependant, très restreint et les performances se détériorent lorsque l'état initial s'éloigne de l'origine. Dans le but d'élargir cette région de stabilité autour de l'origine et améliorer ainsi, les performances dynamiques, on peut suggérer l'utilisation d'une loi de commande optimale non linéaire à horizon infini HI $\mathbf{u}(t) = \gamma^{\text{HI}}(\mathbf{x}(t))$ obtenue en minimisant une fonction coût :

$$\Phi_{\text{HI}}(\mathbf{x}(t), \mathbf{u}(\cdot)) = \sum_{k=t}^{\infty} l(\mathbf{x}(k), \mathbf{u}(k)) \quad (3.4)$$

dans le cadre des contraintes définies par (3.1), (3.2) et (3.3). L'hypothèse suivante est, aussi, admise concernant la fonction $l(\mathbf{x}(k), \mathbf{u}(k))$ dans (3.4) :

Hypothèse 4. La fonction $l : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^+$ est continue et dérivable, avec $l(0,0) = 0$.

Les propriétés stabilisatrices d'un régulateur HI, pour des systèmes non linéaires discrets ont été analysées par Keerthi et Gilbert [68] qui introduisent la notion de l'ensemble admissible (admissible set).

Définition 3.1. L'ensemble admissible $\bar{\mathbf{X}}$ pour le système à boucle fermée $\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \gamma(\mathbf{x}(t)))$ est l'ensemble invariant qui est un domaine d'attraction pour l'origine et tel que $\mathbf{x}(t) \in \bar{\mathbf{X}} \Rightarrow \mathbf{x}(t+k) \in \bar{\mathbf{X}}$ et $\gamma(\mathbf{x}(t+k)) \in \mathbf{U}$ pour tout $k \geq 0$.

Définition 3.2. On dit que \bar{X} est un ensemble invariant et domaine d'attraction pour le système à boucle fermée $x(t+1) = f(x(t), \gamma(x(t)))$ si toute trajectoire commençant en \bar{X} reste dans \bar{X} et converge vers l'origine.

Dans leurs travaux Keerthi et Gilbert [68] démontrent que si l'état initial $x(t) \in \bar{X}^{HI}$ où \bar{X}^{HI} représente l'ensemble des états initiaux pour lesquels le problème (3.4) a une solution dans le cadre de (3.1) et des contraintes (3.2) et (3.3), et que si $\gamma^{HI}(x(t))$ représente la loi de commande optimale à horizon infini alors, \bar{X}^{HI} est un ensemble admissible pour le système bouclé $x(t+1) = f(x(t), \gamma^{HI}(x(t)))$ avec la contrainte sur l'état $x(t+N) \rightarrow 0$ pour $N \rightarrow \infty$.

3.2 Résultats sur la stabilité de la commande prédictive non linéaire

Le problème d'optimisation non linéaire à horizon infini HI n'a pas une solution analytique car il implique une infinité de variables et les méthodes développées pour les systèmes linéaires sont inapplicables. Cette situation nécessitait l'emploi de la théorie de stabilité de Lyapunov et, Keerthi et Gilbert étaient les premiers à employer la fonction coût $\Phi_{HI}(x(t), u(\cdot))$ comme fonction de Lyapunov pour établir la stabilité d'une commande prédictive référencée modèle pour des systèmes discrets non linéaires. Ce concept fut, par la suite, largement adopté par la communauté des chercheurs dans le domaine de la commande prédictive pour l'analyse de la stabilité [24].

On peut diviser les schémas de la commande prédictive non linéaire en quatre catégories principales, selon leurs approches du problème de la stabilité:

3.2.1. Commande prédictive à horizon infini

Dans cette approche la formulation de la fonction de coût est la suivante :

$$\Phi_N(x(k), u(\cdot)) = \sum_{k=t}^{t+N-1} l(x(k), u(k)) \quad (3.5)$$

Le problème d'optimisation est, ensuite, traité pour le cas limite $N \rightarrow \infty$ en tenant compte des contraintes (3.1), (3.2) et (3.3) pour tout les valeurs $k \geq t$. Keerthi et Gilbert [68,69] et Rawlings et al. [70] établissent les conditions sur $x(t)$, $f(\cdot, \cdot)$,

$l(\cdot, \cdot)$, \mathbf{X} et \mathbf{U} pour l'existence et la stabilité de la solution au problème (3.5). En plus des hypothèses 1 à 4 déjà formulées plus haut, ces conditions comportent ;

Hypothèse 5. Il existe une fonction de classe K $\beta: \mathbb{R}^+ \rightarrow \mathbb{R}^+$ tel que $0 \leq \beta(\|\mathbf{x}, \mathbf{u}\|) \leq l(\mathbf{x}, \mathbf{u}), \quad \forall (\mathbf{x}, \mathbf{y}) \neq (0, 0)$.

Hypothèse 6. La valeur optimale de la fonction coût $\Phi_N^o(\mathbf{x})$ est continue à l'origine $\mathbf{x} = 0$.

A partir des hypothèses 4 et 5, on peut conclure sur les propriétés suivantes de la fonction $\Phi_N^o(\mathbf{x})$ [62,71] :

1. $\Phi_N^o(\mathbf{x}) \geq 0$
2. $\mathbf{x} = 0 \Leftrightarrow \Phi_N^o(\mathbf{x}) = 0$
3. $0 \leq \beta(\|\mathbf{x}\|) \leq \Phi_N^o(\mathbf{x})$
4. $\Phi_{N+j}^o(\mathbf{x}) \leq \Phi_N^o(\mathbf{x}), \quad \forall j \geq 0$

Pour prouver la stabilité de la commande HI, la fonction coût $\Phi_N^o(\mathbf{x})$ est choisie comme fonction de Lyapunov et il suffit de démontrer que $\Phi_\infty^o(\mathbf{x})$ vérifie les trois conditions du théorème 2.1 (Chapitre 2) pour que le système (3.1) entraîné par la commande optimale $\mathbf{u}^o(t) = \gamma^{HI}(\mathbf{x}(t))$ soit asymptotiquement stable.

Sachant qu'il suffit, dans le cas nominal, de calculer la séquence de commande optimale $\mathbf{u}_{t,\infty}^o = \{\mathbf{u}^o(t), \mathbf{u}^o(t+1), \mathbf{u}^o(t+2), \dots\}$ une première fois à l'instant t pour obtenir les commandes pour tout instant $t' \geq t$. On peut donc écrire

$$\begin{aligned} \Phi_\infty^o(\mathbf{x}(t)) &= l(\mathbf{x}(t), \mathbf{u}^o(t)) + l(\mathbf{x}(t+1), \mathbf{u}^o(t+1)) + l(\mathbf{x}(t+2), \mathbf{u}^o(t+2)) + \dots \\ &= l(\mathbf{x}(t), \mathbf{u}^o(t)) + \Phi_\infty(\mathbf{f}(\mathbf{x}(t), \mathbf{u}^o(t)), \mathbf{u}_{t+1,\infty}^o) \\ &= l(\mathbf{x}(t), \mathbf{u}^o(t)) + \Phi_\infty(\mathbf{x}(t+1), \mathbf{u}_{t+1,\infty}^o) \\ &\geq l(\mathbf{x}(t), \mathbf{u}^o(t)) + \Phi_\infty^o(\mathbf{x}(t+1)) \end{aligned} \tag{3.6}$$

La dernière étape découle du principe de l'optimalité et de la séquence de commande. Ceci démontre que la fonction $\Phi_\infty^o(\mathbf{x})$ est non-croissante et, en tenant compte de l'hypothèse (5) et du fait que $\mathbf{u}^o(t) = \gamma^{HI}(\mathbf{x}(t))$ est une fonction de l'état $\mathbf{x}(t)$, alors :

$$\Phi_\infty^o(\mathbf{x}(t)) - \Phi_\infty^o(\mathbf{x}(t+1)) \geq \beta(\|\mathbf{x}(t)\|) \tag{3.7}$$

Ce résultat satisfait à la deuxième condition du théorème (2.1). Les conditions (1) et (3) du théorème sont satisfaites par les propriétés (3) et (2), respectivement, de la fonction $\Phi_{\infty}^{\circ}(x)$. La stabilité est donc prouvée pour le système (3.1) qui tend asymptotiquement vers l'origine sous l'action d'une commande prédictive à horizon infini.

Remarque 3.1 Etant donné la positivité de la fonction $\beta(\|x(t)\|)$ et que $\Phi_{\infty}^{\circ}(x)$ est minorée par zéro, on déduit que la fonction de Lyapunov $\Phi_{\infty}^{\circ}(x)$ est décroissante et minorée donc, convergente vers zéro pour $t \rightarrow \infty$.

D'après cette remarque et en tenant compte de (3.6), la fonction $l(x(t), u^{\circ}(t))$ converge aussi vers zéro lorsque $t \rightarrow \infty$. Ceci implique selon l'hypothèse (4) que $(x(t), u^{\circ}(t)) \rightarrow (0,0)$.

3.2.2. Commande prédictive à horizon fini avec contrainte d'égalité terminale

D'un point de vue théorique, il est désirable de pouvoir résoudre en temps réel le problème de commande optimale à horizon infini HI tel que décrit dans la sous-section 3.2.1. Cependant ce problème met en jeu un nombre infini de variables d'état et de commande et est, de ce fait, loin d'être soluble en temps réel pour des considérations théoriques et aussi, à cause des limites imposées par les capacités de calcul et de mémoire [6].

Une manière possible d'aborder ce problème fut proposée par Keerthi et Gilbert [68] en imposant que l'état $x(t)$ doit atteindre l'origine après N étapes de prédiction ; c'est-à-dire,

$$x(t + N) = 0 \quad (3.8)$$

Cette condition supplémentaire imposée à l'état prédit est connue sous l'appellation de *contrainte d'égalité terminale* ou *contrainte d'égalité sur l'état terminale* et, le problème (3.5) peut dans ce cas être résolu en tant que problème d'optimisation à horizon fini HF en tenant compte de la dynamique du processus exprimée par (3.1) et des contraintes (3.2), (3.3) et (3.8).

La séquence de commande optimale est dans ce cas :

$$u_{t,t+N-1}^{\circ} = \{u^{\circ}(t), u^{\circ}(t+1), \dots, u^{\circ}(t+N-1)\} \quad (3.9)$$

La valeur optimale de la fonction coût s'écrit

$$\Phi_N^o(\mathbf{x}(t)) = l(\mathbf{x}(t), \mathbf{u}^o(t)) + l(\mathbf{x}(t+1), \mathbf{u}^o(t+1)) + \dots + l(\mathbf{x}(t+N-1), \mathbf{u}^o(t+N-1)) \quad (3.10)$$

Et du moment qu'on impose $\mathbf{x}(t+N) = 0$, cela signifie qu'au-delà de l'étape $N-1$ les commandes en boucle ouverte s'annulent et nous avons nécessairement $\mathbf{u}(t+N) = 0$. Sachant que $l(0,0) = 0$ en vertu de l'hypothèse (4), l'équation (3.10) peut-être réécrite

$$\begin{aligned} \Phi_N^o(\mathbf{x}(t)) &= l(\mathbf{x}(t), \mathbf{u}^o(t)) + \dots + l(\mathbf{x}(t+N-1), \mathbf{u}^o(t+N-1)) + l(0,0) \\ &= l(\mathbf{x}(t), \mathbf{u}^o(t)) + \dots + l(\mathbf{x}(t+N-1), \mathbf{u}^o(t+N-1)) + l(\mathbf{x}(t+N), \mathbf{u}^o(t+N)) \\ &= l(\mathbf{x}(t), \mathbf{u}^o(t)) + \Phi_N(\mathbf{f}(\mathbf{x}(t), \mathbf{u}^o(t)), \mathbf{u}_{t+1,t+N}) \\ &= l(\mathbf{x}(t), \mathbf{u}^o(t)) + \Phi_N^o(\mathbf{x}(t+1), \mathbf{u}_{t+1,t+N}) \\ &\geq l(\mathbf{x}(t), \mathbf{u}^o(t)) + \Phi_N^o(\mathbf{x}(t+1)) \end{aligned} \quad (3.11)$$

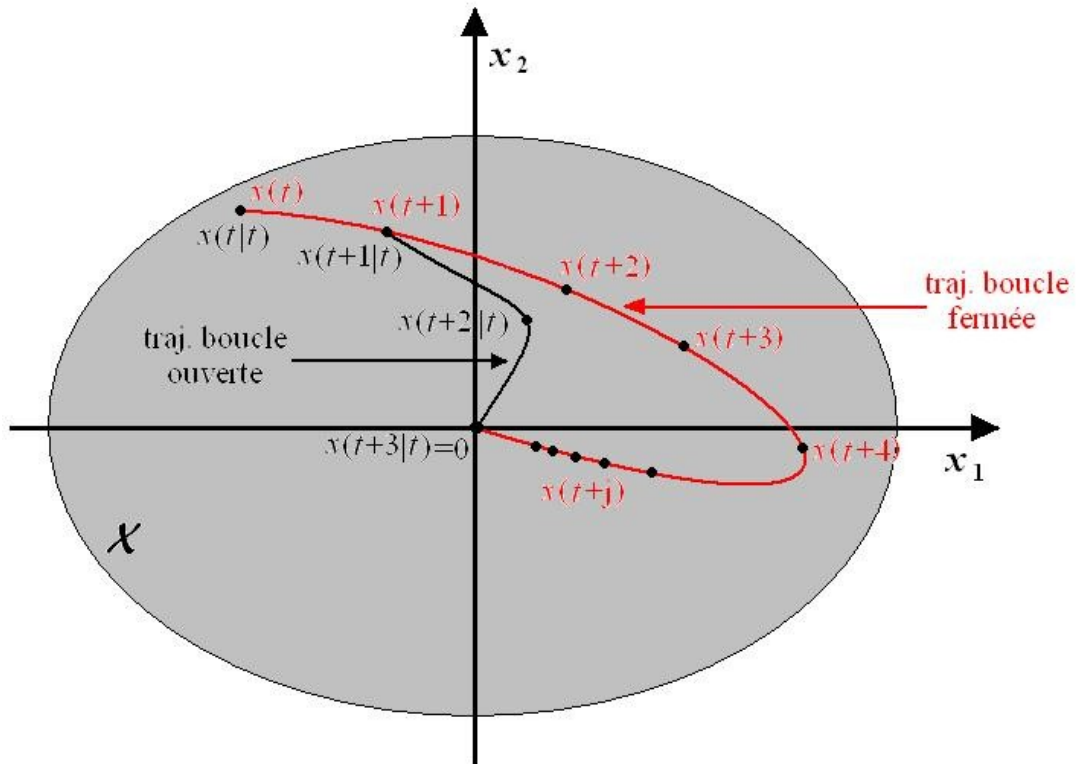


Figure 3.3 : Evolutions des trajectoires d'états en boucles ouverte et fermée lors d'une commande à HF avec contrainte d'égalité terminale

avec la séquence de commande $\mathbf{u}_{t+1,t+N} = \{\mathbf{u}^\circ(t+1), \dots, \mathbf{u}^\circ(t+N-1), 0\}$. A partir de (3.11), on déduit que la fonction $\Phi_N^\circ(\mathbf{x}(t))$ est non-croissante

$$\Phi_N^\circ(\mathbf{x}(t)) - \Phi_N^\circ(\mathbf{x}(t+1)) \geq l(\mathbf{x}(t), \mathbf{u}^\circ(t)) \geq 0 \quad (3.12)$$

Ceci ajouté aux propriétés (2) et (3) de la fonction $\Phi_N^\circ(\mathbf{x}(t))$ implique la satisfaction des conditions du théorème (2.1) et donc, le système (3.1) est asymptotiquement stable à l'origine sous l'action d'une commande prédictive à horizon fini HF avec contrainte d'égalité sur l'état terminal .

Contrairement au cas nominal¹ de la commande prédictive à horizon infini HI où la trajectoire dans l'espace d'état calculée en boucle ouverte coïncide avec celle du processus en boucle fermée, les trajectoires sont différentes dans le cas d'une commande à horizon fini HF avec contrainte d'égalité terminale. Pour illustrer cette situation prenons l'exemple d'un horizon fini $N=3$ (Figure 3.3). A l'instant t l'état du système est $\mathbf{x}(t) \in \mathbf{X}$ et la séquence de commande optimale en boucle ouverte est $\mathbf{u}_{t,t+2}^\circ = \{\mathbf{u}^\circ(t/t), \mathbf{u}^\circ(t+1/t), \mathbf{u}^\circ(t+2/t)\}$, à laquelle correspond la trajectoire $\mathbf{x}_{t,t+3} = \{\mathbf{x}(t/t), \mathbf{x}(t+1/t), \mathbf{x}(t+2/t), \mathbf{x}(t+3/t)\}$ avec l'état terminal $\mathbf{x}(t+3/t) = 0$. Les notations $\mathbf{x}(\cdot/t)$ et $\mathbf{u}^\circ(\cdot/t)$ indiquent l'évolution de l'état et de la commande en boucle ouverte.

A l'instant suivant $t+1$ l'état du processus est $\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}^\circ(t))$ sous l'action de la commande $\mathbf{u}^\circ(t) = \mathbf{u}^\circ(t/t)$. Le résultat de l'optimisation à l'instant $t+1$ est une séquence de commande optimale en boucle ouverte $\mathbf{u}_{t+1,t+3}^\circ = \{\mathbf{u}^\circ(t+1/t+1), \mathbf{u}^\circ(t+2/t+1), \mathbf{u}^\circ(t+3/t+1)\}$ qui est certainement différente de la séquence $\{\mathbf{u}^\circ(t+1/t), \mathbf{u}^\circ(t+2/t), 0\}$ déduite de l'optimisation à l'instant t et de la contrainte imposée sur l'état terminal. En conséquence on a $\mathbf{u}^\circ(t+1/t) \neq \mathbf{u}^\circ(t+1/t+1) \Rightarrow \mathbf{x}(t+2/t) \neq \mathbf{x}(t+2)$.

En conclusion, si la commande prédictive à horizon fini HF avec contrainte terminale est moins coûteuse en termes de temps de calcul et de mémoire, l'optimisation en ligne à chaque instant d'échantillonnage reste impérative même pour le cas nominal. La commande prédictive à horizon fini HF pour les processus

¹ Par cas nominal, il est entendu que le modèle est parfait et que les perturbations sont nulles.

décrits par des fonctions continues est traitée par Chen et Shaw [72] et Mayne et Michalska [73].

3.2.3. Commande prédictive à horizon fini avec contrainte d'inégalité terminale

Dans la stratégie de commande prédictive à horizon fini HF avec contrainte d'égalité sur l'état terminal (sous-section 3.2.2), il est exigé de trouver une solution optimale exacte pour un problème d'optimisation non linéaire. L'application de cet algorithme signifie l'obligation de calculer la solution en temps réel (en ligne) à chaque instant discret t . La stabilité et l'optimalité de cette stratégie de commande dépendent d'une part, de la satisfaction d'une contrainte d'égalité et d'autre part, de la dérivation d'une solution qui soit un minima global de la fonction de coût. Cette double exigence rend très difficile l'implémentation pratique de cette commande prédictive stabilisante et, pour causes [74]:

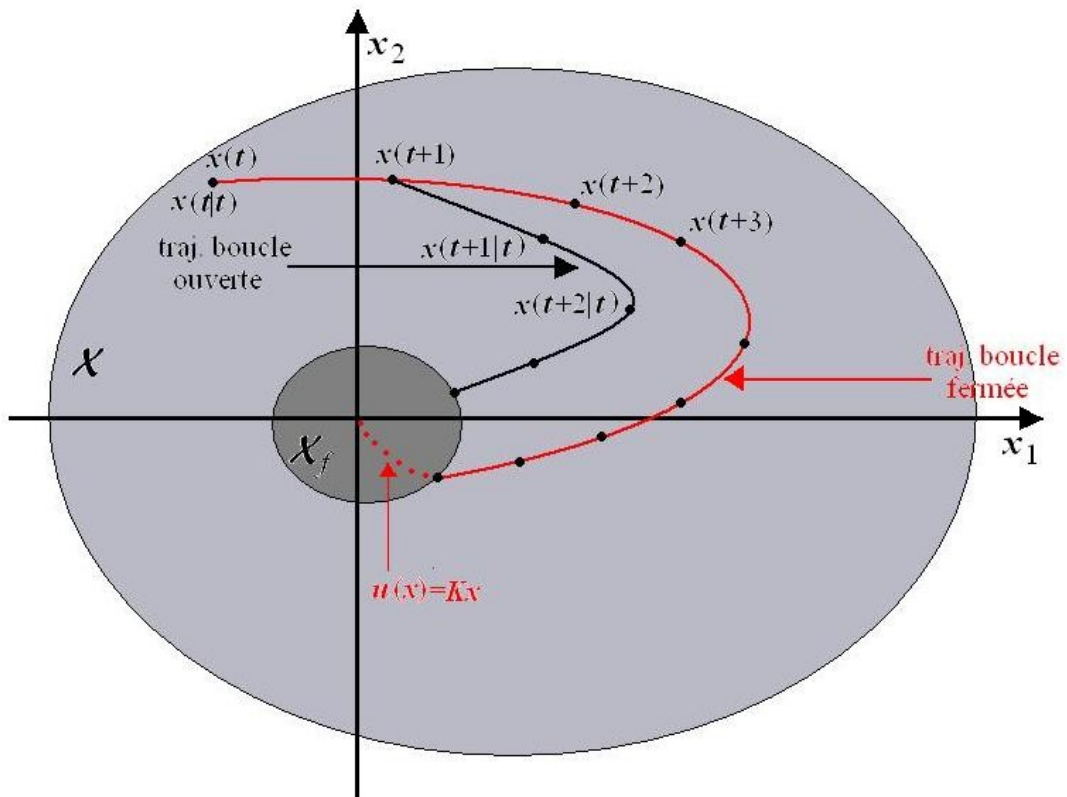


Figure 3.4 : Evolutions des trajectoires d'états en boucles ouverte et fermée lors d'une commande prédictive en *mode duale*

- a) La satisfaction exacte de la contrainte d'égalité non linéaire ne peut, en générale, être atteinte pendant un temps de calcul fini et l'interruption précoce de l'algorithme d'optimisation peut causer des erreurs qui peuvent être fatales pour la stabilité.
- b) Trouver une solution optimale globale pour un problème d'optimisation non linéaire n'est pas toujours possible et, l'algorithme est souvent gourmand en temps de calcul.

Pour contourner ces difficultés associées à la contrainte d'égalité sur l'état terminal $x(t + N) = 0$, une proposition a été avancée par Michalska et Mayne [41] pour les systèmes non linéaires continus et qui allège cette contrainte d'égalité en la remplaçant par une contrainte d'inégalité $x(t + N) \in \mathbf{X}_f$ et, l'emploi d'une commande locale stabilisante $\mathbf{K}_f(\cdot)$ à l'intérieur de \mathbf{X}_f . Le domaine \mathbf{X}_f est supposé être invariant sous l'action de $\mathbf{K}_f(\cdot)$.

Donc, une commande prédictive est appliquée une première fois pour déplacer le système à partir d'un état initial $x(t) \in \mathbf{X}$ vers le domaine \mathbf{X}_f qui contient l'origine pendant un temps fini T_f ($x(t + T_f) \in \mathbf{X}_f$), ensuite et à l'intérieur de \mathbf{X}_f une commande locale asymptotiquement stabilisante (ex: $\mathbf{K}_f(x) = \mathbf{K}x$) prends le relais pour amener l'état vers l'origine (Figure 3.4), d'où l'appellation de *commande prédictive en 'mode dual'* (Dual-mode predictive control).

Dans [74] Scokaert et al., analysent la stabilité de la stratégie en mode dual pour des systèmes non linéaires discrets. Donc, en considérant le système (3.1) avec les contraintes (3.2) et (3.3), le problème optimal consiste à minimiser (3.5) de sorte qu'à la N ème étape on a

$$x(t + N) \in \mathbf{X}_f \quad (3.13)$$

avec \mathbf{X}_f un sous-ensemble convexe et compact de \mathbf{X} et contenant l'origine.

A l'intérieur de \mathbf{X}_f on applique au processus une loi de commande stabilisante $u = \mathbf{K}_f(x)$; ($\mathbf{K}_f(\cdot) : \mathbf{X}_f \rightarrow \mathbf{U}$).

En plus des hypothèses (1), (2), (3) et (4), l'hypothèse (5) est modifiée pour devenir

Hypothèse 5bis. Il existe une fonction de classe K ; $\beta : \mathfrak{R}^+ \rightarrow \mathfrak{R}^+$ tel que $0 \leq \beta(\|(\mathbf{x}, \mathbf{u})\|) \leq l(\mathbf{x}, \mathbf{u})$, $\forall \mathbf{x} \notin \mathbf{X}_f$, pour toute valeur de $\mathbf{u} \in \mathbf{U}$.

Scokaert et al. posent une hypothèse supplémentaire sur la fonction $l(.,.)$:

Hypothèse 7. La fonction $l(.,.)$ est telle que : $l(\mathbf{x}, \mathbf{K}_f(\mathbf{x})) = 0 \quad \forall \mathbf{x} \in \mathbf{X}_f$.

Pour satisfaire cette hypothèse on pose

$$l(\mathbf{x}, \mathbf{u}) = \theta(\mathbf{x})L(\mathbf{x}, \mathbf{u}) \quad (3.14)$$

avec

$$\theta(\mathbf{x}) = \begin{cases} 0 & \text{si } \mathbf{x} \in \mathbf{X}_f \\ 1 & \text{autrement} \end{cases} \quad (3.15)$$

La fonction $L(\mathbf{x}, \mathbf{u})$ peut être de type quadratique $\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}$ avec \mathbf{R} et \mathbf{Q} des matrices positives.

Pour Scokaert et al., la stabilité à l'intérieur de \mathbf{X}_f est une propriété locale de la loi de commande $\mathbf{u} = \mathbf{K}_f(\mathbf{x})$ dans \mathbf{X}_f . Si l'état du système arrive à entrer dans le domaine \mathbf{X}_f en un temps fini, la convergence vers l'origine est garantie étant données les propriétés stabilisantes de \mathbf{K}_f ¹. Il ne reste, donc, qu'à démontrer que toutes les trajectoires dans l'espace d'état commençant en un point quelconque de \mathbf{X} finissent leurs parcours dans \mathbf{X}_f en un temps fini. Pour cela, nous partons de la propriété de compacité de \mathbf{X}_f qui implique qu'il contient une sphère fermée centrée à l'origine et de rayon $r > 0$ et tel que $\mathbf{x}(t) \notin \mathbf{X}_f \Rightarrow \|\mathbf{x}(t)\| > r$.

Selon l'hypothèse 5bis. $\mathbf{x} \notin \mathbf{X}_f \Rightarrow l(\mathbf{x}, \mathbf{u}) \geq \beta(\|\mathbf{x}\|) \geq \beta(r)$ (la fonction $\beta(.)$ étant non-décroissante). Puisque, selon l'hypothèse (7) $l(\mathbf{x}, \mathbf{K}_f(\mathbf{x})) = 0$ pour tout $\mathbf{x} \in \mathbf{X}_f$, il s'en suit que

¹ Michalska et Mayne [41] développent une technique qui permet de spécifier explicitement le domaine \mathbf{X}_f et la loi de commande $\mathbf{K}_f(.)$ pour des systèmes continus non linéaires et, cette méthode peut être étendue aux systèmes non linéaires discrets [74] and [75].

$$\begin{aligned}
\Phi_N^\circ(\mathbf{x}(t)) &= l(\mathbf{x}(t), \mathbf{u}^\circ(t)) + \dots + l(\mathbf{x}(t+N-1), \mathbf{u}^\circ(t+N-1)) \\
&= l(\mathbf{x}(t), \mathbf{u}^\circ(t)) + \dots + l(\mathbf{x}(t+N-1), \mathbf{u}^\circ(t+N-1)) + l(\mathbf{x}(t+N), \mathbf{K}_f(\mathbf{x}(t+N))) \\
&= l(\mathbf{x}(t), \mathbf{u}^\circ(t)) + \Phi_N(\mathbf{f}(\mathbf{x}(t), \mathbf{u}^\circ(t)), \mathbf{u}_{t+1,t+N}) \\
&= l(\mathbf{x}(t), \mathbf{u}^\circ(t)) + \Phi_N(\mathbf{x}(t+1), \mathbf{u}_{t+1,t+N}) \\
&\geq l(\mathbf{x}(t), \mathbf{u}^\circ(t)) + \Phi_N^\circ(\mathbf{x}(t+1))
\end{aligned} \tag{3.16}$$

Ou encore

$$\Phi_N^\circ(\mathbf{x}(t+1)) - \Phi_N^\circ(\mathbf{x}(t)) \leq -l(\mathbf{x}(t), \mathbf{u}^\circ(t)) \tag{3.17}$$

C'est-à-dire que la fonction $\Phi_N^\circ(\cdot)$ qui est non-croissante, décroît d'une valeur au moins égale à $l(\mathbf{x}(t), \mathbf{u}^\circ(t))$ à chaque instant discret t .

Si nous choisissons un entier \bar{N} tel que $\bar{N}\beta(r) > \Phi_N^\circ(\mathbf{x}(t))$ et si, l'état (en boucle ouverte) ne rentre pas dans \mathbf{X}_f à l'instant $N = \bar{N}$, cela signifie que :

$$\mathbf{x}(t+N) > r \Rightarrow l(\mathbf{x}(t+j), \mathbf{u}^\circ(t+j)) \geq \beta(r) \quad ; \quad j = 0, 1, 2, \dots, \bar{N} \tag{3.18}$$

En reprenant de manière itérative l'équation (3.17) pour $j = 0, 1, 2, \dots, \bar{N}$ on écrit

$$\begin{aligned}
\Phi_N^\circ(\mathbf{x}(t), \mathbf{u}^\circ(t)) - \Phi_N^\circ(\mathbf{x}(t+1), \mathbf{u}^\circ(t+1)) &\geq l(\mathbf{x}(t), \mathbf{u}^\circ(t)) \geq \beta(r) \\
\Phi_N^\circ(\mathbf{x}(t+1), \mathbf{u}^\circ(t+1)) - \Phi_N^\circ(\mathbf{x}(t+2), \mathbf{u}^\circ(t+2)) &\geq l(\mathbf{x}(t+1), \mathbf{u}^\circ(t+1)) \geq \beta(r) \\
&\cdot \\
&\cdot \\
&\cdot \\
\Phi_N^\circ(\mathbf{x}(t+\bar{N}-1), \mathbf{u}^\circ(t+\bar{N}-1)) - \Phi_N^\circ(\mathbf{x}(t+\bar{N}), \mathbf{u}^\circ(t+\bar{N})) &\geq \\
&l(\mathbf{x}(t+\bar{N}-1), \mathbf{u}^\circ(t+\bar{N}-1)) \geq \beta(r)
\end{aligned} \tag{3.19}$$

En additionnant ces inégalités, on obtient :

$$\Phi_N^\circ(\mathbf{x}(t)) - \Phi_N^\circ(\mathbf{x}(t+\bar{N})) \geq \bar{N}\beta(r) \tag{3.20}$$

Tenant compte de la positivité des deux fonctions $\Phi_N^\circ(\cdot)$ et $\beta(\cdot)$, (3.20) implique que $\Phi_N^\circ(\mathbf{x}(t)) \geq \bar{N}\beta(r)$, ce qui contredit l'hypothèse initiale sur le choix de \bar{N} pour que : $\bar{N}\beta(r) > \Phi_N^\circ(\mathbf{x}(t))$. On conclut qu'il existe un \bar{N} tel que $\mathbf{x}(t) \in \mathbf{X} \Rightarrow \mathbf{x}(t+\bar{N}) \in \mathbf{X}_f$.

3.2.4. Commande prédictive à horizon fini avec fonction de coût terminale (méthode de Parisini et al.) :

En remplaçant la contrainte d'égalité terminale $x(t+N) = 0$ par une contrainte d'inégalité $x(t+N) \in \mathbf{X}_f$, Michalscka et Mayne [41] proposent une commande prédictive en 'mode dual' où, la loi de commande non linéaire appliquée dans $\mathbf{X}_f^c \cap \mathbf{X}$ "passe la main" à une commande locale stabilisante $\mathbf{K}_f(\cdot)$ dès que l'état $x(\cdot)$ aurait franchi la frontière du domaine \mathbf{X}_f . La loi $\mathbf{K}_f(\cdot)$ est souvent une loi de commande linéaire calculée sur la base d'une linéarisation du système (3.1) autour de l'origine. Cette transition entre deux modes de commande ; l'un non linéaire à l'extérieur de \mathbf{X}_f et l'autre linéaire dans son intérieur complique l'algorithme et constitue, d'après certains auteurs [75], sa faiblesse principale. Une approche différente est, donc, proposée par Parisini et Zoppoli [76] et Parisini et al. [77] où la commande prédictive non linéaire est dérivée en modifiant le problème optimal à horizon fini HF par l'introduction d'une fonction de coût sur l'état terminal :

$$\Phi_N(x(t), u(\cdot), l_f(x(\cdot))) = \sum_{k=t}^{t+N-1} l(x(k), u(k)) + l_f(x(t+N)) \quad (3.21)$$

où $l_f(\cdot)$ est une fonction continue et dérivable ; $l_f : \mathfrak{R}^n \rightarrow \mathfrak{R}^+$ avec $l_f(0) = 0$. Dans cette approche, Parisini et al. [76, 77] dérivent une commande optimale stabilisante sans imposer une quelconque contrainte sur l'état terminal (égalité ou inégalité). Cette contrainte est, cependant remplacée par une fonction de coût $l_f(\cdot)$ qui pénalise l'état terminal et qui définit une région ou domaine d'*attraction* vers l'origine. La stabilité asymptotique de cette stratégie de commande est garantie moyennant la satisfactions des conditions suivantes :

- (i) Le système linéaire $x(t+1) = \mathbf{A}x(t) + \mathbf{B}u(t)$, obtenu par la linéarisation du système (3.1) au voisinage de l'origine, avec

$$\mathbf{A} = \left. \frac{\partial f(x, u)}{\partial x} \right|_{\substack{x=0 \\ u=0}} \quad \text{and} \quad \mathbf{B} = \left. \frac{\partial f(x, u)}{\partial u} \right|_{\substack{x=0 \\ u=0}}$$

est stabilisable.

- (ii) Il existe deux fonctions strictement croissantes ; $\beta : \mathfrak{R}^+ \rightarrow \mathfrak{R}^+$ et $\mu : \mathfrak{R}^+ \rightarrow \mathfrak{R}^+$ qui vérifient $\beta(0) = \mu(0) = 0$ et tel que :

$$\beta(\|(x, u)\|^2) \leq l(x, u) \leq \mu(\|(x, u)\|^2) \text{ pour tout } x \in \mathbf{X} \text{ et } u \in \mathbf{U}.$$

- (iii) La fonction $l_f(\cdot) \in H(a, \mathbf{P})$, où $H(a, \mathbf{P})$ représente une classe de fonctions définies par $\{l_f(\cdot) : l_f(x) = ax^T \mathbf{P}x\}$. Avec $a \in \mathfrak{R}, a > 0$ et $\mathbf{P} \in \mathfrak{R}^{n \times n}$ une matrice symétrique et positive.
- (iv) Il existe un ensemble compact $\mathbf{X}_o \subseteq \mathbf{X}$ contenant l'origine tel que pour toute sphère $S(\delta) \subset \mathbf{X}_o$ centrée à l'origine et de rayon δ , il existe un horizon de commande $M \geq 1$ pour lequel la séquence $\mathbf{u}_{t,t+M-1} = \{\mathbf{u}(t), \mathbf{u}(t+1), \dots, \mathbf{u}(t+M-1)\}$ calculée en boucle ouverte ($\mathbf{u}(t+i) \in \mathbf{U}, i = 0, 1, \dots, M-1$) entraîne le système (3.1) sur une trajectoire d'état admissible ($x(t+i) \in \mathbf{X}, i = 0, 1, \dots, M$) et finissant à l'origine quelque soit l'état initial $x(t) \in \mathbf{X}_o$.

Le théorème de stabilité de la commande prédictive non linéaire démontré par Parisini et al. est énoncé comme suit :

Théorème 3.1 : Si les conditions (i) à (iv) sont vérifiées, il existe, alors, un nombre positif $\tilde{a} \in \mathfrak{R}$ et une matrice positive et symétrique $\mathbf{P} \in \mathfrak{R}^{n \times n}$ tel que : pour tout $N \geq M$ et pour tout $a \geq \tilde{a}$ on a :

- (1) La commande prédictive $\gamma^\circ(x(t)) = \mathbf{u}^\circ(t)$, calculée par optimisation de (3.20), stabilise asymptotiquement l'origine qui est un point d'équilibre du système bouclé $f(x(t), \gamma^\circ(x(t)))$.
- (2) Il existe un réel $\rho > 0$ tel que $W(N, a, \mathbf{P}) = \{x \in \mathbf{X}, \Phi_N^\circ(x(t), N, a, \mathbf{P}) \leq \rho\}$ est un sous-ensemble invariant de \mathbf{X}_o et, est domaine d'attraction de l'origine ; c'est-à-dire que, pour tout $x(t) \in W(N, a, \mathbf{P})$, la trajectoire d'état générée par le régulateur prédictif reste entièrement contenue dans $W(N, a, \mathbf{P})$ et converge vers l'origine.

Ce théorème repose essentiellement sur la démonstration que le coût optimal $\Phi_N^\circ(x(t), N, a, \mathbf{P})$ est une fonction de Lyapunov dans \mathbf{X}_o pour le système (3.1) entraîné par la commande prédictive $\gamma^\circ(x(t)) = \mathbf{u}^\circ(t)$.

Puisque $x = 0$ est un point d'équilibre, on en déduit directement que si l'état initial se confond avec l'origine ($x(t) = 0$) alors ; $\gamma^\circ(x(t)) = \mathbf{u}^\circ(t) = 0$ et plus

spécifiquement ce sont tous les éléments de la séquence $u_{t,t+N-1}$ qui sont nuls par conséquent ;

$$\Phi_N^{\circ}(0) = \Phi_N^{\circ}(0, N, a, \mathbf{P}) = 0 \quad (3.22)$$

En plus, d'après la condition (ii), nous pouvons écrire

$$\Phi_N^{\circ}(\mathbf{x}(t), N, a, \mathbf{P}) \geq l(\mathbf{x}(t), \mathbf{u}^{\circ}(t)) \geq \beta(\|\mathbf{x}(t), \mathbf{u}^{\circ}(t)\|^2) \geq \beta(\|\mathbf{x}(t)\|) > 0 \quad (3.23)$$

Les résultats (3.22) et (3.23) correspondent, respectivement, aux conditions (3) et (1) du théorème de stabilité de Lyapunov (2.1). Il ne reste, donc, qu'à évaluer la quantité $\Phi_N^{\circ}(\mathbf{x}(t), N, a, \mathbf{P}) - \Phi_N^{\circ}(\mathbf{x}(t+1), N, a, \mathbf{P})$ pour satisfaire la condition (2) du théorème de stabilité et démontrer ainsi la stabilité asymptotique.

On commence par écrire :

$$\Phi_N^{\circ}(\mathbf{x}(t)) = l(\mathbf{x}(t), \mathbf{u}^{\circ}(t)) + \dots + l(\mathbf{x}(t+N-1), \mathbf{u}^{\circ}(t+N-1)) + \mathbf{a}\mathbf{x}^T(t+N)\mathbf{P}\mathbf{x}(t+N) \quad (3.24)$$

calculée pour une séquence de commande optimale

$$\mathbf{u}_{t,t+N-1}^{\circ} = \{\mathbf{u}^{\circ}(t), \mathbf{u}^{\circ}(t+1), \dots, \mathbf{u}^{\circ}(t+N-1)\}.$$

D'après la condition (iv) du théorème de Parisini et al. et pour un horizon $N \geq M$; nous avons nécessairement $\mathbf{x}(t+N) = 0$.

Pour l'instant $t+1$, nous pouvons choisir la séquence non-optimale $\mathbf{u}_{t+1,t+N} = \{\mathbf{u}^{\circ}(t+1), \mathbf{u}^{\circ}(t+2), \dots, \mathbf{u}^{\circ}(t+N-1), 0\}$ à laquelle correspond la trajectoire d'état $\{\mathbf{x}(t+1), \mathbf{x}(t+2), \dots, \mathbf{x}(t+N), \mathbf{x}(t+N+1)\}$. La valeur de la fonction coût est alors donnée par :

$$\Phi_N^{\circ}(\mathbf{x}(t+1)) = l(\mathbf{x}(t+1), \mathbf{u}^{\circ}(t+1)) + \dots + l(\mathbf{x}(t+N), \mathbf{u}^{\circ}(t+N)) + \mathbf{a}\mathbf{x}^T(t+N+1)\mathbf{P}\mathbf{x}(t+N+1) \quad (3.25)$$

Tenant compte du fait que $\mathbf{x}(t+N) = \mathbf{x}(t+N+1) = 0$ et que $\mathbf{u}(t+N) = 0$, on peut écrire à partir de (3.24) et en appliquant le principe de l'optimalité:

$$\begin{aligned} \Phi_N^{\circ}(\mathbf{x}(t)) &= l(\mathbf{x}(t), \mathbf{u}^{\circ}(t)) + \Phi_N^{\circ}(\mathbf{x}(t+1)) \\ &\geq l(\mathbf{x}(t), \mathbf{u}^{\circ}(t)) + \Phi_N^{\circ}(\mathbf{x}(t+1)) \end{aligned} \quad (3.26)$$

Ce qui démontre que la fonction $\Phi_N^{\circ}(\mathbf{x}(t), N, a, \mathbf{P})$ satisfait à la condition (2) du théorème (2.1). Le système bouclé $\mathbf{f}(\mathbf{x}(t), \mathbf{y}^{\circ}(\mathbf{x}(t)))$ est, par conséquent, asymptotiquement stable à l'origine. Dans la première version du théorème présentée par Parasini et Zoppoli [76], la stabilité en boucle fermée était garantie

sous certaines conditions mais, aucune méthode constructive n'était avancée pour le calcul des paramètres \tilde{a} et \mathbf{P} . C'est dans la deuxième version du théorème [77] que les paramètres sont explicitement déterminés ainsi que le domaine d'attraction $W(N, a, \mathbf{P})$.

Chapitre 4

Les réseaux de neurones non bouclés et la théorie de l'approximation

Ce chapitre est consacré aux réseaux de neurones non bouclés (feedforward neural networks) et leurs applications dans l'approximation des fonctionnelles non linéaires. Les concepts relatifs aux réseaux de neurones sont présentés et les aptitudes d'approximation des réseaux de neurones non bouclés sont analysées.

4.5 Les réseaux de neurones : histoire et concept

Le concept du réseau de neurones artificiels est directement inspiré du paradigme de traitement de l'information basé sur les règles de fonctionnement des systèmes nerveux biologiques et en particulier du cerveau humain. En fait, ce paradigme constitue un "arrière-plan biologique" pour le développement de nouveaux modèles et techniques pour la résolution des divers problèmes liés au traitement rapide de l'information, la reconnaissance des formes, la commande et les systèmes adaptatifs.

Ce paradigme définit une structure pour le traitement de l'information avec les caractéristiques suivantes [28] :

Architecture parallèle : Cette structure est constituée d'un grand nombre d'éléments de traitement ou processeurs –neurones– en connexion parallèle et activant ensemble pour l'accomplissement de tâches spécifiques.

Similarité des neurones : Les processeurs de base –neurones– d'un réseau de neurones sont tous identiques dans leur structure et sont mathématiquement formalisés par des fonctions algébriques non linéaires standards.

L'apprentissage : Dans les systèmes biologiques, le processeur de base –neurone– reçoit des signaux émanant d'autres

neurones voisins avant de produire, lui-même, une activité électrique qu'il transmet, sous forme d'impulsions, vers les neurones présents dans son voisinage. Le phénomène d'apprentissage s'effectue par un ajustement du mode de transmission de l'information de chaque neurone en fonction de son état d'activation. De même, dans un réseau de neurones artificiels, l'apprentissage s'accomplit par un ajustage adaptatif de certains paramètres multiplicatifs associés à chacune des interconnexions du réseau.

D'un point de vue historique, les premiers réseaux de neurones formels furent d'abord proposés et étudiés par les pionniers du domaine ; McCulloch et Pitts [78] (cité dans [28]) en se basant sur leurs connaissances en neurologie. Leur modèle du neurone fut un élément binaire à partir duquel peuvent se construire des fonctions logiques simples. Cependant, ce n'est qu'à partir de l'année 1958 que les réseaux de neurones s'attirèrent l'intérêt de la communauté scientifique, suite à l'introduction par Rosenblatt [79] (cité dans [80]) du concept du *perceptron* ; une architecture de réseaux à trois couches. Dans sa version originale, le perceptron de Rosenblatt est constitué de trois types d'unités : *sensorielle*, *d'association* et de *réponse*. Le perceptron était conçu pour des fonctions de classification, mais sa faiblesse principale résidait dans la condition de séparation linéaire imposée aux formes à classifier, en plus de son pouvoir limité à généraliser ses aptitudes de classification en dehors des exemples déjà traités durant la phase d'apprentissage [81] . Après une période de frustration durant les années soixante caractérisée, surtout, par un manque flagrant des ressources et subventions pour la recherche en réseaux de neurones, vint ensuite les décennies soixante-dix et quatre-vingt qui ont vu une re-émergence des activités de recherche dans le domaine et notamment suite aux travaux de Grossberg [82] qui développa plusieurs architectures tels que les réseaux ART (Adaptive Resonance Nets), Albus [83] qui introduira le fameux CMAC (Cerebellar Model Articulation Controller) qui avait l'avantage de pouvoir réaliser des approximations des fonctions hautement non-linéaires. Un algorithme d'apprentissage des réseaux de neurones ; la rétro-propagation, et qui deviendra, par la suite, très populaire était d'abord proposé par Webros [84] en 1974 cité et

repris ensuite par Rumelhart [81]. La technique de rétro-propagation était la solution-miracle du problème d'apprentissage des couches cachées dans un réseau multicouches. Notons aussi les contributions importantes de Kohonen [85] et de Hopfield [86] pour leurs travaux sur les réseaux de neurones bouclés (recurrent networks) et leurs applications aux problèmes de classification automatique et d'optimisation.

4.6 Du neurone biologique au neurone artificiel

A l'heure actuelle, la possibilité de cerner, dans ces moindres détails, la complexité des structures morphologiques et des phénomènes physiologiques du système nerveux humain reste toujours exclue. Toutefois, des études expérimentales sur des cellules isolées du système nerveux [82] ont permis de développer des idées et des théories assez générales sur les caractéristiques des cellules nerveuses – neurones – et des connexions intra-cellulaires, et ainsi de construire des modèles mathématiques approchés des structures neuroniques.

La cellule nerveuse, ou neurone, est l'élément fonctionnel de base du système nerveux. Le rôle du neurone consiste à traiter l'information qu'il reçoit des neurones voisins et à transmettre les résultats du traitement vers d'autres neurones. En conséquence, la démarche de création d'un modèle neuronique ou d'un neurone artificiel qui intègrent les caractéristiques du neurone biologique et qui permettent la simulation de son comportement doit commencer, d'abord et nécessairement, par une étude de ses propriétés morphologiques et physiologiques pour aboutir, enfin, à une description mathématique approchée du neurone.

4.6.1 Morphologie et physiologie du neurone biologique

Le neurone est formé d'un corps cellulaire ou soma, d'une multitude d'arborisations courtes et fines nommées dendrites, et d'un long prolongement sortant ; l'axone ou cylindraxe (Fig. 4.1.). Sur les dendrites se trouvent des synapses ; terminaisons des axones d'autres neurones. Notons que chaque axone se termine par plusieurs milliers de branches.

A l'état normal, le milieu intracellulaire a un certain potentiel de repos U_i variant entre -60 et -80mV et qui augmente (dépolarisation) ou diminue (hyperpolarisation) selon les variations de la conductibilité de la membrane cellulaire [87].

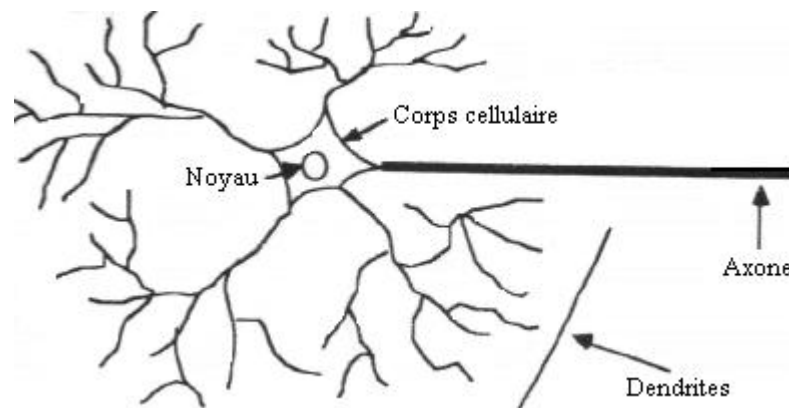


Figure 4.1 : Cellule neuronique

Les variations locales sur des portions de la membrane sont dues aux actions des synapses qui agissent en convertisseurs des influx parvenant d'autres neurones en suivant les axones. Il en résulte une variation localisée ΔU_i du potentiel intracellulaire accompagnée du passage d'une impulsion locale de courant de dépolarisation ($\Delta U_i > 0$) ou d'hyperpolarisation ($\Delta U_i < 0$) à travers la membrane, selon le type de la synapse. La variation locale du potentiel intracellulaire ΔU_i dans le sens de la dépolarisation ($\Delta U_i > 0$) est appelée potentiel post-synaptique d'excitation (PPSE), et dans le sens de l'hyperpolarisation ($\Delta U_i < 0$), potentiel post-synaptique d'inhibition (PPSI), et les synapses correspondantes sont soit, excitatrices ou inhibitrices.

Selon une hypothèse répandue, l'amplitude d'une impulsion de dépolarisation est conditionnée par les deux facteurs suivants :

- a) dimensions et position de la synapse excitée
- b) dimensions du dendrite réunissant la synapse donnée au corps neuronique

L'action exercée sur le neurone par les synapses est, donc, dépendante du 'poids' de la synapse qui peut varier d'une synapse à une autre, ce qui détermine, semble-t-il, l'aptitude du réseau neuronique à l'apprentissage.

La réaction du neurone aux nombreux signaux synaptiques incidents (excitateurs et inhibiteurs) est traduite par un processus régénératif, durant lequel la cellule génère une impulsion de sortie appelée potentiel d'action ou spike qui se propage le long de l'axone. De nombreuses expériences physiologiques ont permis d'établir que la génération d'un potentiel d'action suit une augmentation du potentiel intérieur du neurone au-delà d'un certain seuil U_0 , appelé limen du neurone. Des recherches au moyen de microélectrodes ont montré qu'entre autres régimes de fonctionnement le neurone génère un influx de sortie –spike– en réponse à plusieurs influx incidents superposés.

4.6.2 Modèle du neurone artificiel

En résumant les données physiologiques disponibles, l'on distingue deux étapes dans le fonctionnement du neurone: premièrement, la manipulation de l'information

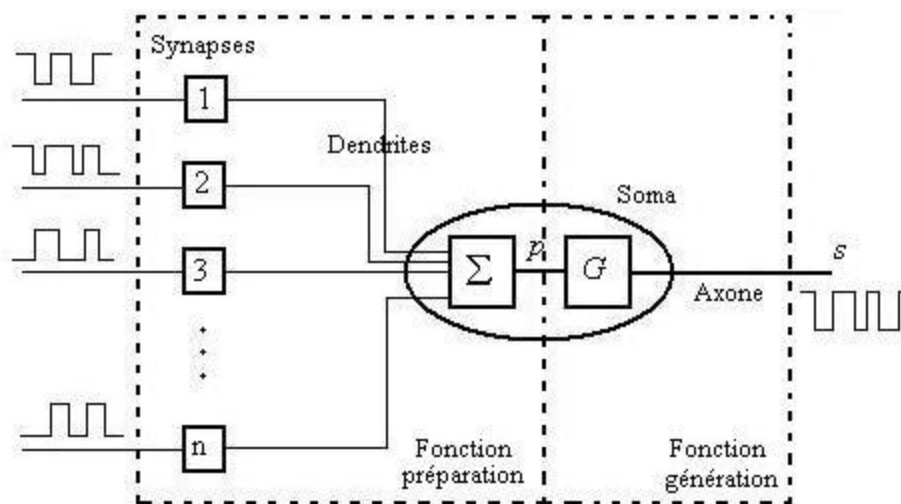


Figure 4.2. : Schéma fonctionnel du modèle neuronique

nerveuse incidente ou étape de préparation et, deuxièmement, la transmission de l'information traitée ou étape de génération.

En conclusion, le schéma fonctionnel du modèle neuronique peut-être représenté comme composé de deux unités : Une unité de préparation et un générateur (Fig.4.2.), et l'on résume les caractéristiques essentielles du neurone dans les points suivants :

- Le neurone est un élément multi-entrées (e_1, e_2, \dots, e_n) et à sortie unique (S).
- Les signaux d'entrée subissent deux transformations en cascade. La première (sommation temporelle) reflète la transformation de ces signaux en un paramètre $p = \sum(e_1, e_2, \dots, e_n)$ qui caractérise le degré d'excitation du neurone. La deuxième représente une action génératrice non-linéaire (limiteur) qui transforme le paramètre p en un signal de sortie $S = G(p)$ ou $S = G(\sum(e_1, e_2, \dots, e_n))$.
- Le neurone possède une aptitude à l'apprentissage, grâce à sa structure dendritique et aux poids synaptiques. Ainsi, le neurone "apprend" à produire (ou non) une forme impulsionnelle à sa sortie pour des distributions et des amplitudes particulières des excitations synaptiques incidentes.

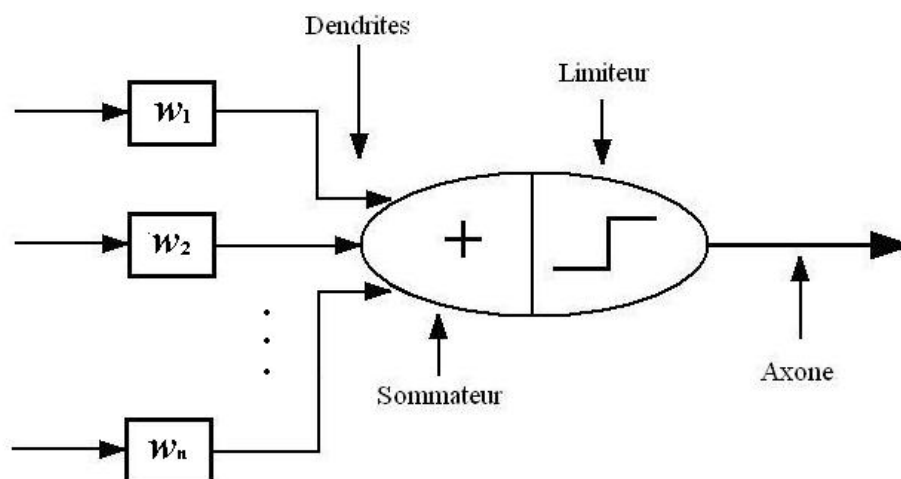


Figure 4.3 : Modèle de McCulloch-Pitts

Le modèle du neurone formel de McCulloch-Pitts, est alors aisément déduit et est constitué (Fig 4.3.) d'un sommateur et d'un limiteur.

Le sommateur calcule une combinaison linéaire des entrées (e_1, e_2, \dots, e_n) et d'un seuil θ . La sortie du sommateur étant appliquée directement au limiteur. Selon la nature du limiteur, la sortie peut prendre des valeurs bien déterminées.

Exemple :

$$G(p) = \begin{cases} 1 & p > 0 \\ -1 & p < 0 \end{cases} \quad (4.1)$$

Si les poids synaptiques sont données par (w_1, w_2, \dots, w_n) alors, la sortie du sommateur est :

$$p = \sum_{i=1}^n w_i e_i + \theta \quad (4.2)$$

Cette somme est appliquée directement au limiteur. La sortie du neurone est égale à +1 si $p > 0$ et, est égale à -1 si $p < 0$.

4.7 Architectures des réseaux de neurones et techniques d'apprentissage

Pris en tant qu'unité, le neurone ne représente aucun intérêt particulier en terme de capacité de calcul ou de traitement de l'information. Toutefois, l'interconnexion d'une multitude de neurones, sous forme de réseaux neuroniques, permet de créer des relations complexes entre plusieurs variables et de construire, ainsi, des entités possédant un potentiel énorme pour le calcul et le traitement.

La majorité des architectures neuroniques était initialement conçue pour des applications de reconnaissance de formes et de classification (discrimination). Cependant, ces mêmes architectures ont été appliquées avec succès dans l'approximation des fonctions et l'identification des systèmes dynamiques non-linéaires.

On distingue deux grandes classes de réseaux de neurones : les réseaux de neurones non bouclés qui regroupent principalement les perceptrons multicouches et les réseaux à fonction à base radiale et, les réseaux de neurones bouclés ou récurrents. Dans ce qui suit, nous limiterons notre étude à la première classe.

4.7.1 Les perceptrons multicouches

La structure-type d'un perceptron multicouches de \mathfrak{R}^m dans \mathfrak{R}^n peut-être représenté par le graphe orienté de la figure (4.4). Cette représentation montre une couche d'entrée comportant m nœuds-sources, une ou plusieurs couches cachées formées chacune, d'une multitude de neurones et d'une couche de sortie comportant n nœuds-sorties. Le signal-vecteur d'entrée $x \in \mathfrak{R}^m$ se propage le long du réseaux dans le sens entrée-sortie à partir de la couche d'entrée, en passant par toutes les couches cachées, et jusqu'à la couche de sortie où se génère un signal-vecteur de sortie $y \in \mathfrak{R}^n$. Ce réseau est dit *complètement connecté*, car chaque neurone de n'importe quelle couche – cachée ou de sortie – est relié à tous les neurones de la couche qui le précède.

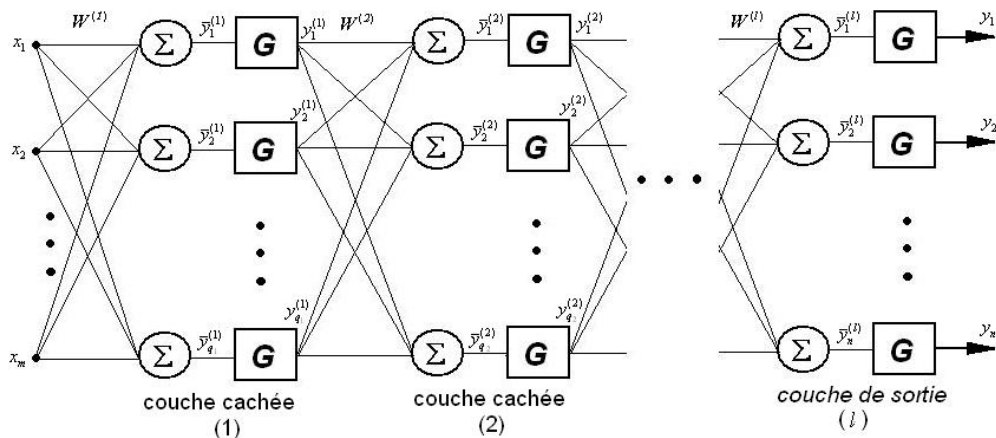


Figure (4.4) : Structure d'un perceptron multi-couches

Dans cette représentation les paramètres du réseau sont données comme suit :

- Le nombre total des couches cachées est égal à $(l - 1)$, la couche de sortie étant la $l^{i\grave{e}me}$.
- Le nombre des neurones dans la $k^{i\grave{e}me}$ couche ($k = 1, 2, \dots, l - 1, l$) est noté q_k avec $q_l = n$.
- Les matrices des coefficients de pondération sont notées $W^{(k)}$ de dimensions $(q_k \times q_{k-1})$. $W^{(1)}$ reliant la couche d'entrée à la première

couche cachée est une matrice ($q_l \times m$) et $W^{(l)}$ reliant la dernière couche cachée à la couche de sortie du réseau est de dimensions ($n \times q_{l-1}$).

- Pour le $j^{i\text{ème}}$ neurone de la $k^{i\text{ème}}$ couche, on note $\bar{y}_j^{(k)}$ la sortie du sommateur et $y_j^{(k)} = G(\bar{y}_j^{(k)})$ la sortie du neurone avec $G: \mathfrak{R} \rightarrow \mathfrak{R}$ la fonction d'activation.

La sortie $\bar{y}_j^{(k)}$ du sommateur du $j^{i\text{ème}}$ neurone de la $k^{i\text{ème}}$ couche cachée est, alors, donnée par :

$$\bar{y}_j^{(k)} = \sum_{i=1}^{q_{k-1}} W_{ji}^{(k)} G(\bar{y}_i^{(k-1)}) + b_j^{(k)} \quad j = 1, 2, \dots, q_k \quad (4.3)$$

où $W_{ji}^{(k)}$ dénote le poids synaptique de la connexion entre le neurone j de la couche (k) et le neurone i de la couche ($k-1$) et, $b_j^{(k)}$ le seuil appliqué au neurone j .

Pour simplifier la notation, on définit un opérateur, tel que

$$\begin{aligned} \Phi_k(\bar{\mathbf{y}}^{(k)}) &= \mathbf{y}^{(k)} \\ &= [G(\bar{y}_1^{(k)}), G(\bar{y}_2^{(k)}), \dots, G(\bar{y}_{q_k}^{(k)})]^T \end{aligned} \quad (4.4)$$

on obtient l'expression compacte pour le vecteur $\bar{\mathbf{y}}^{(k)}$:

$$\bar{\mathbf{y}}^{(k)} = \mathbf{W}^{(k)} \Phi_{k-1}(\bar{\mathbf{y}}^{(k-1)}) + \mathbf{b}^{(k)} \quad (4.5)$$

et la sortie du perceptron peut donc s'écrire sous forme d'une fonction $\Psi: \mathfrak{R}^m \rightarrow \mathfrak{R}^n$:

$$\Psi(\mathbf{x}) = \Phi_l[\mathbf{W}^{(l)} \Phi_{l-1}[\mathbf{W}^{(l-1)} \Phi_{l-2} \dots \dots \Phi_1[\mathbf{W}^{(1)} \cdot \mathbf{x} + \mathbf{b}^{(1)}] + \mathbf{b}^{(2)}] + \dots \dots + \mathbf{b}^{(l)}] \quad (4.6)$$

La fonction d'activation $G: \mathfrak{R} \rightarrow \mathfrak{R}$ dans les couches cachées est, typiquement, choisie parmi :

- (i) La tangente hyperbolique

$$G(z) = \alpha \tanh(\beta z) = \alpha \frac{1 - e^{-\beta z}}{1 + e^{-\beta z}}$$

- (ii) La fonction logistique ou sigmoïde

$$G(z) = \frac{1}{1 + e^{-z}}$$

L'importance des perceptrons pour les automaticiens provient, surtout, de leurs aptitudes à approximer toute fonction $f: \mathbf{K} \rightarrow \mathfrak{R}^n$ continue et bornée sur le

compact $K \subset \mathfrak{R}^m$ ($m \in \mathbf{N}$). C'est-à-dire, qu'il existe toujours un jeu de paramètres $W = (W^{(1)}, \dots, W^{(l)}, b^{(1)}, \dots, b^{(l)})$ associé au perceptron $\Psi(\cdot)$ tel que :

$$\|f(x) - \Psi(x)\| \leq \varepsilon \quad (4.7)$$

quel que soit le réel ε strictement positif [88].

Cette assertion est, cependant, purement 'existentielle' et non constructive. Car aucune méthode n'est avancée pour le choix des paramètres l et q_k pour atteindre l'objectif exprimé par l'inégalité précédente.

Cependant, l'estimation des paramètres W du perceptron $\Psi(\cdot)$ repose sur des techniques d'optimisation non-linéaires tels que les algorithmes d'apprentissage par prédiction de l'erreur [89-91] (prediction error learning algorithms) et, les algorithmes basés sur le principe de la rétro-propagation [27,92 et 93]. Pour un perceptron à structure fixe ou statique (l et q_k fixés à priori), l'objectif est de déterminer un jeu de paramètres W^* moyennant l'optimisation d'un critère de performance J fonction de l'erreur $\varepsilon = \|f - \Psi\|$ par application de la technique de rétro-propagation.

Soit, donc, un ensemble de couples $(x_t, f(x_t))$ sur $K \times \mathfrak{R}^n$ avec ($t=1,2,\dots,T$) et, que l'on cherche les éléments du vecteur W qui minimisent :

$$J(W) = \frac{1}{T} \sum_t \frac{1}{2} \|f(x_t) - \Psi(x_t, W)\|^2 \quad (4.8)$$

En posant $\varepsilon(x, f(x)) = f(x) - \Psi(x, W)$ pour représenter le vecteur de l'erreur de sortie, on aura alors :

$$J(W) = \frac{1}{T} \sum_t \frac{1}{2} \|\varepsilon(x_t, f(x_t))\|^2 \quad (4.9)$$

avec $\varepsilon_j(x_t, f(x_t)) = \varepsilon_j(t)$ l'erreur sur la sortie du neurone j du perceptron $\Psi(x, W)$ pour un vecteur d'entrée x_t .

Selon l'algorithme RP [92], la correction $\Delta w_{ji}^{(k)}(t)$ appliquée au poids $w_{ji}^{(k)}(t)$ est donnée par :

$$\Delta w_{ji}^{(k)}(t) = \eta \delta_j^{(k)}(t) G(\bar{y}_i^{(k-1)}(t)) \quad (4.10)$$

Où η représente le *pas d'apprentissage* et $\delta_j^{(k)}(t)$ le *gradient local* de la sortie du neurone j .

Ici, on distingue deux cas :

1. le neurone j appartient à la couche de sortie ($k=l$) :

$$\delta_j^{(l)}(t) = \varepsilon_j(t) G'(\bar{y}_j^{(l)}(t)) \quad (4.11)$$

2. le neurone j appartient à une couche cachée ($k < l$) :

$$\delta_j^{(k)}(t) = \left(\sum_q \delta_q^{(k+1)}(t) w_{qj}^{(k+1)}(t) \right) G'(\bar{y}_j^{(k)}(t)) \quad (4.12)$$

Si dans le premier cas, le calcul du gradient local est simplifié par la disponibilité de l'erreur $\varepsilon_j(t)$ pour chaque itération t , dans le second cas le calcul des corrections sur les poids nécessite la connaissance des poids et des gradients locaux de la couche suivante ($k+1$), d'où la dénomination *rétro-propagation* car, le processus d'apprentissage par mise à jour des poids synaptiques s'effectue par une propagation de l'erreur dans le sens inverse du flux de l'information, c'est-à-dire de la sortie vers l'entrée.

4.7.2 Réseaux à fonction à base radiale (RBF)

Une autre classe des réseaux de neurones non bouclés, autre que les perceptrons, est souvent utilisée dans les applications de classification et d'identification. Il s'agit des réseaux RBF [92,94-98].

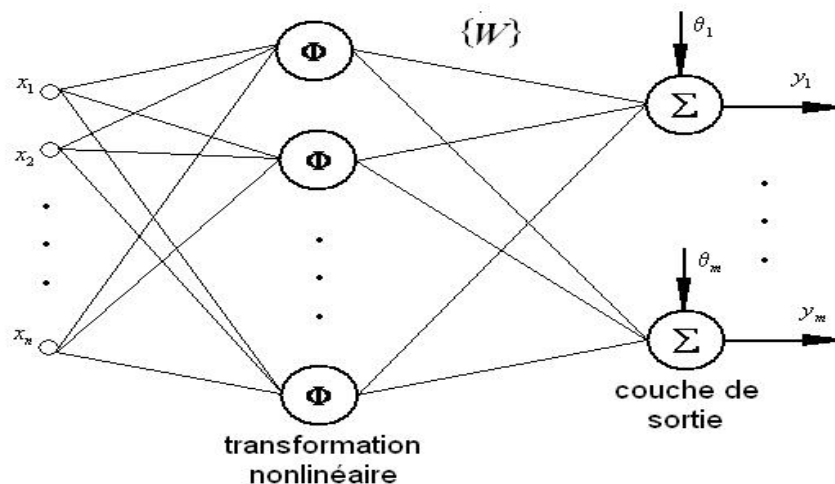


Figure (4.5) : Structure d'un réseau RBF

Fondamentalement, un réseau RBF est une structure à deux couches. La première est formée d'une multitude de nœuds contenant, chacun, un vecteur paramètre appelé *centre*. La sortie de chaque nœud n'est autre que la distance Euclidienne $\|x - c_j\|_{\mathfrak{R}^n}$ entre l'entrée du réseau $x \in \mathfrak{R}^n$ et le centre du nœud $c_j \in \mathfrak{R}^n$. Cette distance est ensuite passée comme argument à une fonction d'activation non linéaire à symétrie radiale $\phi: \mathfrak{R}^n \rightarrow \mathfrak{R}$. La deuxième couche est un sommateur linéaire (Figure 4.5).

En générale, la réponse d'un réseau RBF est une fonction $\Psi: \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ avec :

$$\Psi_j(x) = \sum_{i=1}^K w_{ij} \phi_i(\|x - c_i\|_{\mathfrak{R}^n}, \sigma_i) \quad 1 \leq j \leq m \quad (4.13)$$

les w_{ij} 's étant les poids synaptiques du réseau, σ_i 's sont des réels positifs appelés *largeurs* et K le nombre des nœuds.

Des choix typiques de la fonction d'activation $\phi(\cdot)$ sont :

- (i) La fonction 'thin-plate-spline'

$$\phi(z, \sigma) = z^2 \log(z)$$

- (ii) La fonction multi quadratique

$$\phi(z, \sigma) = (z^2 + \sigma^2)^{1/2}$$

- (iii) La fonction multi quadratique inverse

$$\phi(z, \sigma) = \frac{1}{(z^2 + \sigma^2)^{1/2}}$$

- (iv) La fonction Gaussienne

$$\phi(z, \sigma) = e^{-z^2/\sigma^2}$$

L'aptitude à l'approximation des réseaux RBF est comparable à celle d'un perceptron bi-couches. Ceci a été prouvé, en particulier, par Hartman et al. [98] et Park et Sandberg [99]. Dans la théorie de l'approximation par les réseaux RBF, il est souvent fait état que $\phi(\cdot)$ est, par hypothèse, continue et bornée [99]. On peut, toutefois, remarquer que les fonctions 'thin-plate-spline' et 'multi quadratique' ne remplissent pas cette condition ($\phi(z, \sigma) \rightarrow \infty$ pour $z \rightarrow \infty$), sans que leurs performances d'approximation ne se détériorent par rapport aux autres fonctions

d'activation [100]. Dans le chapitre 5, nous aborderons avec détails les techniques d'estimation des paramètres d'un réseau RBF.

4.8 L'approximation des fonctions non-linéaires par les réseaux de neurones

Dans cette section, nous présentons la théorie de l'approximation ainsi que l'application des réseaux de neurones comme "approximateurs" universels des fonctionnelles non-linéaires.

Pour commencer, la problématique de l'approximation peut-être formulée de la manière suivante : soit une application $f : K \rightarrow \mathfrak{R}^n$ continue sur le sous-ensemble compact¹ $K \subset \mathfrak{R}^m$. L'application f n'est pas, nécessairement, représentée de façon explicite mais on suppose qu'il existe un nombre fini de pair $(x_k, f(x_k))$ appartenant à $K \times \mathfrak{R}^n$; $k=1,2,\dots,s$. Où s est le nombre des échantillons entrée/sortie disponibles. Le problème c'est de trouver une représentation explicite \hat{f} de f , telle que \hat{f} est une approximation de f sur le compact $K \subset \mathfrak{R}^m$ avec une précision arbitraire. La fonction \hat{f} est souvent exprimée à l'aide de fonctions standards et d'un nombre fini de paramètres réels.

Souvent, le traitement de ce problème mathématique est approché par une démonstration de l'existence de la fonction \hat{f} . La construction de la l'approximation \hat{f} et l'estimation de ses différents paramètres font appel à d'autres techniques et méthodes.

4.8.1 Préliminaires mathématiques

Pour formaliser le problème d'approximation, introduisant d'abord quelques concepts et résultats théoriques [101].

Définition 4.1 Une norme sur un espace vectoriel X est une fonctionnelle positive $\|\cdot\|$ qui à chaque élément $x \in X$ associe le nombre $\|x\| > 0$, appelé *norme de l'élément x* , vérifiant les conditions suivantes :

$$1) \|x\| = 0 \Leftrightarrow x = 0 ;$$

¹ Cette condition de compacité imposé au sous-ensemble K implique qu'il est, à la fois, fermé et borné.

$$2) \quad \|\lambda x\| = |\lambda| \|x\|, \quad \lambda \in \mathfrak{R} \quad (\text{homogénéité de la norme})$$

$$3) \quad \|x + y\| \leq \|x\| + \|y\| \quad (\text{inégalité triangulaire})$$

L'espace vectoriel \mathbf{X} muni de la norme $\|\cdot\|$ est, alors, appelé *espace normé*.

Définition 4.2 Un espace vectoriel \mathbf{X} est un *espace métrique* si à tout couple d'éléments $x, y \in X$ est associé un nombre réel $\rho(x, y)$, *distance* de x à y , vérifiant les conditions :

$$1) \quad \rho(x, y) \geq 0; \quad \rho(x, y) = 0 \text{ si et seulement si } x = y;$$

$$2) \quad \rho(x, y) = \rho(y, x);$$

$$3) \quad \rho(x, y) \leq \rho(x, z) + \rho(z, y) \quad \text{pour tout } z \in X \quad (\text{inégalité triangulaire}).$$

La fonction $\rho: X \times X \rightarrow \mathfrak{R}$ s'appelle *métrique*. L'espace euclidien à n dimensions (\mathfrak{R}^n) est un exemple simple d'espace métrique.

Les espaces métriques dont les éléments sont des fonctions constituent des exemples plus complexes. L'espace $\mathbf{C}(K)$ qui regroupe toutes les fonctions réelles continues sur un compact $K \subset \mathfrak{R}^m$, dans lequel la distance entre deux fonctions f et g est définie comme suit :

$$\rho(f, g) = \sup_{t \in K} |f(t) - g(t)| \quad (4.14)$$

est un espace métrique. Si $K = [a, b] \in \mathfrak{R}$, on écrit $C[a, b]$ et la norme d'un élément $f \in C[a, b]$ est définie par

$$\|f\| = \sup_t \{|f(t)| : t \in [a, b]\} \quad (4.15)$$

Bien évidemment, le sous-espace P_n regroupant tous les polynômes $p_n(\cdot)$ de degré égale ou inférieur à n est un sous-espace de l'espace métrique $C[a, b]$.

Théorème 4.1 (Théorème d'approximation de Weierstrass) : Soit une fonction $f \in C[a, b]$. Alors ; pour tout $\varepsilon > 0$ il existe $N \in \mathbf{N}$ tel que $n > N$ entraîne $\|f(x) - p_n(x)\| < \varepsilon$ pour tous les $x \in [a, b]$.

Naturellement, La généralisation du théorème de Weierstrass aux fonctions multidimensionnelles $f: \mathfrak{R}^m \rightarrow \mathfrak{R}^n$ à l'aide des polynômes trouve plusieurs applications telles que la reconnaissance des formes et l'identification. Parmi ces généralisations, on trouve le théorème de Stone-Weierstrass qui repose sur

certaines définitions et propriétés des espaces métriques dont la plus importante est la *complétude*.

Définition 4.3 On dit qu'une suite $\{x_n\}$ de points d'un espace métrique \mathbf{X} est suite de Cauchy, si $\rho(x_m, x_n) \rightarrow 0$ lorsque $m, n \rightarrow \infty$, c'est-à-dire pour tout $\varepsilon > 0$ il existe N_ε pour lequel $\rho(x_m, x_n) < \varepsilon$ pour $m, n \geq N_\varepsilon$.

Définition 4.4 On dit qu'un espace métrique \mathbf{X} est *complet* si chaque suite de Cauchy $\{x_n\}$ est convergente, c'est-à-dire il existe un point $x_0 \in \mathbf{X}$ tel que $x_n \rightarrow x_0$.

L'espace métrique $\mathbf{C}(\mathbf{K})$ des fonctions continues sur le compact $\mathbf{K} \subset \mathbb{R}^m$ est un espace complet dans le sens où toute suite de Cauchy, dont les éléments sont des fonctions $f_n \in \mathbf{C}(\mathbf{K})$, converge vers une limite f_0 dans l'espace $\mathbf{C}(\mathbf{K})$.

Définition 4.5 On appelle *complétion* d'un espace métrique \mathbf{X} le plus petit espace métrique contenant l'espace \mathbf{X} . Le terme «plus petit» étant compris au sens que la *complétion* est contenue dans tout autre espace complet renfermant l'espace \mathbf{X} .

Définition 4.6 Soit \mathbf{X} un espace métrique. On dit qu'un ensemble $E \subset \mathbf{X}$ est *dense* dans une partie $X_0 \subset \mathbf{X}$ si pour tout $x \in X_0$ et tout $\varepsilon > 0$ il existe un point $z \in E$ satisfaisant $\rho(x, z) < \varepsilon$, ou autrement dit, pour tous $x \in X_0$ existe une suite $\{x_n\} \subset E$ tel que $x_n \rightarrow x$.

Définition 4.7 Un espace métrique \mathbf{X} est dit *séparable*, s'il contient une partie dense dénombrable.

A partir des définitions 4.6 et 4.7, il se déduit que tout espace métrique \mathbf{X} est dense dans sa complétion et que, si un espace métrique X_0 est sous-espace d'un espace métrique séparable \mathbf{X} , il est lui-même séparable.

L'espace métrique $C[a, b]$ est séparable car, pour ensemble dense dénombrable dans $C[a, b]$ on peut prendre l'ensemble de tout les polynômes à coefficients rationnels. En effet, d'après le théorème d'approximation de Weierstrass (Théorème 4.1) toute fonction continue dans $C[a, b]$ peut être uniformément approximée par un polynôme avec n'importe quelle précision donnée à l'avance. Toutefois, en opérant de petites variations on peut rendre rationnels tous les coefficients du polynôme sans affecter ; de façon significative, la précision de l'approximation. Donc, pour toute fonction continue $f(t)$, il existe un polynôme à

coefficients rationnels, aussi proche que l'on veut de $f(t)$ au sens de la métrique de l'espace $C[a, b]$.

Définition 4.8 Soit X un espace vectoriel. On dit que X est une *algèbre* si et seulement si, $\forall x, y \in X$, et $\forall \alpha \in \mathbb{R}$

- 1) $x + y \in X$;
- 2) $xy \in X$;
- 3) $\alpha x \in X$.

En plus, l'algèbre X est dite *commutative* si

$$xy = yx \text{ pour tout } x, y \in X.$$

L'algèbre X possède un *élément neutre* s'il existe un élément $e \in X$ tel que $xe = ex = x$, pour tout $x \in X$.

Dans le cas où X est un espace normé, la norme doit satisfaire

$$\|xy\| \leq \|x\| \|y\|$$

et l'espace X est appelé *algèbre normée*. Si l'algèbre normée X est munie d'une métrique et qu'elle est complète au sens de la définition 4.4 alors, elle est dite *algèbre complète* ou encore une *algèbre de Banach*.

Un sous-espace X_0 d'une algèbre X est appelé une *sous-algèbre* (de X) si X_0 est lui-même une algèbre.

En revenant maintenant à l'espace métrique $\mathbf{C}(\mathbf{K})$ des fonctions continues sur le compact \mathbf{K} et muni de la norme $\|f\| = \sup_t \{|f(t)| : t \in \mathbf{K}\}$, on peut vérifier que

$\mathbf{C}(\mathbf{K})$ est une *algèbre commutative de Banach* avec la fonction constante $f(t) = 1$ comme élément neutre.

La généralisation du théorème de Weierstrass (4.1) repose sur les concepts et définitions avancées plus haut. En effet le but est de trouver les conditions suffisantes que doit satisfaire une sous-algèbre E de $\mathbf{C}(\mathbf{K})$ pour que E soit dense dans $\mathbf{C}(\mathbf{K})$. Ces conditions sont résumées ici comme suit :

- 1) la sous-algèbre E *sépare les points* du compact \mathbf{K} , c'est-à-dire quels que soient $t, t' \in \mathbf{K}$ ($t \neq t'$), il existe une fonction $p \in E$ pour laquelle $p(t) \neq p(t')$.

- 2) quel que soit $t \in K$, il existe une fonction $p \in E$ pour laquelle $p(t) \neq 0$ (Exemple : la fonction identiquement égale à l'unité sur K).

L'énoncé du théorème de Stone-Weierstrass est le suivant :

Théorème 4.2 (Stone-Weierstrass) Soit K un espace métrique compact, et soit E une sous-algèbre de $C(K)$. Si E sépare les points de K et contient la fonction identiquement égale à l'unité sur K , alors E est dense dans $C(K)$.

Ce théorème fut la base d'une série de travaux démontrant les aptitudes d'approximation des réseaux de neurones multi-couches [88, 97-99].

Qu'ils soient des réseaux à fonctions sigmoïdes, ou des réseaux à fonction à base radiale (RBF), il suffit de montrer que l'ensemble de toutes les combinaisons linéaires finies de sigmoïdes ou de RBFs, respectivement, est une algèbre qui contient la fonction identiquement égale à l'unité et sépare les points d'un compact K pour aboutir au résultat d'existence d'un réseau de neurone capable d'approximer n'importe quelle fonction $f : K \rightarrow \mathbb{R}^n$ appartenant à $C(K)$, avec $K \subset \mathbb{R}^m$.

Donc, c'est à partir du théorème de Stone-Weierstrass que les aptitudes d'approximation avec n'importe quelle précision donnée à l'avance, des fonctions non-linéaires multidimensionnelles, par la classe des réseaux à fonctions sigmoïde et des réseaux à fonctions à base radiale (RBF) furent démontrés. Notons au passage que les propriétés d'interpolation des deux types de réseaux diffèrent par leurs caractères *globaux* pour les premiers et *local* pour les seconds.

4.8.2 Les perceptrons multicouches sont des approximateurs universels

Nous reprenons ici les résultats de Hornik et al. [88] qui établissent le fait que les perceptrons multicouches forment une classe de fonctions capables d'approximer virtuellement toute fonction avec une précision arbitraire donnée à l'avance.

Définition 4.9 Soit A^r l'ensemble de toutes les fonctions $h : \mathbb{R}^r \rightarrow \mathbb{R}$ de la forme $A(x) = W^T x + b$, où $W, x \in \mathbb{R}^r$ et $b \in \mathbb{R}$.

Définition 4.10 Pour toute fonction $G(\cdot): \mathfrak{R} \rightarrow \mathfrak{R}$ et pour tout entier $r \in \mathbb{N}$, on définit la classe de fonctions $\Sigma^r(G)$ tel que :

$$\Sigma^r(G) = \left\{ f: \mathfrak{R}^r \rightarrow \mathfrak{R} : f(\mathbf{x}) = \sum_{j=1}^q \beta_j G(A_j(\mathbf{x})), \mathbf{x} \in \mathfrak{R}^r, \beta_j \in \mathfrak{R}, A_j \in \mathbf{A}^r, q = 1, 2, \dots \right\}$$

La classe $\Sigma^r(G)$ décrit les perceptrons à couche unique avec $G(\cdot)$ comme fonction d'activation. Les paramètres β_j sont les poids du réseau liant la couche cachée au nœud de sortie et, q le nombre des neurones dans la couche cachée.

Définition 4.11 Pour toute fonction $G(\cdot): \mathfrak{R} \rightarrow \mathfrak{R}$ et pour tout entier $r \in \mathbb{N}$, on définit la classe de fonctions $\Sigma\Pi^r(G)$ tel que :

$$\Sigma\Pi^r(G) = \left\{ f: \mathfrak{R}^r \rightarrow \mathfrak{R} : f(\mathbf{x}) = \sum_{j=1}^q \beta_j \cdot \prod_{k=1}^{l_j} G(A_{jk}(\mathbf{x})), \mathbf{x} \in \mathfrak{R}^r, \beta_j \in \mathfrak{R}, A_{jk} \in \mathbf{A}^r, l_j \in \mathbb{N}, q = 1, 2, \dots \right\}$$

La classe $\Sigma\Pi^r(G)$ décrit les perceptrons à couches multiples.

Notons que la classe $\Sigma^r(G)$ définie par (4.11) est une sous-classe de $\Sigma\Pi^r(G)$ pour le cas particulier $l_j = 1$ pour toute valeur de l'indice j .

Définition 4.12 On définit par C^r l'espace de toutes les fonctions continues de $\mathfrak{R}^r \rightarrow \mathfrak{R}$.

Lorsque la fonction d'activation $G(\cdot)$ est continue, toutes les fonctions appartenant aux classes $\Sigma^r(G)$ et $\Sigma\Pi^r(G)$ sont continues et donc ; $\Sigma^r(G), \Sigma\Pi^r(G) \subset C^r$.

Définition 4.13 On dit qu'un sous-espace $C^r(\mathbf{K})$ de C^r est uniformément dense sur un compact dans C^r si pour tout compact $\mathbf{K} \subset \mathfrak{R}^r$, $C^r(\mathbf{K})$ est dense dans C^r au sens de la métrique $\rho_{\mathbf{K}}(f, g) = \sup_{t \in \mathbf{K}} |f(t) - g(t)|$.

Théorème 4.3 (Hornik et al. [88]) Soit $G(\cdot)$ une fonction quelconque continue et non constante de \mathfrak{R} dans \mathfrak{R} . Alors $\Sigma\Pi^r(G)$ est uniformément dense sur un compact dans C^r .

La démonstration de ce théorème est une application du théorème de Stone-Weierstrass. En effet, soit un compact quelconque $\mathbf{K} \subset \mathfrak{R}^r$. Il est évident que la

classe $\Sigma\Pi^r(G)$ est une algèbre au sens de la définition (4.8) pour toute fonction d'activation continue et mesurable $G(\cdot) : \mathfrak{R} \rightarrow \mathfrak{R}$.

Soit $x, y \in \mathbf{K}$ et $x \neq y$. Il existe, donc, une fonction $A \in \mathbf{A}^r$ tel que : $A(x) \neq A(y) \Rightarrow G(A(x)) \neq G(A(y))$. Ceci indique que la classe $\Sigma\Pi^r(G)$ sépare les points de \mathbf{K} . En plus, il existe des fonctions $G(A(\cdot))$ constantes et non nulles. Prenons, par exemple, $b \in \mathfrak{R}$ tel que $G(b) \neq 0$ et posons $A(x) = \theta \cdot x + b$ ($\theta = 0$). Donc, pour tout $x \in \mathbf{K}$ on a $G(A(x)) \equiv G(b) \neq 0$. Ce implique qu'il existe, au moins, un élément de la classe $\Sigma\Pi^r(G)$ qui ne s'annule nulle part sur le compact \mathbf{K} .

En vertu du théorème (4.2) de Stone-Weierstrass, la classe des fonctions $\Sigma\Pi^r(G)$ est dense dans l'espace $C^r(\mathbf{K})$. Le compact \mathbf{K} étant arbitraire, on conclue que $\Sigma\Pi^r(G)$ est dense dans C^r .

Ce résultat peut aussi s'exprimer de la façon suivante :

Pour tout compact $\mathbf{K} \subset \mathfrak{R}^r$, et pour toute fonction continue $f : \mathbf{K} \rightarrow \mathfrak{R}$ il existe un perceptron multicouche élément de la classe $\Sigma\Pi^r(G)$ tel que :

$$\left| f(\mathbf{x}) - \sum_{j=1}^q \beta_j \cdot \prod_{k=1}^{l_j} G(A_{jk}(\mathbf{x})) \right| < \varepsilon \quad (4.16)$$

et ceci quelque soit le réel $\varepsilon > 0$ arbitrairement petit.

La démarche précédente reste aussi valable pour l'espace $C^{r,s}$ des fonctions continues $f : \mathfrak{R}^r \rightarrow \mathfrak{R}^s$ [88], c'est-à-dire qu'il existe toujours un perceptron multi-entrées multi-sorties capable d'approcher une fonction $f \in C^{r,s}$ avec une précision arbitraire donnée à l'avance.

4.8.3 L'approximation des fonctions par les réseaux à fonctions à base radiale

Dans la sous-section précédente (4.4.2), nous avons présenté les résultats théoriques relatifs aux aptitudes d'approximation des fonctions continues par les perceptrons multicouches. Dans cette partie, nous reprenons quelques résultats concernant les réseaux de neurones à fonctions à base radiale ou réseaux RBF.

La représentation diagrammatique 4.5 montre q'un réseaux RBF est, fondamentalement, un perceptron avec une couche cachée unique caractérisée par des fonctions d'activation à symétrie radiale et ayant une réponse localisée aux signaux externes. La fonction d'activation le plus souvent utilisée pour les réseaux RBF est la fonction Gaussienne. On se penchera, donc, sur l'étude des propriétés d'approximation de cette catégorie des réseaux de neurones.

Définition 4.14 Soit $\Phi: \mathfrak{R}^r \rightarrow \mathfrak{R}$, une fonction à symétrie radiale, bornée, continue est intégrable tel que :

$$\int_{\mathfrak{R}^r} \Phi(\mathbf{x}) d\mathbf{x} \neq 0 \quad (4.17)$$

Définition 4.15 Soit $K \subset \mathfrak{R}^r$ un espace compact et $P \subset K$ un ensemble non vide d'éléments de K . On définit la classe des fonctions bornées $\gamma: K \rightarrow \mathfrak{R}$ tel que : $\Gamma = \{ \gamma(\cdot, \mu) : \mu \in P \}$

On assume dans ce qui suit que les fonctions $\gamma(\cdot, \mu)$ remplissent les conditions suivantes [102]:

- (i) pour tout $\mu_1, \mu_2 \in P$ fixes, il existe $\mu_3 \in P$ et une constante ζ_{μ_1, μ_2} tel que : $\gamma(\cdot, \mu_1) + \gamma(\cdot, \mu_2) - \gamma(\cdot, \mu_3) = \zeta_{\mu_1, \mu_2}$.
- (ii) Pour tout $\mathbf{x}_1, \mathbf{x}_2 \in K$ et $\mathbf{x}_1 \neq \mathbf{x}_2$, il existe $\mu \in P$ tel que $\gamma(\mathbf{x}_1, \mu) \neq \gamma(\mathbf{x}_2, \mu)$.

Nous assumons aussi que la fonction $\Phi: \mathfrak{R}^r \rightarrow \mathfrak{R}$ prend la forme $\Phi(\mathbf{x}) = e^{\gamma(\mathbf{x}, \mu)}$.

Définition 4.16 Soit la classe des fonctions continues $\Sigma^r(\Phi)$ tel que :

$$\Sigma^r(\Phi) = \left\{ g : \mathfrak{R}^r \rightarrow \mathfrak{R}, g(\mathbf{x}) = \sum_{j=1}^q \beta_j \Phi_j(\mathbf{x}), \beta_j \in \mathfrak{R} \right\}$$

La classe $\Sigma^r(\Phi)$ décrit les réseaux RBF avec $\Phi_j(\mathbf{x}) = e^{\gamma(\mathbf{x}, \mu_j)}$ comme fonctions d'activation. Les β_j 's étant les paramètres poids du réseau et $q \in \{1, 2, \dots\}$ représente le nombre des unités dans la couche cachée.

Théorème 4.4 [102] Pour tout compact $K \subset \mathfrak{R}^r$ la classe des fonctions $\Sigma^r(\Phi)$ est dense dans l'espace $C^r(K)$ des fonctions continues sur K .

C'est-à-dire , qu'étant donnée une fonction continue quelconque dans $C^r(\mathbf{K})$, un nombre réel arbitraire $\varepsilon > 0$, il existe $q \in \{1,2,\dots\}$, $\beta_1, \beta_2, \dots, \beta_q \in \mathfrak{R}$ et $\mu_1, \mu_2, \dots, \mu_q \in \mathbf{P}$ tel que :

$$\left| f(\mathbf{x}) - \sum_{j=1}^q \beta_j e^{\gamma(\mathbf{x}, \mu_j)} \right| < \varepsilon \quad (4.18)$$

La classe $\Sigma^r(\Phi)$ forme une algèbre car sous ensemble de l'espace des fonctions continues C^r et aussi en vertu de la condition (i) sur les fonctions les fonctions $\gamma(\cdot, \mu)$ cité plus haut. La propriété de séparation du compact \mathbf{K} par la classe $\Sigma^r(\Phi)$ découle directement de la propriété (ii). En effet, pour $x_1 \neq x_2$, il existe $\mu \in \mathbf{P}$ tel que $\gamma(x_1, \mu) \neq \gamma(x_2, \mu)$, ce qui implique $e^{\gamma(x_1, \mu)} \neq e^{\gamma(x_2, \mu)}$. La fonction $e^{\gamma(\mathbf{x}, \mu)}$ étant par elle-même élément de la classe $\Sigma^r(\Phi)$. Reste à prouver qu'il existe une fonction appartenant à $\Sigma^r(\Phi)$ qui ne s'annule pour aucun point $\mathbf{x} \in \mathbf{K}$. Sachant la fonction $\gamma(\mathbf{x}, \mu)$ bornée ; il existe donc $M \in \mathfrak{R}$ tel que $|\gamma(\mathbf{x}, \mu)| < M$, et l'on a $e^{\gamma(\mathbf{x}, \mu)} \neq 0$ pour tout $\mathbf{x} \in \mathbf{K}$ et tout $\mu \in \mathbf{P}$. En conséquence et en vertu du théorème de Stone-Weirstrass ; $\Sigma^r(\Phi)$ est dense dans $C^r(\mathbf{K})$. Le compact \mathbf{K} étant arbitraire, on conclue que $\Sigma^r(\Phi)$ est dense dans C^r .

Pour le cas particulier des fonctions Gaussiennes, c'est-à-dire pour $\gamma(\mathbf{x}, \mu_j) = \sigma_j \|\mathbf{x} - \mu_j\|^2$, où σ_j appartient au sous-ensemble non vide $S \subset (-\infty, 0)$, nous définissons l'ensemble produit $\mathbf{H} = S \times \mathbf{P}$ contenant les éléments $\mathbf{h} = (\sigma, \mu)$. Alors, la classe des fonctions bornées $\gamma: \mathbf{K} \rightarrow \mathfrak{R}$ introduite dans la définition (4.14) est redéfinie par : $\Gamma = \{ \gamma(\cdot, \mathbf{h}) : \mathbf{h} \in \mathbf{H} \}$.

En posant $\gamma(\mathbf{x}, \mathbf{h}) = \sigma \|\mathbf{x} - \mu\|^2$, il est claire que la condition (ii) exigée par le théorème (4.4) est vérifiée, c'est-à-dire que : pour tout $x_1, x_2 \in \mathbf{K}$ et $x_1 \neq x_2$, il existe $\mathbf{h} \in \mathbf{H}$ tel que $\sigma \|\mathbf{x}_1 - \mu\|^2 \neq \sigma \|\mathbf{x}_2 - \mu\|^2$. La condition (i) est aussi satisfaite en vérifiant la véracité de l'identité :

$$\begin{aligned} \alpha \|\mathbf{x} - \mu_1\|^2 + \delta \|\mathbf{x} - \mu_2\|^2 &= (\alpha + \delta) \|\mathbf{x} - \alpha(\alpha + \delta)^{-1} \mu_1 - \delta(\alpha + \delta)^{-1} \mu_2\|^2 \\ &\quad + \alpha \delta (\alpha + \delta)^{-1} \|\mu_1 - \mu_2\|^2 \end{aligned} \quad (4.19)$$

Cette identité est démontrée par addition et soustraction des termes $\delta\|\mathbf{x} - \boldsymbol{\mu}_1\|^2$ et $\alpha\|\mathbf{x} - \boldsymbol{\mu}_2\|^2$ au premier membre de l'équation précédente. On conclut par le corollaire suivant :

Corollaire 4.1 Pour toute fonction continue quelconque dans $C^r(\mathbf{K})$, un nombre réel arbitraire $\varepsilon > 0$, il existe $q \in \{1, 2, \dots\}$, $\beta_1, \beta_2, \dots, \beta_q \in \mathfrak{R}$, $\sigma_1, \sigma_2, \dots, \sigma_q \in (-\infty, 0)$ et $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_q \in \mathbf{P}$ tel que :

$$\left| f(\mathbf{x}) - \sum_{j=1}^q \beta_j e^{\sigma_j \|\mathbf{x} - \boldsymbol{\mu}_j\|^2} \right| < \varepsilon \quad (4.20)$$

Chapitre 5

Identification des systèmes dynamiques non linéaires par les réseaux de neurones

5.1 Modélisation et identification

Dans la théorie des systèmes, l'identification ou la modélisation sont des concepts fondamentaux pour la caractérisation d'un système. L'identification est, d'abord, une méthode pour définir une description mathématique d'un système ; un modèle qui exprime les caractéristiques du système par un opérateur P d'un espace d'entrée U vers un espace de sortie Y ($P : U \rightarrow Y$) et la définition de la classe \mathcal{P} des opérateurs à laquelle appartient P . Etant donnée, donc, une classe d'opérateurs \mathcal{P} avec $P \in \mathcal{P}$, le problème de l'identification est de déterminer une classe $\hat{\mathcal{P}} \subset \mathcal{P}$ et un élément $\hat{P} \in \hat{\mathcal{P}}$ tel que \hat{P} soit l'approximation la plus proche possible de P au sens d'un critère défini a priori. Le choix de la classe des modèles d'identification $\hat{\mathcal{P}}$ et de la technique à appliquer dans la détermination de \hat{P} , dépendent de plusieurs facteurs tels la précision de l'approximation, la simplicité et la fiabilité du modèle et son aptitude à représenter fidèlement le système [27].

5.1.1 Identification des systèmes statiques et dynamiques

Dans une situation *statique*, l'identification est utilisée dans un objectif de reconnaissance moyennant un opérateur ou fonction de décision $P : U \rightarrow Y$.

Dans les systèmes dynamiques, l'opérateur P décrivant le système est défini, par exemple, à partir d'un ensemble de couples entrée-sortie $(u_i(t), y_i(t)) \in U \times Y$ avec la variable temps $t \in [0, T]$. L'objectif de l'identification étant de trouver \hat{P} tel que :

$$\|\hat{y} - y\| = \|\hat{P}(u) - P(u)\| < \varepsilon, \quad u \in U \quad (5.1)$$

où ε est un réel strictement positif. Dans la formule précédente, $\hat{P}(u) = \hat{y}$ représente la sortie du modèle identifié et, donc $\hat{y} - y = e$ n'est autre que l'écart entre la sortie du modèle et la sortie mesurée sur le système réel.

5.1.2 Modèle fondamental et modèle empirique

Un modèle fondamental (white box model) pour un système physique (mécanique, électrique, chimique, thermique, . . .) est obtenu entièrement sur la base de lois et principes fondamentaux et, est globalement fiable et valable pour des applications de simulation, d'optimisation ou de commande. L'inconvénient majeur de ce type de modèle réside dans les difficultés rencontrées lors de sa construction ajoutées aux complexités inhérentes aux modèles fondamentaux pour des systèmes constitués d'une multitude de sous systèmes. Dans de tels cas, l'application rigoureuse des principes fondamentaux conduirait à des modèles représentés par des centaines d'équations différentielles et/ou algébriques ; une situation non souhaitable pour des applications de commande et d'optimisation en temps réel [103].

Lorsque l'information nécessaire à la construction d'un modèle fondamentale est incomplète et/ou que le modèle lui-même est assez compliqué et inadéquat à l'application en cours, l'alternative immédiate est celle d'un modèle empirique (black box model). Par définition, la modélisation empirique ou identification est le processus par lequel des informations mesurées sur les entrées et les sorties du système dynamique sont transformées en une relation décrivant le comportement du système et, en particulier, la prédiction de ses sorties pour des signaux d'entrées similaires à ceux utilisés dans la construction du modèle. Il reste, cependant, que le modèle empirique est généralement exploité dans un sous-espace restreint de l'espace d'entrée et que ses aptitudes à l'extrapolation sont considérablement dégradées hors de ce domaine d'opération. Cette faiblesse des modèles empiriques peut-être, partiellement, corrigée par des mises à jour continues du modèle au fur et à mesure que de nouvelles informations sont collectées à partir du système. Un autre facteur important et influent sur la fiabilité des modèles empiriques est directement lié à la qualité de l'information entrée-sortie utilisée lors de la construction du modèle. La génération des données (data

génération) en vu d'une identification d'un système ou d'un processus peut s'avérer problématique pour les raisons suivantes :

- (i) S'il est souvent possible de conduire des expérimentations sur des systèmes ou processus de laboratoire pour la collecte de l'information pertinente à leur identification, la situation est différente pour des systèmes en exploitation (industrie) où les coûts des tests peuvent être prohibitifs.
- (ii) Les difficultés dans le choix des signaux d'entrée capables d'exciter les différents modes de la dynamique du système et avoir, ainsi, des données riches en informations pour une meilleure identification et tel que les erreurs de modélisation seraient uniformément distribuées à travers tout le domaine d'exploitation.
- (iii) Des modèles construits et validés en se basant sur des données collectées en boucle ouverte peuvent s'avérer inadéquats dès qu'ils sont insérés dans une boucle fermée en présence d'un contrôleur.

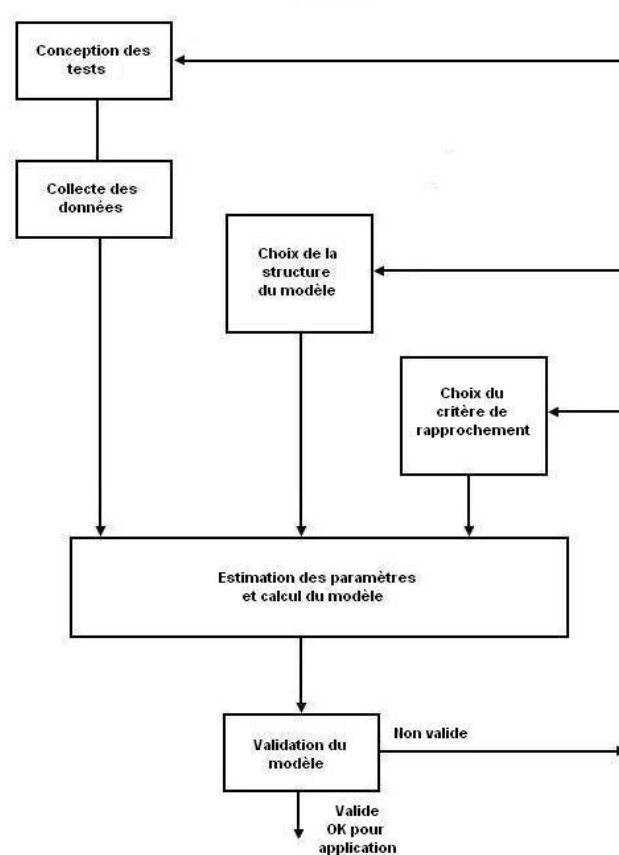


Figure 5.1 : Etapes de l'identification

5.1.3 Etapes pratiques de l'identification

Les étapes essentielles dans l'identification d'un système et la construction d'un modèle ont fait l'objet de plusieurs études [104-107] que nous résumons ici selon un organigramme représenté par la figure 5.1.

- *Conception des tests et collecte des données*

L'objectif de cette étape cruciale est d'obtenir les données expérimentales appropriées et nécessaires pour la construction du modèle. Cela passe par le choix des différentes variables d'entrée et de sortie à mesurer ainsi que des caractéristiques des signaux d'excitation à appliquer aux entrées du système. Dans ce contexte et selon le concept important de '*persistent excitation*' [105,108 et 109] qui pose une condition sur l'énergie portée par le signal d'excitation, celui-ci doit être suffisamment variable pour exciter le système et générer des données riches en information sur sa dynamique. A titre d'exemple, les signaux PRBS/RBS (Pseudo-Random Binary Sequence - Figure 5.1) ou suite binaire pseudo aléatoire sont souvent utilisés pour l'identification des processus linéaires. D'autres formes d'excitations peuvent être synthétisées par superposition d'une multitude de signaux tels que: bruit blanc, PRBS/RBS, suite de fonctions échelon, signal carré ou triangulaire ou sommes

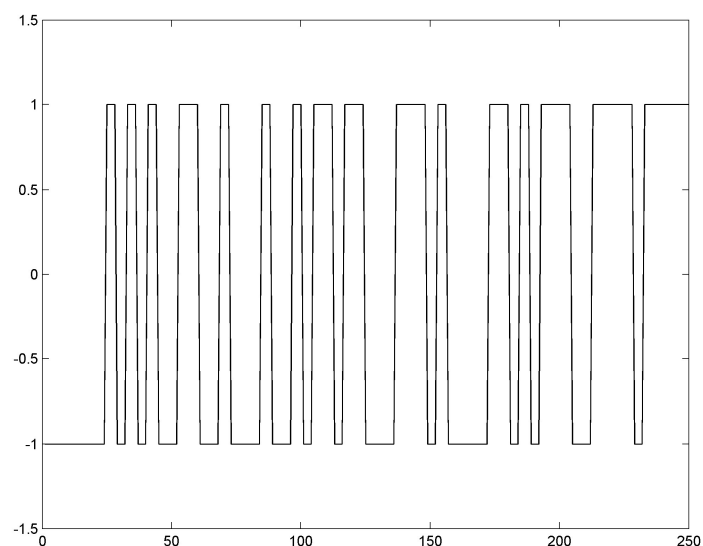


Figure 5.2 : Un signal PRBS

aléatoires de sinusoides. Pour un système multi entrées multi sorties (MIMO) avec $y: \mathbf{U} \rightarrow \mathbf{Y}$ et $\mathbf{U} \subset \mathfrak{R}^m, \mathbf{Y} \subset \mathfrak{R}^n$, un échantillon de donnée pris à l'instant t prends la forme d'un vecteur

$$\mathbf{z}(t) = [y_1(t), \dots, y_n(t), u_1(t), \dots, u_m(t)]^T \in \mathfrak{R}^{(n+m) \times 1} \quad (5.2)$$

L'empilement de N échantillons consécutifs forme une matrice

$$\mathbf{Z}_N = \begin{bmatrix} \mathbf{z}(1) \\ \mathbf{z}(2) \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{z}(N) \end{bmatrix} \in \mathfrak{R}^{N \times (n+m)} \quad (5.3)$$

- *Choix du modèle*

Le choix de la structure du modèle peut être subdivisé en trois sous problèmes :

- (i) Choix du type du modèle entre fondamental (White Box) et empirique (Black Box), et entre une structure linéaire ou non linéaire.
- (ii) Fixation des dimensions du modèle et choix de sa classe. Ceci implique le choix des variables à être utilisées dans la construction du modèle ainsi que de l'ensemble ou classe $\hat{\mathbf{P}}$ dans laquelle sera spécifié le modèle $\hat{y}(\cdot)$.
- (iii) Paramétrisation du modèle de telle sorte que les paramètres peuvent être calculés et convergent vers des valeurs numériques acceptables en utilisant les algorithmes d'estimation disponibles.

Concernant le choix entre une structure linéaire ou non linéaire du modèle, cela dépend à la fois de l'application et de la nature du système lui-même. Si le système est linéaire ou bien faiblement non linéaire, le choix se fera alors dans la catégorie des modèles à structures linéaires avec tous les avantages de simplicité et de disponibilité d'outils d'analyse. Si, par contre, le système exhibe un comportement fortement non linéaire ou que son fonctionnement s'étend sur un large domaine tel que son caractère non linéaire prend de l'influence, dans ce cas le choix doit porter sur un modèle à structure non linéaire. Dans certaines applications, des modèles linéaires sont assez satisfaisants pour décrire des systèmes non linéaires car, ces derniers sont astreints à évoluer aux voisinages

immédiats d'un point de fonctionnement fixe, ce qui rend l'hypothèse de la linéarité justifiée et acceptable.

- *Choix du critère de rapprochement*

Il s'agit de définir une fonction coût ou un critère de rapprochement pour mesurer l'écart entre le modèle et le système réel en utilisant des données expérimentales. Pour un système discret ce critère peut prendre la forme suivante :

$$J_N(\theta) = \sum_{k=1}^N f(e(k, \theta)) \quad (5.4)$$

Où $e(k, \theta) = y(t) - \hat{y}(t, \theta)$ représente l'erreur de modélisation, $f(\cdot)$ est souvent une fonction quadratique, θ le vecteur des paramètres du modèle et N le nombre total des échantillons disponibles.

Le critère $J_N(\cdot)$ est ensuite utilisé pour calculer l'ensemble des éléments de θ selon

$$\theta^* = \arg\{\min_{\theta}(J_N(\theta))\} \quad (5.5)$$

C'est donc un problème d'optimisation (minimisation) du critère de rapprochement $J_N(\cdot)$ par rapport au vecteur des paramètres θ et qui est souvent traité par les méthodes de *minimisation de l'erreur de prédiction* (prediction error minimisation-PEM).

- *Estimation des paramètres*

Cette étape est consacrée à la résolution du problème d'optimisation défini plus haut pour le calcul des valeurs numériques des paramètres de la structure du modèle choisi. Il convient, cependant, de distinguer deux situations lors de l'application d'un algorithme PEM : Soit la structure du modèle est *fonction non linéaire par rapport aux paramètres*, ou bien *fonction linéaire par rapport aux paramètres*.

- (i) Méthode des moindres carrés (Least Squares LS) : c'est la technique d'estimation utilisée dans le cas d'une structure linéaire
- (ii) Méthode des moindres carrés récursifs (Recursive Least Squares RLS) : appelée aussi méthode non linéaire des moindres carrés et s'applique dans les situations de paramétrisation non linéaire. Dans cette situation le minimum du critère quadratique de rapprochement

(fonction coût) ne peut être obtenu analytiquement. Des algorithmes numériques de recherche sont mis en œuvre tel : l'*algorithme du gradient* ou l'*algorithme de Gauss-Newton*.

(iii) Estimation par filtre de Kalman (KF) et filtre étendu de Kalman (EKF).

- *Validation du modèle*

Après l'estimation de ses paramètres, le modèle est prêt à subir les tests nécessaires pour révéler ses aptitudes et/ou ses lacunes à représenter la dynamique du système ou du processus réel. En générale, les données de validation présentent des similarités statistiques avec les données utilisées auparavant dans l'identification du modèle, mais différent dans leurs formes et variations temporelles. La génération des données de validation suit les mêmes règles que celles des données d'identification.

La méthode la plus naturelle pour la validation (aussi la plus utilisée dans la pratique de l'identification) est de comparer la réponse simulée du modèle et la réponse du système réel pour les mêmes signaux d'entrée (Fig.5.3).

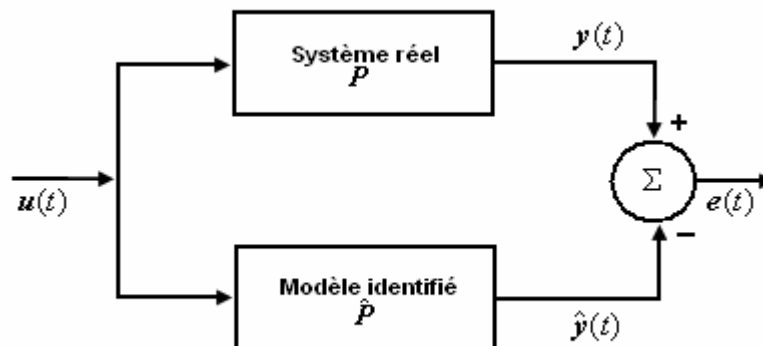


Figure 5.3 : Schéma de validation d'un modèle

Toutefois, des techniques de validation plus avancées sont proposées et sont, surtout, basées sur des analyses statistiques de la suite des résidus $\{e(k, \theta)\}$ obtenus à partir des données de validation. Ces techniques exploitent des paramètres tels que le maximum des valeurs absolues des résidus, leur valeur moyenne, leur variance, la covariance et la corrélation entre les résidus et les valeurs passées des signaux d'entrée [110].

5.2 Structures des modèles empiriques

Nous assumons dans ce qui suit que la sortie $\hat{y}(\cdot)$ du modèle que représente l'opérateur $\hat{P} \in \hat{\mathcal{P}}$ est écrite

$$\begin{aligned}\hat{y}(t, \theta) &= g(\mathbf{y}^{t-1}, \mathbf{u}^t, \theta) \\ &= g(\varphi(t), \theta)\end{aligned}\tag{5.6}$$

Où $\theta \in \mathbb{R}^d$ représente le vecteur des paramètres, \mathbf{y}^{t-1} dénote l'ensemble de toutes les valeurs du vecteur de sortie jusqu'à l'instant $t-1$, \mathbf{u}^t les signaux d'entrée jusqu'à l'instant t . $\varphi(t) = \varphi(\mathbf{y}^{t-1}, \mathbf{u}^t)$ est connu sous l'appellation de vecteur de régression. Dans certaines situations $\varphi(t)$ dépend des paramètres du modèle et on écrit $\varphi(t) = \varphi(\mathbf{y}^{t-1}, \mathbf{u}^t, \theta)$.

5.2.1 Modèles linéaires

On assume, généralement, que l'opérateur \mathbf{P} caractérisant un système linéaire mono-entrée mono-sortie (SISO) découle de la relation entrée-sortie généralisée suivante (Fig. 5.3) [105] :

$$y(t) = G(q)u(t) + H(q)e(t)\tag{5.7}$$

Où $G(q)$ et $H(q)$ des fonctions de transfert et q l'opérateur de retard temporel ($(q^{-d}u(t) = u(t-d))$). La suite $\{e(t)\}$, $t=1,2,\dots,N$ est une perturbation qui prend la forme d'un bruit blanc.

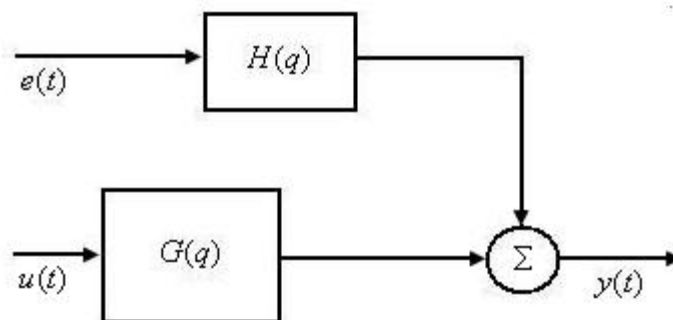


Figure 5.3 : Modèle linéaire entrée/sortie

En remplaçant $e(t)$ par $y(t) - \hat{y}(t, \theta)$ dans (5.7), on obtient

$$\hat{y}(t, \theta) = H(q)^{-1} G(q) u(t) + (1 - H(q)^{-1}) y(t) \quad (5.8)$$

Ou, encore, en introduisant le vecteur de régression

$$g(t, \theta, \varphi(t)) = H(q)^{-1} G(q) u(t) + (1 - H(q)^{-1}) y(t) \quad (5.9)$$

En optant pour une paramétrisation générale de la structure du modèle (5.7)

$$y(t) = q^{-n_k} \frac{B(q)}{A(q)F(q)} u(t) + \frac{C(q)}{A(q)D(q)} e(t) \quad (5.10)$$

L'équation (5.9) est réécrite

$$g(.) = q^{-n_k} \frac{B(q)D(q)}{C(q)F(q)} u(t) + \left(1 - \frac{A(q)D(q)}{C(q)} \right) y(t) \quad (5.11)$$

Avec n_k un entier dénotant le retard temporel entre les signaux d'entrée et de sortie et les polynômes $A(q)$, $B(q)$, $C(q)$, $D(q)$ et $F(q)$ donnés par

$$A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a} \quad (5.12)$$

$$B(q) = b_1 + b_2 q^{-1} + \dots + b_{n_b} q^{-n_b} \quad (5.13)$$

$$C(q) = 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c} \quad (5.14)$$

$$D(q) = 1 + d_1 q^{-1} + \dots + d_{n_d} q^{-n_d} \quad (5.15)$$

$$F(q) = 1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f} \quad (5.16)$$

Tel que le vecteur des paramètres devient $\theta = [a_1, \dots, a_{n_a}, \dots, f_1, \dots, f_{n_f}]^T$.

Plusieurs cas de structures de modèles linéaires se présentent selon les degrés des polynômes $A(q)$, $B(q)$, $C(q)$, $D(q)$ et $F(q)$,

(i) *Modèle FIR (Finite Impulse Response)*

$$A(q) = C(q) = D(q) = F(q) = 1 \Rightarrow g(.) = q^{-n_k} B(q) u(t) \Rightarrow g(.) = \theta^T \varphi(t) \quad (5.17)$$

(ii) *Modèle ARX (AutoRegressive with eXogenous input)*

$$C(q) = D(q) = F(q) = 1 \Rightarrow g(.) = q^{-n_k} B(q) u(t) + (1 - A(q)) y(t) \Rightarrow g(.) = \theta^T \varphi(t) \quad (5.18)$$

(iii) *Modèle OE (Output-Error)*

$$A(q) = C(q) = D(q) = 1 \Rightarrow g(.) = q^{-n_k} \frac{B(q)}{F(q)} u(t) \Rightarrow g(.) = \theta^T \varphi(t, \theta) \quad (5.19)$$

(iv) *Modèle ARMAX (AutoRegressive Moving Average with eXogenous input)*

$$D(q) = F(q) = 1 \Rightarrow g(\cdot) = q^{-n_k} \frac{B(q)}{C(q)} u(t) + \left(1 - \frac{A(q)}{C(q)}\right) y(t) \Rightarrow g(\cdot) = \theta^T \varphi(t, \theta) \quad (5.20)$$

Notons que dans les cas *OE* et *ARMAX*, le vecteur de régression $\varphi(t, \theta)$ est lui-même paramétrisé.

Dans le cas des systèmes multivariables, il est souvent plus commode de travailler avec une structure de modèle en espace d'état prenant la forme

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{F}(\theta)\mathbf{x}(t) + \mathbf{G}(\theta)\mathbf{u}(t) + \mathbf{K}(\theta)\mathbf{e}(t) \\ \mathbf{y}(t) &= \mathbf{H}(\theta)\mathbf{x}(t) + \mathbf{D}(\theta)\mathbf{u}(t) + \mathbf{e}(t) \end{aligned} \quad (5.21)$$

Le prédicteur correspondant est, alors, écrit

$$\begin{aligned} \hat{\mathbf{x}}(t+1) &= (\mathbf{F}(\theta) - \mathbf{K}(\theta)\mathbf{H}(\theta))\hat{\mathbf{x}}(t) + (\mathbf{G}(\theta) - \mathbf{K}(\theta)\mathbf{D}(\theta))\mathbf{u}(t) + \mathbf{K}(\theta)\mathbf{y}(t) \\ \hat{\mathbf{y}}(t) &= \mathbf{H}(\theta)\hat{\mathbf{x}}(t) + \mathbf{D}(\theta)\mathbf{u}(t) \end{aligned} \quad (5.22)$$

Dans cette représentation $\mathbf{x}(t) \in \mathfrak{R}^p$ est le vecteur d'état, $\mathbf{u}(t) \in \mathfrak{R}^m$ est le vecteur d'entrée, $\mathbf{y}(t) \in \mathfrak{R}^n$ est le vecteur de sortie et $\mathbf{e}(t) \in \mathfrak{R}^n$, le vecteur de bruit. Les matrices $\mathbf{F}(\theta), \mathbf{G}(\theta), \mathbf{K}(\theta), \mathbf{H}(\theta)$ et $\mathbf{D}(\theta)$ sont de dimensions appropriées.

Une méthode de passage d'un modèle *ARMAX* multivariables vers un modèle d'état avec estimation des paramètres par un algorithme basé sur les moindres carrés est développée dans [23].

5.2.2 Modèles nonlinéaires

La classe des structures de modèles empiriques non linéaires est trop vaste [27, 103 et 111] et nous nous limiterons à présenter les prédicteurs non linéaires *NARX* et *NARMAX*. Nous supposons dans ce qui suit que le système non linéaire multivariables (*MIMO*) est décrit par la relation

$$\begin{aligned} \mathbf{y}(t) &= \mathbf{f}(t, \theta, \mathbf{y}^{t-1}, \mathbf{u}^t) + \mathbf{e}(t) \\ &= \mathbf{f}(\varphi(t), \theta) + \mathbf{e}(t) \end{aligned} \quad (5.23)$$

Où $\mathbf{f}(\cdot)$ est une fonction non linéaire inconnue et $\mathbf{e}(t)$ un vecteur de perturbation additif sous forme d'un bruit blanc.

En remplaçant $\mathbf{e}(t)$ par $\mathbf{y}(t) - \hat{\mathbf{y}}(t, \theta)$ dans l'équation précédente, on obtient la fonction du prédicteur

$$\hat{\mathbf{y}}(t, \theta) = \mathbf{f}(\varphi(t), \theta) \quad (5.24)$$

Selon le choix du vecteur de régression $\varphi(t)$, plusieurs structures de modèles non linéaires émergent.

(i) *Modèle NARX (Nonlinear AutoRegressive with eXogenous input)*

Dans cette structure le vecteur de régression $\varphi(t)$ contient les valeurs passées des entrées et sorties :

$$\varphi(t) = [y(t-1), \dots, y(t-n_y), u(t-d), \dots, u(t-d-n_u+1)]^T \quad (5.25)$$

Le prédicteur s'écrit alors,

$$\hat{y}(t, \theta) = f(y(t-1), \dots, y(t-n_y), u(t-d), \dots, u(t-d-n_u+1), \theta) \quad (5.26)$$

Cette structure est illustrée par la figure 5.4

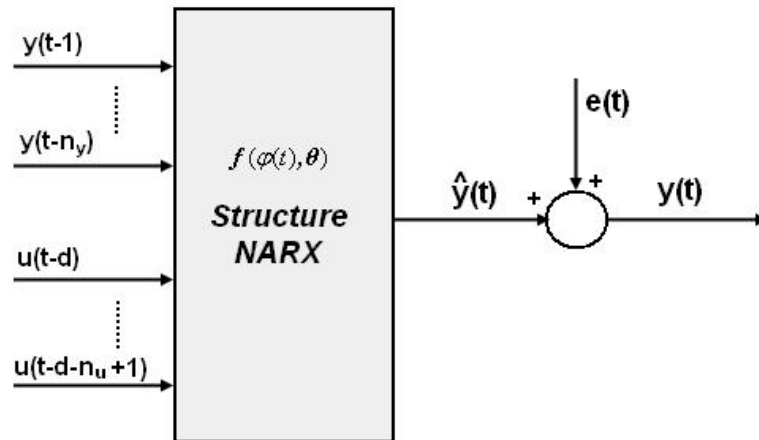


figure 5.4 : Modèle NARX

Les valeurs du retard d et des ordres du modèle NARX n_y et n_u sont sélectionnées pendant la phase de l'identification. L'avantage de cette structure réside dans sa simplicité et sa convergence rapide. Elle est souvent utilisée, bien que sa représentation de l'effet de la perturbation ou du bruit ne soit pas toujours adéquate. A partir de (5.26), la sortie prédite sur un horizon $k \geq 1$ à l'instant présent t s'écrit

$$\hat{y}(t+k | t) = f(y(t+k-1), \dots, y(t+k-n_y), u(t+k-d), \dots, u(t+k-d-n_u+1), \theta) \quad (5.27)$$

A défaut des mesures futures de la sortie, les prédictions $\hat{y}(t+1|t), \dots, \hat{y}(t+k-1|t)$ sont calculées itérativement à l'aide de (5.26) et utilisés en substitution dans (5.27). Une autre alternative consiste à identifier un prédicteur d'horizon k en utilisant un vecteur de régression $\varphi(t+k)$,

$$\hat{y}(t+k|t) = f(\varphi(t+k), \theta) \quad (5.28)$$

$$\varphi(t+k) = [y(t+k-1), \dots, y(t+k-n_y), u(t+k-d), \dots, u(t+k-d-n_u+1)]$$

Dans ce cas, le prédicteur (5.28) peut être utilisé de manière efficace dans une stratégie de commande prédictive, bien qu'il n'est pas considéré comme un modèle du système.

- (ii) *Modèle NARMAX (Nonlinear AutoRegressive Moving Average with eXogenous input)*

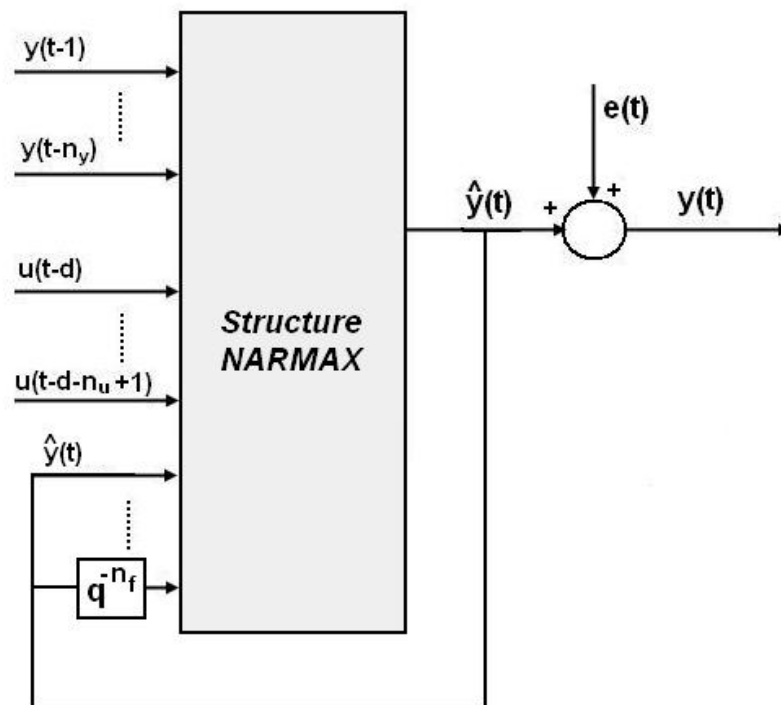


figure 5.5 : Structure NARMAX

Dans cette structure le vecteur de régression $\varphi(t)$ contient non seulement les valeurs passées des entrées et sorties mesurées, mais aussi les erreurs de prédiction:

$$\varphi(t, \theta) = [y(t-1), \dots, y(t-n_y), u(t-d), \dots, u(t-d-n_u+1), e(t, \theta), \dots, e(t-n_e, \theta)]^T \quad (5.29)$$

Le prédicteur s'écrit,

$$\hat{y}(t, \theta) = f(y(t-1), \dots, y(t-n_y), u(t-d), \dots, u(t-d-n_u+1), e(t, \theta), \dots, e(t-n_e, \theta)) \quad (5.30)$$

L'entier n_e représente le nombre des erreurs prédites passées.

Une représentation analogue peut être formulée en substituant les erreurs de prédiction passées par les sorties prédites. le vecteur de régression $\varphi(t, \theta)$ s'écrit :

$$\varphi(t, \theta) = [y(t-1), \dots, y(t-n_y), u(t-d), \dots, u(t-d-n_u+1), \hat{y}(t, \theta), \dots, \hat{y}(t-n_f, \theta)]^T \quad (5.31)$$

Avec n_f le nombre des prédictions de sortie passées. Cette structure est illustrée par la figure 5.5

La technique la plus souvent appliquée pour la réalisation des structures de modèles non linéaires est la modélisation par les réseaux de neurones. Comme nous l'avons déjà souligné dans le chapitre IV, les perceptrons multicouches et les réseaux de neurones RBF possèdent des aptitudes importantes à approximer les fonctions non linéaires. Dans la section suivante nous présentons les détails théoriques ainsi que les simulations relatives à notre contribution à l'identification des systèmes non linéaires par les réseaux RBF.

5.3 Approche pratique à l'identification non linéaire par les réseaux de neurones RBF

Plusieurs stratégies de commande, en particulier la commande prédictive ou commande à horizon fuyant, nécessitent la disponibilité d'un modèle fiable, fidèle et précis décrivant le système à commander et reflétant ses non-linéarités et ses complexités dynamiques. Parmi les choix possibles pour la réalisation d'une structure de modèle non linéaire, les réseaux de neurones occupent une position privilégiée à cause de leurs aptitudes prouvées, théoriquement et pratiquement, à l'approximation des fonctionnelles non linéaires avec une précision arbitrairement choisie *à priori*. Cette puissance de l'outil neuronique dans l'identification non linéaire fut, d'abord, mise en évidence grâce au travail pionnier de Narendra et al [27], sur la modélisation non linéaire par les réseaux de

neurones et leurs applications dans la commande adaptative à modèle de référence MRAC (Model Reference Adaptive Control). Depuis, plusieurs techniques appliquant les réseaux de neurones aux stratégies de commande non linéaires furent avancées. Dans le domaine particulier de la commande prédictive la liste des contributions est trop longue pour être revue en entier et, nous citerons ici quelques travaux tels ceux de Hunt et al. [28] et Willis et al.[112], qui proposèrent une architecture de commande prédictive où des modèles à réseaux de neurones fournissent les prédictions futures des sorties du processus à l'algorithme d'optimisation qui calcul la séquence des signaux de commande sur un horizon de prédiction. La contribution de Tan et al.[113] décrit un schéma de commande adaptative prédictive à modèle neuronique avec une mise à jour en temps réel des paramètres du perceptron, par un algorithme RLS, pour accommoder et s'adapter aux variations continues du système. D'autres contributions, les unes aussi importantes que les autres furent avancées, mais nous citerons le travail de Neumerkel et al.[114] pour son application en temps réel d'une commande prédictive à modèle RBF appliquée à une machine asynchrone et celui de Sbarbaro et al.[115] sur l'applicabilité des réseaux RBF dans différents niveaux d'un système industriel automatisé. L'option RBF semble acquérir de plus en plus la confiance des concepteurs des produits de technologie prédictive tel Predictive Control Ltd. (PCL) avec son produit 'Connoisseur' [116] capable d'intégrer des modèles RBF pour prendre en charge les processus industriels fortement non linéaires.

Dans ce travail nous adoptons la technique d'identification par les réseaux RBF à fonction d'activation Gaussienne (GRBF) d'abord, pour leurs aptitudes à l'approximation et la simplicité de leur structure topologique et en suite, à cause de leur popularité parmi la communauté des scientifiques et chercheurs dans plusieurs disciplines. Le problème principal dans l'identification par les réseaux RBF ou GRBF est de déterminer le nombre des centres ainsi que leur distributions dans l'espace d'entrée avant de procéder à l'estimation des autres paramètres du réseaux. Dans la suite de cette section nous introduisons sommairement, d'abord, les techniques d'apprentissage les plus utilisées pour les réseaux GRBF, avant de décrire avec détails la technique adoptée dans ce travail et qui est basée sur l'algorithme *Resource-Allocating Network* (RAN) de J. Platt [117], amélioré plus

tard par V. Kadiramanathan et al. [118] qui pourvurent le RAN d'un estimateur des paramètres utilisant le Filtrage Etendu de Kalman et produisirent, ainsi, un nouvel algorithme appelé RANEKF. Nous introduisons aussi une nouvelle structure de réseaux GRBF bouclé capable d'émuler un modèle non linéaire dans l'espace d'état. Cette nouvelle structure possède des propriétés analytiques intéressantes pour implémenter des stratégies de commande prédictive non linéaires stabilisantes en régulation ou en poursuite (tracking). Nous finirons ce chapitre en présentant le concept de l'identification non linéaire inverse.

5.3.1 Stratégies d'apprentissage des réseaux GRBF

Dans le chapitre 4, section 4.4.3, nous avons appliqué les résultats théoriques de Sandberg [102] pour prouver que les réseaux GRBF jouissent de la propriété d'approximateurs universels (voir corollaire 4.1). Cette propriété fut aussi mise en évidence, entre autres, par Girosi et Poggio [97] et Hartman et al. [98]. Cependant, ce résultat est uniquement existentiel et, le choix de la structure optimale d'un réseau RBF/GRBF pour identifier une application non linéaire spécifique est un problème abordé de diverses manières dans la littérature. Dans le réseau RBF original appliqué à l'interpolation d'une fonction multivariées, la couche cachée contenait autant de centres que d'échantillons de données disponibles. Cette structure n'est pas pratique vu le grand nombre des échantillons utilisés. A partir de cette situation plusieurs méthodes ont émergés pour la sélection des centres et l'apprentissage des réseaux RBF/GRBF par estimation des autres paramètres.

L'approche la plus simple était basée sur une sélection aléatoire des centres du réseau parmi les échantillons disponibles. Cette méthode peut-être acceptable dans la mesure où les données sont distribuées de manière représentative à travers tout l'espace d'entrée [95]. Les largeurs des gaussiennes sont fixées par la formule : $\sigma = \frac{d}{\sqrt{M}}$. Avec M le nombre total des centres sélectionnés et où d représente la valeur maximale des distances euclidiennes entre les centres choisis. Les poids du réseau sont déterminés par application directe de la méthode de *pseudo inverse*.

Dans une deuxième technique développée par Moody et Darken [96], les centres sont sélectionnés *a priori* à l'aide de l'algorithme de segmentation en k-moyennes (k-means clustering algorithm). Les données sont partitionnées dans l'espace d'entrée en groupements locaux ou 'nuages' à haute concentration de points et, les centres sont naturellement positionnés au juste milieu de chacun des 'nuages' formés. Les largeurs de chaque gaussiennes sont déterminées par la moyenne des distances qui séparent le centre du 'nuage' avec ses plus proches voisins –ppv– dont le nombre est choisi de manière *ad hoc*. L'apprentissage des poids du réseau GRBF se fait, encore une fois, à l'aide de la procédure de *pseudo inverse*.

Notons, au passage, que le problème de la segmentation des données se pose dans d'autres disciplines (classification et reconnaissance des formes) et est traité par plusieurs techniques autres que la k-moyennes, tel l'algorithme EM (Expectation Maximisation) [119, 120], et l'algorithme de segmentation par poursuite de la moyenne [121].

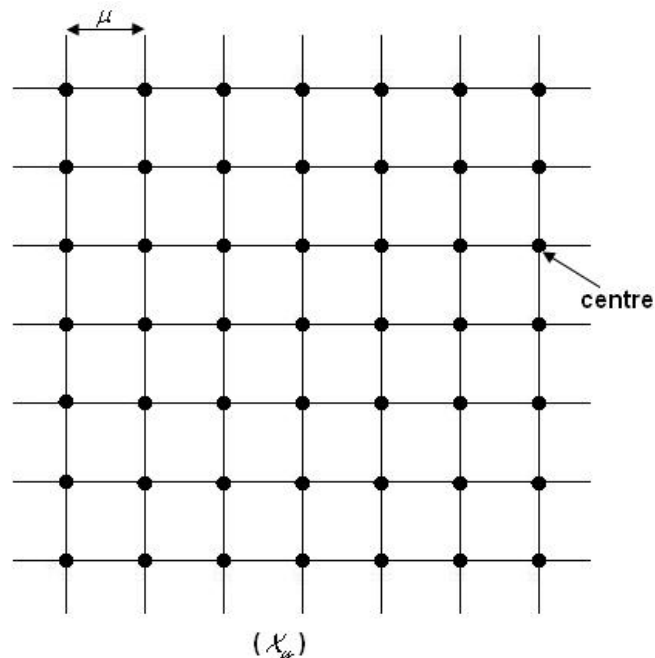


figure 5.6 : Maille bidimensionnelle d'un réseau RBF

Dans une troisième approche les centres des gaussienne sont, typiquement, placés sur des points régulièrement espacés d'une maille carrée ou multi-

dimensionnelle (Fig 5.6) couvrant une région compacte X_u de l'espace d'entrée. La distance μ entre les points de la maille détermine le nombre des centres couvrant la région X_u et la précision d'approximation du réseau dépend de la valeur μ . Une procédure systématique pour la détermination des largeurs et des poids du réseau est présentée dans le travail de Sanner et al. [122].

5.3.2 Algorithme RANEKF modifié à apprentissage aléatoire

Les méthodes d'apprentissage citées plus haut sont toutes de type *off line* car, la construction du réseau RBF/GRBF et/ou l'estimation de ses paramètres nécessitent la disponibilité, au préalable, d'un paquet de données d'apprentissage. L'algorithme *Resource-Allocating Network* (RAN) proposé par J. Platt [117], repose sur une philosophie différente, celle d'une construction *on line* du réseau et de son élargissement au fur et à mesure que les données sont mesurées et mises à disposition de l'algorithme. L'apprentissage se déroule de manière séquentielle, l'algorithme est initialisé avec un *réseau vide* auquel sont alloués des centres, de manière graduelle, en se basant sur des critères de '*nouveauté*' du vecteur de donnée alimentant l'entrée. Les paramètres du réseau, y compris les positions des centres déjà alloués, sont ajustés de façon adaptative à l'aide de l'algorithme des moindres carrés (MC). Un vecteur d'entrée est considéré comme '*nouveau*', et est ajouté à la structure dynamique du réseau, s'il est suffisamment éloigné du reste des centres (au sens d'une distance euclidienne) et, si l'écart entre la sortie du réseau courant et la valeur cible est importante. Si les critères de '*nouveauté*' ne sont pas vérifiés, la structure du réseau est gardée en l'état et les paramètres sont réajustés. Kadiramanathan et al. [118] interpréta le RAN du point de vue de l'approximation des fonctionnelles par apprentissage séquentiel ou adaptatif et, améliora ses performances en introduisant le filtre étendu de Kalman pour l'estimation et l'adaptation des paramètres. Le nouvel algorithme, appelé RANEKF, possède de meilleures propriétés de convergence par rapport au RAN original. Cette philosophie de réseau RBF à structure adaptative était appliquée dans l'identification *on line* des systèmes non linéaires variables par L. Yingwei et al. [123] ainsi que dans une stratégie de commande adaptative prédictive basée sur des prédicteurs RBF à structures variables avec mise à jour *on line* des

paramètres par G. P. Liu et al. [124]. De leur part, Ahmida and Charef [49] ramènent l'algorithme RANEKF, traditionnellement utilisé en apprentissage séquentiel ou *on line*, vers des applications *off-line* où les données d'identification sont collectées à l'avance et, apportent deux modifications au fonctionnement de l'algorithme :

- (1) Les données collectées subissent une opération de permutation aléatoire avant qu'elles ne soient présentées à l'algorithme.
- (2) Les critères de '*nouveauté*' sont modifiés en introduisant l'écart relatif entre la sortie du réseau courant et la valeur cible, à la place de l'écart absolu. Cette nouvelle condition est introduite dans le but de garantir que l'erreur serait toujours une petite fraction de la valeur cible, et ceci, sur tout l'espace de sortie.

L'algorithme *RANEKF modifié* est appliqué à l'identification d'une structure NARX mono-entrée mono-sortie (SISO) telle que définie par les équations (5.25) et (5.26) et illustrée par la figure (5.4).

Lorsqu'un réseau à fonctions gaussiennes à base radiale (GRBF) est choisie pour représenter la structure du modèle NARX, la sortie prédite est donnée par :

$$\hat{y}(t) = f(\varphi(t), \theta) = w_0 + \sum_{k=1}^K w_k \Phi_k(\varphi(t), \mathbf{c}_k, \sigma_k) \quad , \quad t = 0, 1, \dots \quad (5.32)$$

Où, les fonctions $\Phi_k(\cdot)$ sont des gaussiennes définies par

$$\Phi_k(\varphi(t), \mathbf{c}_k, \sigma_k) = \exp\left(-\frac{\|\varphi(t) - \mathbf{c}_k\|^2}{\sigma_k^2}\right) \quad (5.33)$$

les $\mathbf{c}_k \in \mathcal{R}^{n_y + n_u}$ sont les centres du réseau GRBF , les σ_k sont les largeurs des gaussiennes et $\|\cdot\|$ dénote la norme euclidienne . Le nombre des centres est donné par l'entier K et les poids synaptiques sont les w_k , avec w_0 un biais ou seuil. Le vecteur de régression $\varphi(t)$ représente l'entrée du réseau et, le vecteur des paramètres θ est donné par

$$\theta = [w_0, w_1, \dots, w_K, \sigma_1, \sigma_2, \dots, \sigma_K]^T \quad (5.34)$$

Pour spécifier complètement la structure NARX, le problème d'identification implique la détermination des paramètres K, \mathbf{c}_k , w_k , w_0 et σ_k du réseau GRBF à

partir d'un ensemble de N observations $\{\varphi(t), y(t)\}$, $1 \leq t \leq N$; et tel que l'erreur $e(t) = y(t) - \hat{y}(t)$ soit minimisée.

Avant de démarrer l'exécution de l'algorithme *RANEKF modifié*, on procède d'abord à une permutation aléatoire des N colonnes de la matrice formée par les observations $\{\varphi(\cdot), y(\cdot)\}$. A la i ème itération, l'algorithme reçoit le vecteur $\varphi(i)$ à l'entrée et $y(i)$ comme valeur cible correspondante. L'algorithme est initialisé en affectant la première valeur cible présentée $y(1)$ au biais et en fixant une valeur aléatoire $\varphi(\cdot)$ parmi les données d'apprentissage comme premier centre du réseau GRBF. En recevant les vecteurs d'observation $\{\varphi(\cdot), y(\cdot)\}$, de nouveaux centres sont alloués au réseau sur la base de certains critères de 'nouveaueté'. A la i ème itération, les deux conditions suivantes doivent être vérifiées pour que le vecteur $\varphi(i)$ soit accepté comme un nouveau centre :

$$\|\varphi(i) - \mathbf{c}_{nr}\| > \varepsilon(i) \quad (5.35)$$

$$e_r(i) = \left| \frac{y(i) - f(\varphi(i), \theta)}{y(i)} \right| > \eta \quad (5.36)$$

Avec \mathbf{c}_{nr} le centre le plus proche au vecteur $\varphi(i)$ dans l'espace d'entrée, $e_r(\cdot)$ l'erreur relative sur la réponse du réseau, $\varepsilon(i)$ et η sont des seuils positifs appropriés. En particulier, $\varepsilon(i)$ est réduite graduellement à partir d'une valeur maximale ε_{max} jusqu'à une valeur minimale permise ε_{min} , selon la formule :

$$\varepsilon(i) = \max\{\varepsilon_{max}\gamma^i, \varepsilon_{min}\}, \quad 0 < \gamma < 1 \quad (5.37)$$

Dés qu'un nouveau centre (unité gaussienne) est sélectionné, les paramètres qui lui sont associés sont affectés des valeurs suivantes :

$$\begin{aligned} w_{K+1} &= e_a(i) \\ &= y(i) - f(\varphi(i), \theta) \end{aligned} \quad (5.38)$$

$$c_{K+1} = \varphi(i) \quad (5.39)$$

$$\sigma_{K+1} = \rho \|\varphi(i) - \mathbf{c}_{nr}\| \quad (5.40)$$

Avec $\rho > 0$ appelé facteur de chevauchement.

Lorsqu'une observation $\{\varphi(\cdot), y(\cdot)\}$ ne satisfait pas à l'un des critères de nouveauté, un filtre étendu de Kalman (Extended Kalman Filter-EKF) est appliqué

pour réajuster les poids et les largeurs du réseau contenus dans le vecteur des paramètres $\theta = [w_0, w_1, \dots, w_K, \sigma_1, \sigma_2, \dots, \sigma_K]^T$ comme suit :

$$\theta(i) = \theta(i-1) + g(i)e_a(i) \quad (5.41)$$

$g(i)$ est le vecteur gain de Kalman calculé par la formule récurrente suivante :

$$g(i) = \mathbf{S}(i-1)\mathbf{a}(i)[\mathbf{R}(i) + \mathbf{a}^T(i)\mathbf{S}(i-1)\mathbf{a}(i)]^{-1} \quad (5.42)$$

où $\mathbf{a}(i)$ représente le vecteur gradient de la fonction $f(\varphi(t), \theta)$ calculé par rapport au vecteur des paramètres θ :

$$\mathbf{a}(i) = \nabla_{\theta} f(\varphi(i), \theta(i)) \quad (5.43)$$

$\mathbf{R}(i)$ est la variance du bruit de mesure et, $\mathbf{S}(i)$ la matrice covariance de l'erreur et qui est mise à jour selon la formule :

$$\mathbf{S}(i) = [\mathbf{I} - g(i)\mathbf{a}^T(i)]\mathbf{S}(i-1) + Q_0\mathbf{I} \quad (5.44)$$

La matrice de covariance est une matrice carrée positive et symétrique de dimensions $s \times s$, avec s le nombre total des paramètres estimés ($s=2K+1$) et Q_0 un réel positif introduit pour limiter la valeur permise du pas en direction du vecteur gradient [..].

Lorsqu'un nouveau centre est alloué le nombre K est incrémenté de l'unité ($K \leftarrow K + 1$) et la matrice $\mathbf{S}(i)$ est modifiée par :

$$\mathbf{S}(i) = \begin{pmatrix} \mathbf{S}(i-1) & 0 \\ 0 & \mathbf{S}_0\mathbf{I} \end{pmatrix} \quad (5.45)$$

L'algorithme d'apprentissage RANEKF modifié se résume comme suit :

Etape 0 0.1 initialiser les paramètres de l'algorithme

$$i = 1, \eta, \varepsilon_{max}, \varepsilon_{min}, \gamma, \rho, N, \mathbf{R}_0, \mathbf{S}_0, Q_0$$

0.2 obtenir la première observation $\{\varphi(1), y(1)\}$ et poser $w_0 = y(1)$

0.3 sélectionner un nombre aléatoire $k_0 ; 1 \leq k_0 \leq N$ et poser

$$c_1 = \varphi(k_0)$$

0.4 initialiser $K=1$

Etape 1 1.1 présenter une nouvelle observation $\{\varphi(i), y(i)\}$

1.2 calculer $\varepsilon(i)$ selon (5.37)

Etape 2 2.1 If (5.35) et (5.36) do

Allouer nouveau centre et appliquer (5.38), (5.39) et (5.40)

Mettre à jour $S(i)$ selon (5.45)

Incrémenter $K=K+1$

Else do

Calculer le vecteur $a(i)$ selon (5.43)

Calculer le vecteur gain de Kalman $g(i)$ selon (5.42)

réajuster le vecteur paramètres $\theta(i)$ selon (5.41)

Mettre à jour $S(i)$ selon (5.44)

Etape 3 3.1 incrémenter $i=i+1$

If $i \leq N$ do

goto Etape 1

Else do

Stop algorithme

Les applications d'identification de structures NARX par l'algorithme RANEKF modifié portent sur deux exemples types de systèmes non linéaires. Le premier est un bras manipulateur à articulation unique [125-127] et le deuxième est un système non linéaire discret mono-entrée mono-sortie (SISO) [27,128].

Premier exemple de simulation

Le système non linéaire considéré dans cette première simulation est un simple bras manipulateur (single link manipulator) décrit par l'équation différentielle non linéaire de deuxième ordre :

$$ml^2 \ddot{\theta}(t) + v\dot{\theta}(t) + mgl \sin \theta(t) = u(t) \quad (5.46)$$

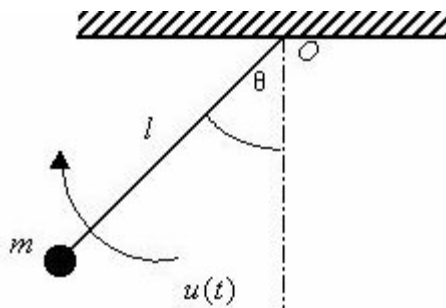


Figure 5.7 : Bras manipulateur

Où la masse $m=0.25\text{Kg}$, la longueur du bras est $l=1.0\text{m}$, le coefficient de friction $\nu = 1.0 \text{ kg m}^2 / \text{s}$, l'accélération due à la gravité $g = 9.8 \text{ m/s}^2$, $\theta(t)$ l'angle que fait le bras avec la verticale mesuré en radians, et $u(t)$ représente le couple appliqué en Nm.

Le manipulateur peut effectuer des mouvements de rotation autour de l'axe passant par O (Fig.5.7) par application d'un couple u qui control l'amplitude de l'angle $\theta(t)$. Dans cette simulation le domaine d'opération est limité dans les intervalles $[-0.65 \quad 0.65]$ et $[-1.5 \quad 1.5]$ pour l'angle $\theta(t)$ et le couple $u(t)$, respectivement.

Les données d'identification sont générées par application de signaux sinusoïdaux $u(t)$ avec des fréquences (Hz) dans l'intervalle $[0.01 \quad 0.05]$. Une matrice de données contenant un total de 2500 colonnes est, ensuite, formée. Chaque colonne de la matrice représente une observation $\{\varphi(t), \theta(t)\}$, avec

$$\varphi(t) = [\theta(t-1), \theta(t-2), \theta(t-3), u(t-1), u(t-2)]^T \quad (5.47)$$

Les paramètres de l'algorithme RANEKF modifié sont fixés comme suit :

$$\varepsilon_{\max} = 1, \quad \varepsilon_{\min} = 0.65, \quad \gamma = 0.98, \quad \eta = 0.01, \quad \rho = 0.95, \quad \mathbf{Q}_0 = 0.001 \quad \text{et} \quad \mathbf{R}_0 = \mathbf{S}_0 = 1.$$

Une portion des données d'identification après permutation aléatoire est illustrée dans la figure 5.8.

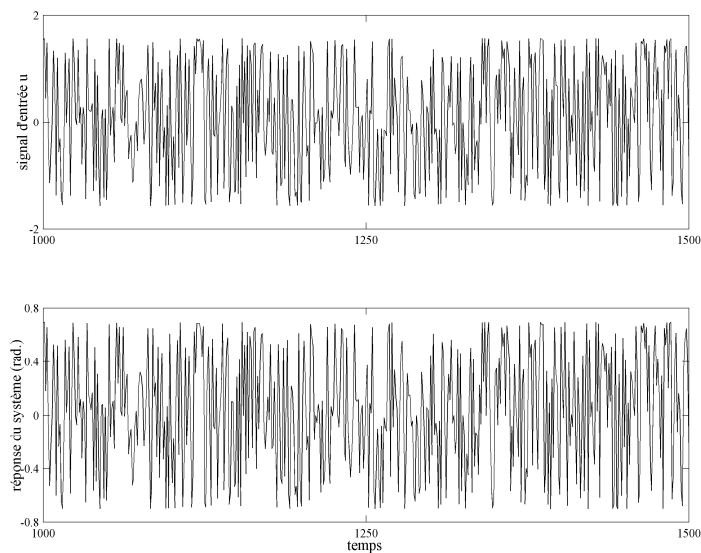


Figure 5.8 : Disposition aléatoire des données de l'identification

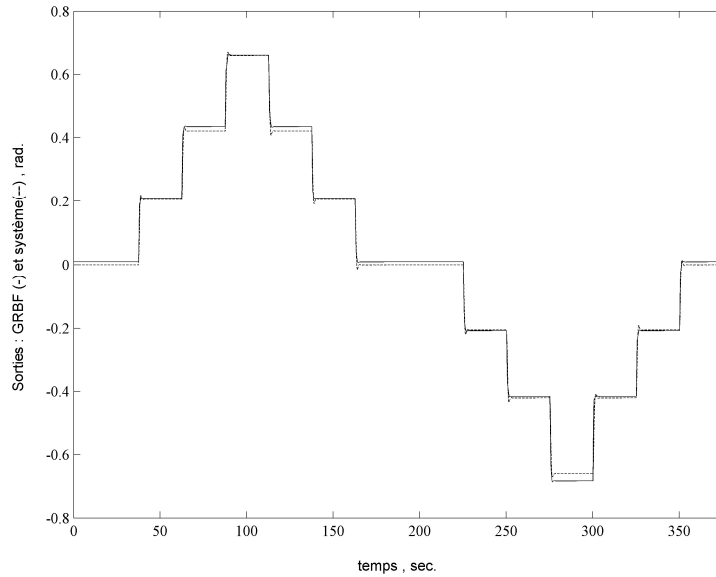


Figure 5.9 : Réponse du modèle GRBF et du système à une excitation en escalier

La structure NARX du modèle GRBF obtenu, contenant onze centres, est testée avec deux types de signaux d'entrée; un signal en escalier (Fig.5.9) variant dans l'intervalle [-1.5 1.5] et un signal somme de sinusoïdes (Fig. 5.10) donné par :

$$u(t) = \frac{\pi}{10} \sin\left(\frac{2\pi}{100}t\right) + \frac{\pi}{10} \sin\left(\frac{2\pi}{50}t\right) + \frac{\pi}{10} \sin\left(\frac{2\pi}{20}t\right) \quad (5.48)$$

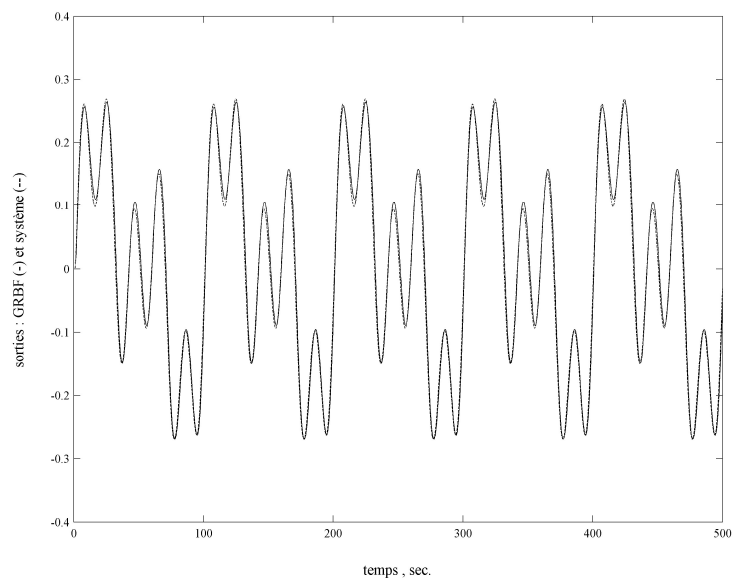


Figure 5.10 : Réponse du modèle GRBF et du système à une excitation périodique

Deuxième exemple de simulation

Le système SISO non linéaire du deuxième exemple est décrit par l'équation discrète suivante

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) \quad (5.49)$$

L'objectif de la simulation est de construire un modèle GRBF capable d'approximer le système non linéaire décrit par (5.49) dans un domaine d'opération défini par l'intervalle $[-1 \ 1]$ pour le signal d'entrée et qui correspond à des variations de la sortie dans l'intervalle $[-1.5 \ 1.5]$.

Les données utilisées pour l'identification du modèle sont obtenues par l'application d'un signal d'entrée synthétique constitué d'une partie aléatoire variant dans l'intervalle $[-1 \ 1]$ à laquelle sont ajoutés des signaux carrés et sinusoïdaux. La totalité du signal obtenu est superposée à un bruit blanc de moyenne nulle et de variance 0.1. Dans cet exemple la matrice des observations $\{\varphi(t), y(t+1)\}$ contient 14800 colonnes, avec un vecteur de régression donné par :

$$\varphi(t) = [u(t), u(t-1), y(t)]^T \quad (5.50)$$

avec $y(t+1)$ la sortie du prédicteur d'ordre un.

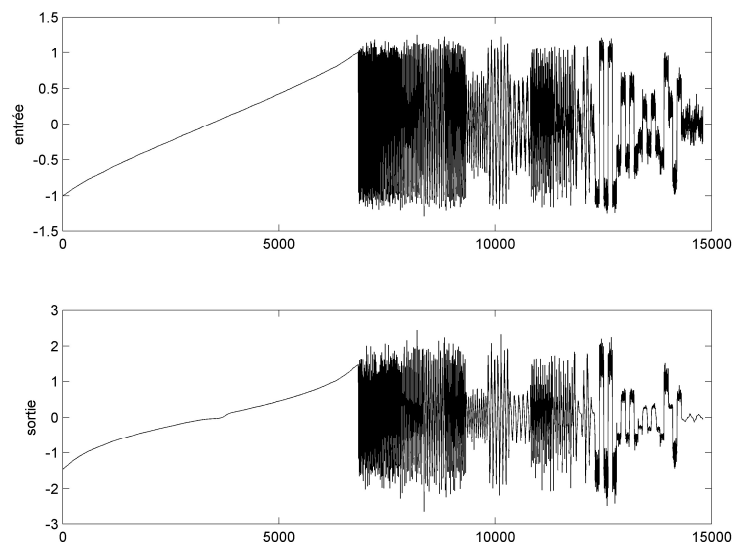


Figure 5.10 : Données d'identification avant la permutation aléatoire

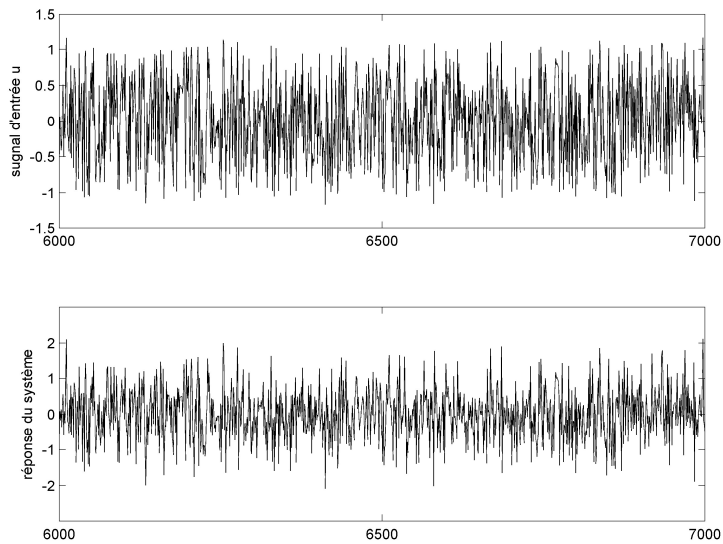


Figure 5.11 : Données d'identifications après permutation aléatoire

Les signaux d'entrée et de sortie utilisés dans cet exemple d'identification, avant et après permutation aléatoire, sont illustrés dans les (5.10) et (5.11). Les paramètres de l'algorithme RANEKF modifié sont fixés comme suit :

$$\varepsilon_{\max} = 0.80, \quad \varepsilon_{\min} = 0.65, \quad \gamma = 0.999, \quad \eta = 0.01, \quad \rho = 0.85, \quad \mathbf{Q}_0 = 0.0045 \quad \text{et} \quad \mathbf{R}_0 = \mathbf{S}_0 = 1.$$

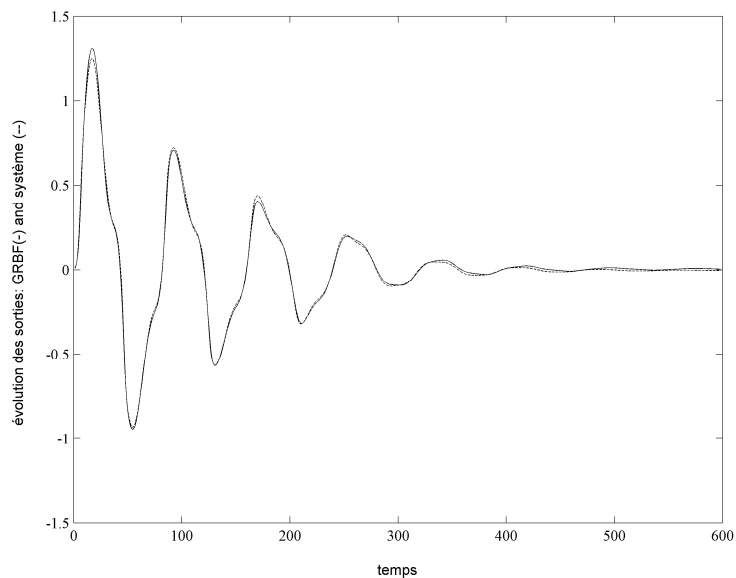


Figure 5.12 : Réponse du modèle GRBF et du système à une excitation sinusoïdale exponentiellement décroissante

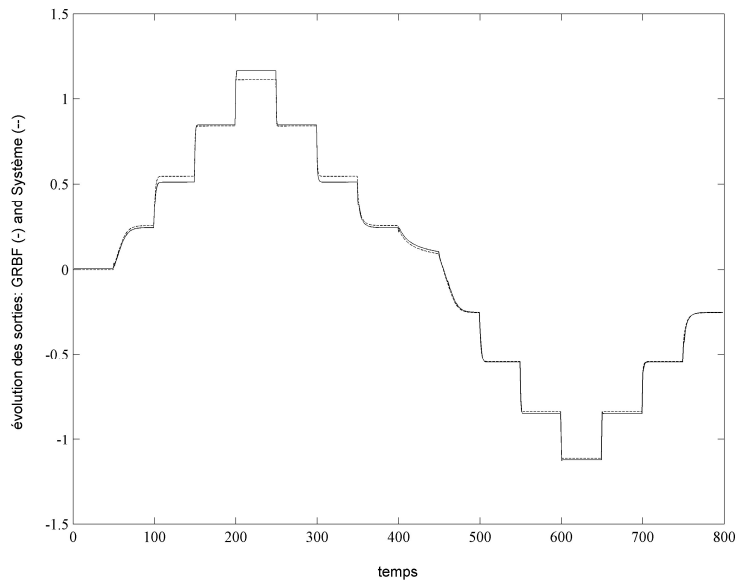


Figure 5.13 : Réponse du modèle GRBF et du système à une excitation en escalier

Le modèle GRBF identifié comprend vingt (20) centres et ses aptitudes à l'approximation sont testées avec des signaux non utilisés dans la phase d'apprentissage. La figure 5.12 représente la réponse du système non linéaire (5.49) et du modèle GRBF identifié pour une entrée : $u(t) = \exp(-0.005t)\sin\left(\frac{2\pi t}{75}\right)$.

La figure 5.13 illustre les réponses pour une entrée sous forme d'un escalier variant dans l'intervalle $[-1 \ 1]$ avec des pas de hauteur 0.25.

Ces simulations montrent clairement l'efficacité de l'algorithme RANEKF modifié dans l'apprentissage des réseaux GRBF et l'identification des systèmes dynamiques non linéaires. L'avantage principal de la philosophie du Réseau par Allocation des Ressources (Resource Allocating Network—RAN) réside dans la prise en charge totale des opérations de sélection et positionnement des centres ainsi que l'estimation des paramètres par un seul algorithme.

5.3.3 Emulation d'un modèle non linéaire dans l'espace d'état par une structure de réseaux RBF bouclée

L'écriture des systèmes dynamiques sous forme d'une représentation d'état permet de bénéficier directement de toute la richesse de la théorie et des outils disponibles pour l'analyse et la synthèse des systèmes de commande linéaires ou non linéaires basés sur des modèles d'état. Dans le cas d'un système linéaire l'identification d'un modèle d'état peut se réaliser, indirectement, par transformation d'un modèle empirique préalablement construit (ARX, ARMAX, ...) en un modèle d'état linéaire [28]. Une autre alternative, l'identification directe du modèle d'état, à l'aide d'une approche plus avancée connue sous l'appellation : identification par sous-espace (subspace identification) [25].

Dans le cas non linéaire, par contre, la situation est beaucoup plus compliquée et, la solution souvent adoptée est l'application des réseaux de neurones pour la paramétrisation d'une relation entrée-sortie d'un système non linéaire. En particulier, les réseaux de neurones récurrents sont très souvent utilisés dans l'identification non linéaire dans la forme d'une structure d'état [125,129, et 130]. Cette solution présente, néanmoins, certains inconvénients à cause de la structure complexe des réseaux récurrents (boucles de rétroaction internes) et des difficultés de leur apprentissage.

Nous proposons dans ce qui suit une structure de réseaux de neurones GRBF entraînée pour émuler une représentation d'espace d'état d'un système non linéaire discret en utilisant l'algorithme RANEKF modifié décrit dans la sous-section précédente (5.3.2) [50].

Le système dynamique non linéaire considéré est décrit par l'équation non linéaire suivante :

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad , \quad t = 0,1,2,\dots \quad (5.51)$$

Où $\mathbf{x}(t) \in \mathbf{X} \subset \mathfrak{R}^n$ est le vecteur d'état, $\mathbf{u}(t) \in \mathbf{U} \subset \mathfrak{R}^m$ représente le vecteur de commande. Les régions \mathbf{X} et \mathbf{U} sont des ensembles compacts de \mathfrak{R}^n et \mathfrak{R}^m , respectivement, et renfermant l'origine dans leur intérieurs. L'application $\mathbf{f} : \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}^n$ est une fonctionnelle non linéaire d'équations d'état continue et dérivable par rapport à ses arguments et vérifiant $\mathbf{f}(0,0) = 0$.

Nous assumons que les états du système (5.51) sont tous accessibles et, nous proposons de construire un prédicteur non linéaire récursif des n états :

$$\hat{\mathbf{x}}(t+1) = \hat{\mathbf{f}}(\hat{\mathbf{x}}(t), \mathbf{u}(t), \boldsymbol{\theta}) \quad , \quad t = 0, 1, 2, \dots \quad (5.52)$$

Où $\hat{\mathbf{f}}(\dots)$ est une approximation de la fonction non linéaire $f(\dots)$ et, $\boldsymbol{\theta}$ un vecteur contenant les paramètres du modèle. Nous décomposons ce prédicteur récursif en n prédicteurs séparés pour chacun des états. Chaque prédicteur d'un état est une fonction représentée par un réseau GRBF dont les arguments d'entrée sont les vecteurs d'état et de commande courants et, dont la sortie est la prédiction d'un élément du vecteur d'état $x(t)$:

$$\hat{x}_i(t+1) = \hat{f}_i(\hat{\mathbf{x}}(t), \mathbf{u}(t)) = w_{i0} + \sum_{k=1}^{K_i} w_{ik} \Phi_{ik}(\|(\hat{\mathbf{x}}(t), \mathbf{u}(t)) - \mathbf{c}_{ik}\|, \sigma_{ik}) \quad , \quad t = 0, 1, \dots \quad (5.53)$$

avec $i = 1, 2, \dots, n$, \hat{x}_i le i ème élément du prédicteur, Φ_{ik} sont des fonctions gaussiennes à base radiale, K_i est le nombre des centres du i ème élément du prédicteur, les w_{ik} sont les poids synaptiques du réseau. Le vecteur d'entrée de chaque réseau à l'instant t est :

$$(\hat{\mathbf{x}}(t), \mathbf{u}(t)) = [\hat{x}_1(t), \hat{x}_2(t), \dots, \hat{x}_n(t), u_1(t), \dots, u_m(t)]^T \quad (5.54)$$

Les fonction gaussiennes Φ_{ik} sont données par :

$$\Phi_{ik}(\hat{\mathbf{x}}(t), \mathbf{u}(t)) = \exp\left(-\frac{\|(\hat{\mathbf{x}}(t), \mathbf{u}(t)) - \mathbf{c}_{ik}\|^2}{\sigma_{ik}^2}\right) \quad (5.55)$$

où \mathbf{c}_{ik} et σ_{ik} représentent les centres et les largeurs du i ème réseaux, respectivement.

En conséquence, l'architecture du prédicteur proposé consiste en une structure parallèle de n réseaux GRBF récursifs ayant le vecteur $(\hat{\mathbf{x}}(t), \mathbf{u}(t))$ à l'entrée et produisant la prédiction $\hat{\mathbf{x}}(t+1)$ à la sortie et nous pouvons écrire

$$\hat{\mathbf{x}}(t+1) = \hat{\mathbf{f}}(\hat{\mathbf{x}}(k), \mathbf{u}(k)) = \begin{bmatrix} \hat{f}_1(\hat{\mathbf{x}}(k), \mathbf{u}(k)) \\ \hat{f}_2(\hat{\mathbf{x}}(k), \mathbf{u}(k)) \\ \vdots \\ \vdots \\ \vdots \\ \hat{f}_n(\hat{\mathbf{x}}(k), \mathbf{u}(k)) \end{bmatrix} \quad (5.56)$$

Cette architecture est illustrée par la figure (5.14)

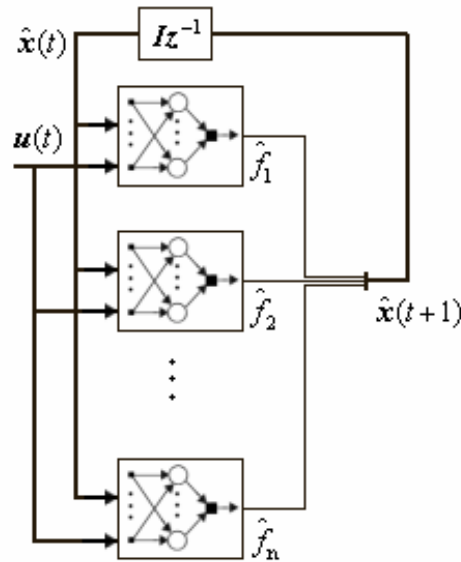


Figure 5.14: Structure GRBF pour l'émulation d'une représentation d'état
D'un système non linéaire

Maintenant que la structure générale du réseau de neurones GRBF pouvant émuler une représentation d'espace d'état telle que celle d'un système non linéaire discret décrit par l'équation (5.51), l'étape suivante est de calculer les paramètres du réseau et tester ses aptitudes et sa précision d'approximation. Pour ce faire, nous revenons au premier exemple de simulation dans la sous-section 5.3.2 et, qui est décrit par l'équation différentielle non linéaire (5.46).

En posant la masse $m=1.0\text{kg}$ et, en discrétisant l'équation différentielle à l'aide d'un développement de Taylor de deuxième ordre, nous obtenons le système d'équations discret non linéaire suivant :

$$\begin{aligned} x_1(t+1) &= x_1(t) + \left[T - \frac{T^2}{2} \right] x_2(t) - 9.8 \frac{T^2}{2} \sin x_1(t) + \frac{T^2}{2} u(t) \\ x_2(t+1) &= \left[1 - T - \frac{T^2}{2} - 9.8 \frac{T^2}{2} \cos x_1(t) \right] x_2(t) - 9.8 \left[T - \frac{T^2}{2} \right] \sin x_1(t) + \left[T - \frac{T^2}{2} \right] u(t) \end{aligned} \quad (5.57)$$

Les deux variables d'état sont $x_1(t)=\theta(t)$ et $x_2(t)=\dot{\theta}(t)$, les position et vitesse angulaires, respectivement, et $T=0.01\text{s}$ est la période d'échantillonnage.

On définit le domaine d'opération du manipulateur par les régions $X \subset \mathbb{R}^2$ et $U \subset \mathbb{R}^1$ tel que: $X = \{x = [x_1, x_2]^T ; /x_1| \leq 0.7\text{rd} , /x_2| \leq 2.5\text{rd/s}\}$, et $U = \{u ; /u| \leq 7\text{Nm}\}$.

Dans cette simulation le vecteur d'entrée du prédicteur est défini par :

$$\text{col}(x(t), u(t)) = [x_1(t), x_2(t), u(t)]^T \quad (5.58)$$

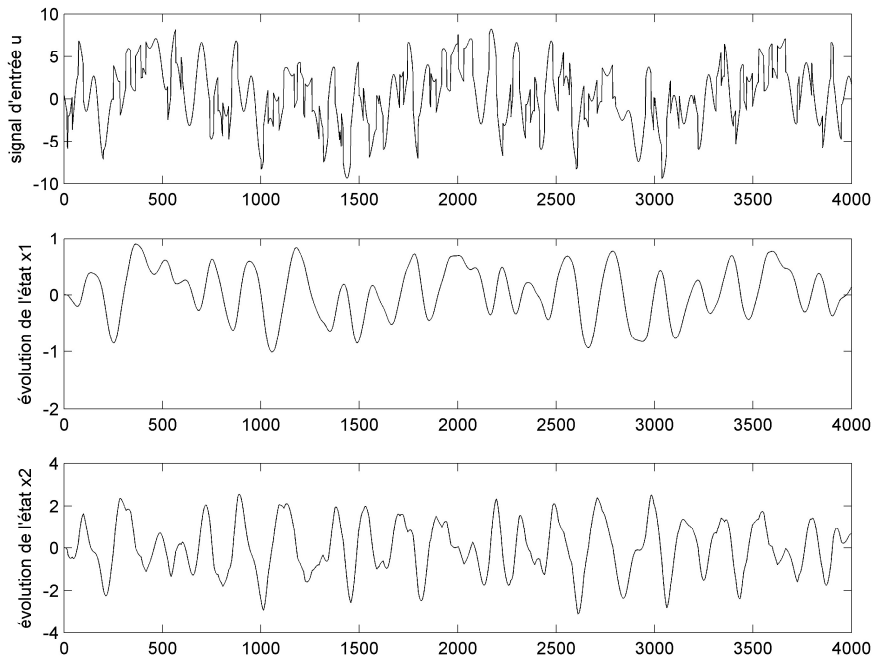


Figure 5.15 : Données d'identification

L'émulation du système (5.57) par une structure similaire à celle illustrée par la figure (5.14) nécessite l'apprentissage de deux réseaux GRBF pour l'approximation des états $x_1(t)$ et $x_2(t)$. Les données d'apprentissage sont collectées par excitation du système discret (5.57) par des signaux synthétiques formés par des fonctions en escalier superposées à des sommes de sinusoides. Ces signaux sont illustrés par la figure 5.15. La phase d'apprentissage à nécessité la présentation de ces données à l'algorithme RANEKF modifié en cinq passes successives ; c'est-à-dire que la matrice d'apprentissage est constituée de vingt mille (20000) échantillons. Les paramètres de l'algorithme sont fixés comme suit : $\varepsilon_{\max} = 1$, $\varepsilon_{\min} = 1$, $\gamma = 1$, $\eta = 0.01$, $\rho = 0.97$, $\mathbf{Q}_0 = 0.03$ et $\mathbf{R}_0 = \mathbf{S}_0 = 1$.

Les réseaux prédicteurs des états $x_1(t)$ et $x_2(t)$ sont constitués de 78 centres chacun. Les deux réseaux sont testés par un signal non utilisé dans la phase

d'apprentissage et formé d'une somme de sinusoïdes. Les résultats du test sont montrés dans les figures 5.16 à 5.18.

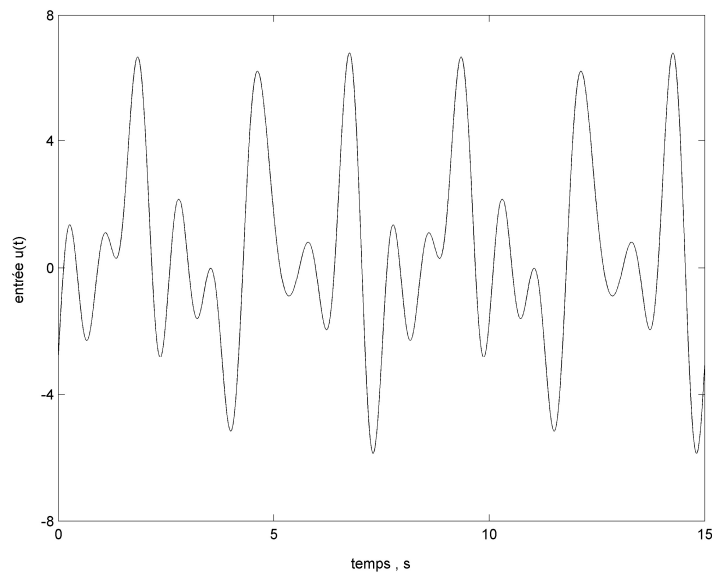


Figure 5.16 : Signal d'excitation utilisé pour la validation du modèle

Après ce test sur les aptitudes d'interpolation de la structure GRBF identifiée, il convient aussi de comparer ses propriétés de stabilité locale avec celles du système 'réel' décrit par les équations (5.75). Une méthode pour tester la stabilité

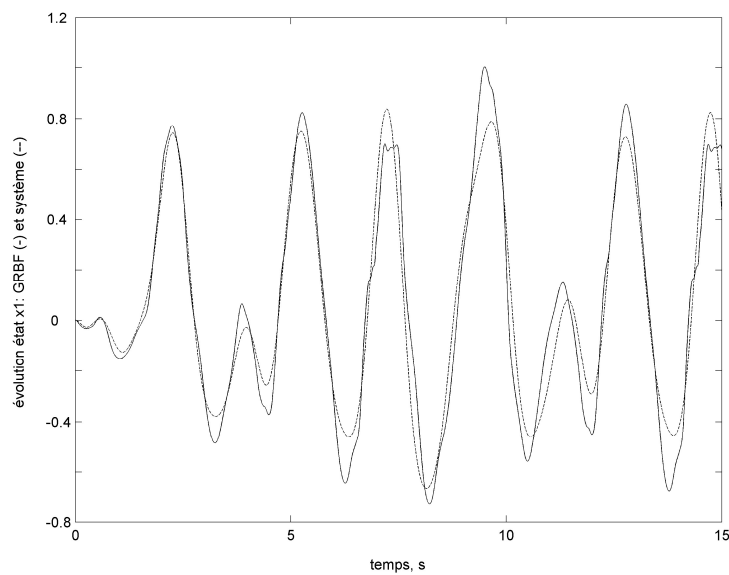


Figure 5.17 : Comparaison des réponses du système et du modèle (variable d'état x_1)

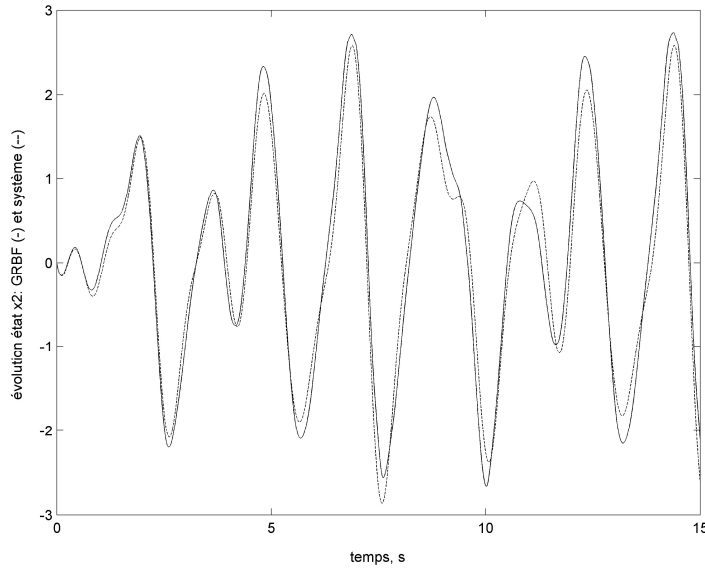


Figure 5.18: Comparaison des réponses du système et du modèle (variable d'état x2)

du réseau consiste à comparer les matrices Jacobiennes $\mathbf{A} = \left. \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}}$ et

$\hat{\mathbf{A}} = \left. \frac{\partial \hat{f}(\hat{\mathbf{x}}, \mathbf{u})}{\partial \hat{\mathbf{x}}} \right|_{\substack{\hat{\mathbf{x}}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}}$ du système 'réel' et du réseau GRBF, respectivement [129]. Si les

valeurs propres des matrices \mathbf{A} et $\hat{\mathbf{A}}$ se rapprochent les une des autres, alors le système identifié et son modèle GRBF sont considérés

avoir des dynamiques équivalentes, y compris les propriétés de stabilité autour du

point $(\mathbf{x}_0, \mathbf{u}_0)$. En plus, Les éléments des matrices $\mathbf{B} = \left. \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \right|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}}$ et

$\hat{\mathbf{B}} = \left. \frac{\partial \hat{f}(\hat{\mathbf{x}}, \mathbf{u})}{\partial \mathbf{u}} \right|_{\substack{\hat{\mathbf{x}}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}}$ doivent être similaires pour avoir des dynamiques équivalentes.

Dans cette simulation, nous avons d'abord testé et comparé les propriétés de stabilité du modèle et du système 'réel' autour de l'origine $(\mathbf{x}_0, \mathbf{u}_0) = (0,0)$. Les

matrices \mathbf{A} , $\hat{\mathbf{A}}$ et leurs valeurs propres sont données par

$$\mathbf{A} = \begin{bmatrix} 0.99951 & 0.00995 \\ -0.09751 & 0.98956 \end{bmatrix}, \quad \hat{\mathbf{A}} = \begin{bmatrix} 0.99597 & 0.01108 \\ -0.09821 & 0.98783 \end{bmatrix} \quad (5.59)$$

$$\zeta_{\mathbf{A}} = \begin{bmatrix} 0.9945 + 0.03074i \\ 0.9945 - 0.03074i \end{bmatrix}, \quad \zeta_{\hat{\mathbf{A}}} = \begin{bmatrix} 0.9919 + 0.03274i \\ 0.9919 - 0.03274i \end{bmatrix} \quad (5.60)$$

Et les matrices \mathbf{B} et $\hat{\mathbf{B}}$ sont

$$\mathbf{B} = \begin{bmatrix} 0.00005 \\ 0.00995 \end{bmatrix}, \quad \hat{\mathbf{B}} = \begin{bmatrix} -0.00055 \\ 0.01074 \end{bmatrix} \quad (5.61)$$

On peut déjà conclure de l'équivalence dynamique du système 'réel' et du modèle GRBF identifié autour du point $(x_0, u_0) = (0,0)$. Cette méthode de comparaison est aussi appliquée pour des points éloignés de l'origine dans le but de s'assurer que le modèle est fiable et assez représentatif de la dynamique du système 'réel'

(x_{01}, x_{02}, u_0)	Valeurs propre de la matrice \mathbf{A}	Valeurs propre de la matrice $\hat{\mathbf{A}}$
(0 , 0 , 0)	$0.99453 \pm 0.030749i$	$0.99190 \pm 0.032747i$
(0.2 , 0 , 1.95)	$0.99454 \pm 0.030432i$	$0.99537 \pm 0.022741i$
(0.4 , 0 , 3.82)	$0.99457 \pm 0.029477i$	$0.99589 \pm 0.025258i$
(0.6 , 0 , 5.53)	$0.99462 \pm 0.027857i$	$0.99359 \pm 0.037889i$
(-0.2 , 0 , -1.95)	$0.99454 \pm 0.030432i$	$0.98903 \pm 0.043780i$
(-0.4 , 0 , -3.82)	$0.99457 \pm 0.029477i$	$0.98145 \pm 0.022426i$
(-0.6 , 0 , -5.53)	$0.99462 \pm 0.027857i$	$0.98872 \pm 0.029596i$

Tableau 5.1

sur tout le domaine d'opération tel que défini précédemment par les régions $X \subset \mathbb{R}^2$ et $U \subset \mathbb{R}^1$. Le tableau (5.1) résume les résultats de comparaison et indique, clairement, que la structure GRBF proposée et identifiée a bien capturé

les propriétés dynamiques du manipulateur décrit par (5.57) dans son domaine d'opération.

5.4 Identification non linéaire inverse

La modélisation inverse des systèmes dynamiques joue un rôle important dans l'élaboration de certaines stratégies de commande non linéaires [131-133]. Parmi ces techniques de commande qui peuvent intégrer un modèle inverse, nous citons : (1) La commande à modèle interne (Internal Model Control-IMC) [128, 134] , (2) la commande à modèle de référence (Model Reference control) [27] et, (3) La commande prédictive [28].

En partant de l'hypothèse que la fonction $f : \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}^n$ dans (5.51) est une bijection on peut, alors, construire sa fonction inverse. Cette fonction inverse représente la dynamique inverse du système décrite par (5.52) et est donnée par :

$$\mathbf{u}(t) = f^{-1}(\mathbf{x}(t+1), \mathbf{x}(t)) \quad (5.62)$$

La fonction f^{-1} prend comme ses arguments les valeurs courantes et futures de l'état et, fourni la valeur courante de la commande. Donc, pour une trajectoire de référence $\mathbf{x}^r(t)$ dans l'espace d'état, connue à priori, la fonction inverse f^{-1} donne la valeur de la commande $\mathbf{u}^r(t)$ qui permet au système de suivre exactement cette trajectoire de référence. Pour construire la fonction inverse f^{-1} , nous allons faire recours à la même structure GRBF décrite dans la sous-section 5.3.3 et illustrée par la figure (5.14) ainsi qu'à l'algorithme d'apprentissage RANEKF modifié introduit dans la sous-section 5.3.2. Pour un système à plusieurs entrées ($\mathbf{u} \in \mathfrak{R}^m$), l'élément j de vecteur de référence $\mathbf{u}^r(t)$ correspondant à une trajectoire $\mathbf{x}^r(t)$ sera décrit par un réseau GRBF :

$$\begin{aligned} u_j^r(t) &= f_j^{-1}(\mathbf{x}^r(t+1), \mathbf{x}^r(t)) \\ &= w_{j0} + \sum_{k=1}^{K_j} w_{jk} \Phi_{jk}(\|\mathbf{x}^r(t+1), \mathbf{x}^r(t) - \boldsymbol{\mu}_{jk}\|, \delta_{jk}) \end{aligned} \quad j=1,2,\dots,m \quad (5.63)$$

avec

$$\Phi_{jk}(\mathbf{x}^r(t+1), \mathbf{x}^r(t)) = \exp\left(-\frac{\|\mathbf{x}^r(t+1), \mathbf{x}^r(t) - \boldsymbol{\mu}_{jk}\|^2}{\delta_{jk}^2}\right) \quad (5.64)$$

et où $\mu_{jk} \in \mathfrak{R}^{2n}$ représentent les centres et $\delta_{jk} \in \mathfrak{R}$ les largeurs . Le nombre des unités gaussiennes dans le j ème réseau est K_j et les poids synaptiques sont les w_{jk} , avec w_{j0} un biais .

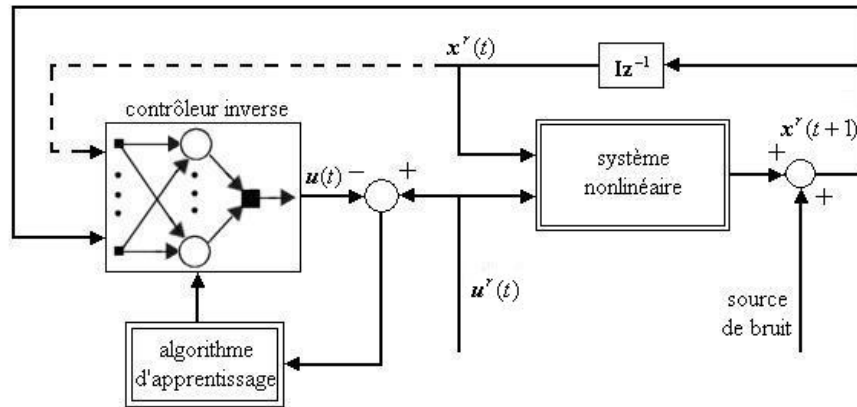


Figure 5.19: Schéma d'entraînement du modèle inverse

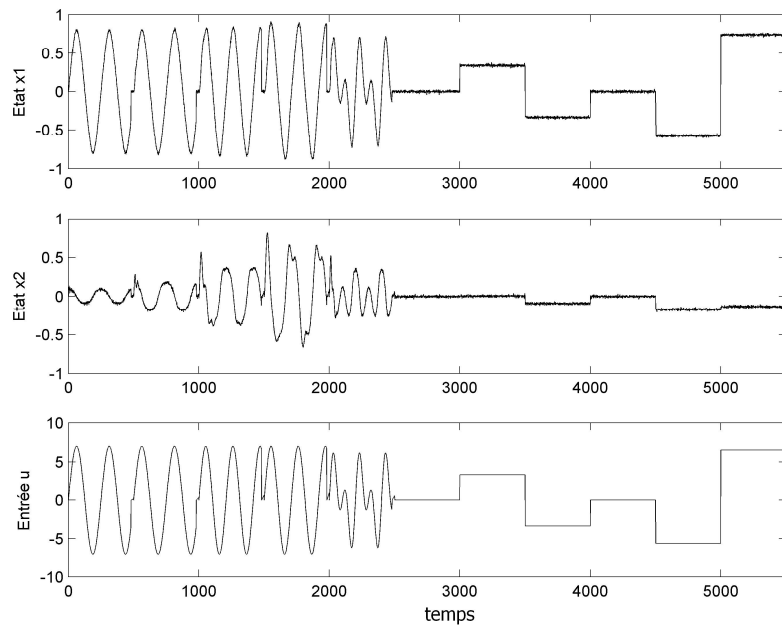


Figure 5.20 : Données utilisées pour l'identification du modèle inverse

Le schéma d'apprentissage adopté, dans ce travail, pour l'identification inverse repose sur l'approche de l'inversion directe introduite par [135] et cité par [28] . Dans ce schéma, illustré par la figure (5.19), un signal de commande

synthétique $u^r(t)$ est appliqué au système non linéaire et, les états correspondants sont mesurés pour être utilisés comme données d'apprentissage.

L'algorithme RANEKF modifié est alimenté avec N mesures sur l'état et l'entrée. Durant la k ième itération, l'algorithme reçoit le vecteur $(x^r(k+1), x^r(k))$ à l'entrée et la commande correspondante $u^r(k)$ est prise pour cible de l'apprentissage supervisé.

Dans ce travail, nous allons construire le model GRBF inverse du système (5.57). Les données de l'apprentissage sont générées en appliquant plusieurs entrées sous formes de fonction sinusoïdales ajoutées à une suite de variations en échelons comme le montre la figure 5.20. Nous assumons que le bruit est un processus gaussien de valeur moyenne nulle et de variance 0.01. Les entrées d'excitation varient dans l'intervalle $\pm 7N.m$ et les paramètres de l'algorithme sont fixés comme suit :

$$\varepsilon_{\min} = \varepsilon_{\max} = 0.175, \quad \gamma = 1, \quad \eta = 0.07, \quad \rho = 0.85, \quad \mathbf{Q}_0 = 0.025, \quad \text{and} \quad \mathbf{R}_0 = \mathbf{S}_0 = 1$$

Lors de l'apprentissage, il nous paraissait que la réduction du vecteur d'entrée aux seules valeurs de l'état future $[x_1(t+1), x_2(t+1)]^T$ ne dégrade pas la performance

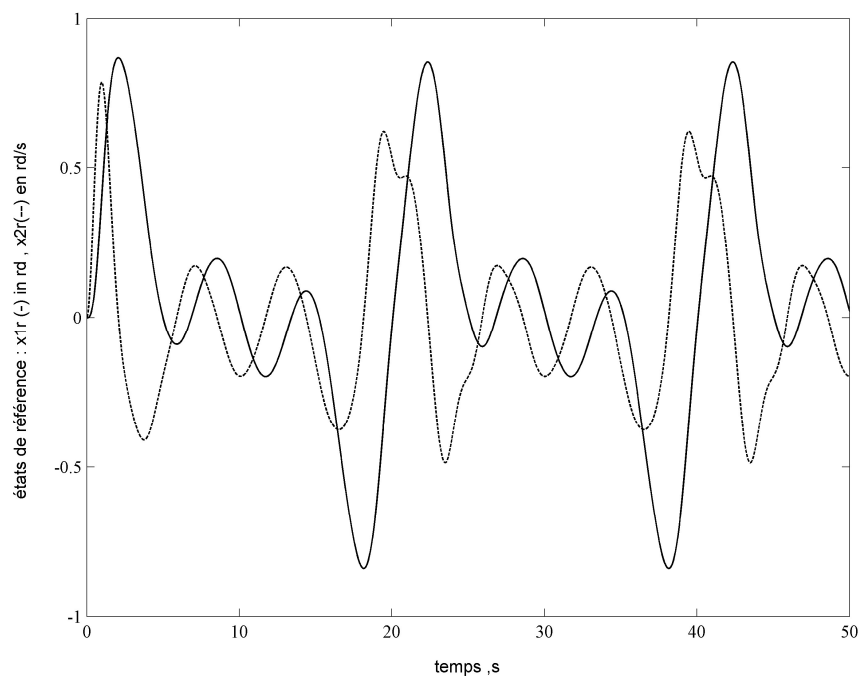


Figure 5.21 : Trajectoire d'état de référence

du réseau et réduit considérablement le temps du calcul. Le test de performance du réseau GRBF inverse identifié, formé de 46 fonctions gaussiennes, est illustré par les figures 5.21 et 5.22.

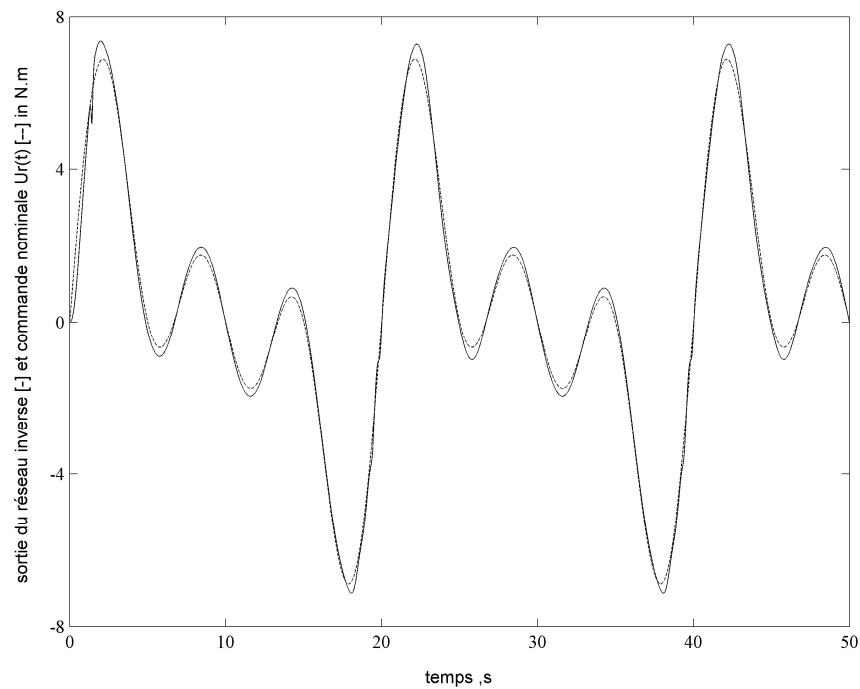


Figure 5.22 : Sortie du modèle inverse comparée au signal d'excitation désiré

Chapitre 6

Commande prédictive stabilisante basée sur une classe de modèles neuroniques à fonctions gaussiennes

6.5 Les modèles neuroniques dans la commande prédictive non linéaire

Dans les chapitres 1 et 2 nous avons présenté les bases théoriques de la stabilisation non linéaire par la commande prédictive pour des systèmes dynamiques discrets décrits par des équations d'état. Dans sa grande majorité, la littérature publiée se limite au traitement de la régulation stabilisante [24] (déplacement asymptotique de l'état vers l'origine ou tout autre point d'équilibre statique) et, peu de travaux sont consacrés à la stabilisation dans le cas d'une commande en poursuite de trajectoire dans le cadre d'un schéma prédictif non linéaire [39]. Les contributions sont, encore, plus rares pour la stabilisation en poursuite par commande prédictive non linéaire basée sur des modèles empiriques [35] (NARX, réseaux de neurones, ...). Pour le cas particulier de la stabilisation non linéaire par la commande prédictive référencée modèle neuronique (en régulation et/ou en poursuite), nous avons répertorié, essentiellement, trois contributions directement liées au sujet [34,36, et 125] malgré l'existence d'une vaste littérature sur les applications des réseaux de neurones dans l'élaboration de schémas divers de la commande prédictive [29] et, ceci depuis l'article pionnier de Narrendra et al. [27].

Parmi les facteurs contribuant au succès de la mise en œuvre d'une commande prédictive, la disponibilité d'un modèle fiable et fidèle du processus reste primordiale. En effet, sans la présence de ce modèle les erreurs de prédictions des variables d'état seront assez importantes pour causer des instabilités et/ou des problèmes de robustesse. Ceci est d'autant plus vrai pour le

cas d'un processus non linéaire. C'est pourquoi, beaucoup d'efforts ont été consacrés à la modélisation et l'identification non linéaire (Voir chapitre 5) pour construire des modèles qui reflètent, le plus fidèlement possible, les complexités et les propriétés dynamiques du processus sous commande. C'est dans ce contexte que les réseaux de neurones ont émergés comme une alternative prometteuse aux techniques d'identification existantes jusqu'au début des années 90 et, ceci à cause de leurs capacités d'apprentissage et d'approximation.

La liste des contributions sur les applications des modèles neuroniques dans l'élaboration des commandes prédictives non linéaires est trop longue pour être revue en entier et, nous citerons ici quelques travaux précurseurs tels ceux de Hunt et al. [28] et Willis et al.[112], qui proposèrent une architecture de commande prédictive où des modèles à réseaux de neurones fournissent les prédictions futures des sorties du processus à l'algorithme d'optimisation qui calcul la séquence des signaux de commande sur un horizon de prédiction. La contribution de Tan et al. [113] décrit un schémas de commande adaptative prédictive à modèle neuronique avec une mise à jour en temps réel des paramètres du perceptron par un algorithme RLS, pour accommoder et s'adapter aux variations continues du système. Une autre structure est proposée par Liu et al.[124] et dans laquelle l'algorithme RAN de Platt [117] est utilisé dans l'élaboration d'une commande prédictive généralisée (GPC) basée sur un model GRBF dynamique. Dans les travaux de Zhu et al. [136], une loi de commande prédictive non linéaire auto-ajustable est dérivée pour un processus décrit par un model linéaire ARMA couplé à un terme additif non linéaire représenté par un réseaux de neurone multicouches. Dans cette structure les paramètres du modèle ARMA sont estimés en-ligne par un algorithme RLS et, les poids du réseaux neuronique par une procédure de rétro propagation.

D'autres contributions, les unes aussi importantes que les autres furent avancées, mais nous citerons le travail de Neumerkel et al.[114] pour son application en temps réel d'une commande prédictive à modèle RBF appliquée à une machine asynchrone et celui de Sbarbaro et al.[115] sur l'applicabilité des réseaux RBF dans différents niveaux d'un système industriel automatisé. L'option RBF semble acquérir de plus en plus la confiance des concepteurs des produits de technologie prédictive tel Predictive Control Ltd. (PCL) avec son produit

‘Connoisseur’ [116] capable d’intégrer des modèles RBF pour prendre en charge les processus industriels fortement non linéaires.

Dans ce chapitre nous proposons d’intégrer la structure GRBF décrite dans le chapitre 5 (sous-section 5.3.3) dans une stratégie de commande prédictive non linéaire stabilisante. Notre proposition englobe, aussi bien, la stabilisation en régulation qu’en poursuite de trajectoire. Dans ce dernier cas, nous introduisons une nouvelle structure de commande basée sur une propriété analytique intéressante de la structure réseaux GRBF introduite dans la sous-section 5.3.3 et, qui permet enfin, de transformer un problème de commande non linéaire en poursuite de trajectoire en un problème de régulation. Cette transformation parfaitement réalisable pour les systèmes linéaires, restait difficile dans le cas non linéaire [39]. Son importance provient surtout du fait qu’elle permet de simplifier la problématique de stabilisation non linéaire et d’appliquer aisément les algorithmes stabilisants disponibles. Dans le reste de ce chapitre nous formulerons la problématique générale d’une commande prédictive en poursuite de trajectoire pour les systèmes non linéaires. Enfin, nous avançons notre structure de commande prédictive stabilisante basée sur des modèles GRBF et, nous démontrons les performances par des simulations.

6.6 Problématique d’une commande prédictive non linéaire en poursuite de trajectoire dans l’espace d’état

6.6.1 Description du système

Le système dynamique considéré est un système discret non linéaire décrit par l’équation d’état :

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad , \quad t = 0, 1, 2, \dots \quad (6.1)$$

Avec $\mathbf{x}(t) \in \mathbf{X} \subset \mathbb{R}^n$ le vecteur d’état, et $\mathbf{u}(t) \in \mathbf{U} \subset \mathbb{R}^m$ le vecteur de commande. Les domaines \mathbf{X} et \mathbf{U} sont des sous-ensembles compacts de \mathbb{R}^n et \mathbb{R}^m , respectivement et contenant l’origine. On admet, par ailleurs, que :

Hypothèse 1. La fonction $f(.,.) : \mathfrak{R}^{n \times m} \rightarrow \mathfrak{R}^n$ est une bijection continue et dérivable par rapport à ses arguments et admet l'origine comme point d'équilibre ($f(0,0) = 0$).

Hypothèse 2. Tous les états du système (6.1) sont accessibles et observables.

Hypothèse 3. Pour tout état d'équilibre $(x_e, u_e) \in \mathbf{X} \times \mathbf{U}$, le système linéaire $x_l(t+1) = \mathbf{A}x_l(t) + \mathbf{B}u(t)$ obtenu par linéarisation de (6.1) au voisinage de (x_e, u_e) ;

$$\mathbf{A} = \left. \frac{\partial f(x, u)}{\partial x} \right|_{\substack{x=x_e \\ u=u_e}} \quad \text{et} \quad \mathbf{B} = \left. \frac{\partial f(x, u)}{\partial u} \right|_{\substack{x=x_e \\ u=u_e}}$$

est stabilisable.

6.6.2 Problème optimal de poursuite de trajectoire dans l'espace d'état

Pour le système (6.1), l'objectif est de synthétiser une loi de commande $u(t) = \kappa_{tr}(t)$ capable d'entraîner le système sur une trajectoire de référence dans l'espace d'état $x_d(t) \in \mathbf{X}$ à partir d'un état initial $x_0 = x(t_0)$ pour tout $t \geq t_0$ et, tel que :

1. La fonction de coût quadratique suivante

$$J = \sum_{i=t}^{t+N_c} (x(i) - x_d(i))^T \mathbf{Q} (x(i) - x_d(i)) + (u(i) - u_d(i))^T \mathbf{R} (u(i) - u_d(i)) \quad (6.2)$$

$$\mathbf{Q} > \mathbf{0} \quad , \quad \mathbf{R} > \mathbf{0}$$

est minimisée.

2. Le système nominal $x(t+1) = f(x(t), u(t))$ est asymptotiquement stable.
3. $\lim_{t \rightarrow \infty} |x(t) - x_d(t)| = 0$.

Dans cette formulation du problème de poursuite dans l'espace d'état, la trajectoire $x_d(t) \in \mathbf{X}$ est soit, un état de référence constant ou, une multitudes de références constantes prenant la forme d'une fonction en escaliers. La valeur de commande $u_d(.)$ représente la commande nominale correspondante à l'état de référence $x_d(.)$ et, N_c est un entier positif qui dénote l'horizon de prédiction.

Avant de proposer une solution de ce problème de commande en poursuite dans l'espace d'état dans le cas général d'un système non linéaire, nous allons restreindre le problème pour un modèle linéaire et définir les contours de l'approche à suivre dans le cas non linéaire.

Considérons, donc, le cas particulier où le système (6.1) est linéaire et, est décrit par :

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (6.3)$$

Lorsque le système (6.3) atteint un état d'équilibre statique $(\mathbf{x}_d, \mathbf{u}_d)$, (6.3) s'écrit :

$$\mathbf{x}_d = \mathbf{A}\mathbf{x}_d + \mathbf{B}\mathbf{u}_d \quad (6.4)$$

Par soustraction des équations (6.3) et (6.4) et en définissant les vecteurs :

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d \quad (6.5)$$

$$\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}_d \quad (6.6)$$

On obtient le système linéaire suivant :

$$\tilde{\mathbf{x}}(t+1) = \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{B}\tilde{\mathbf{u}}(t) \quad (6.7)$$

et la fonction coût (6.2) devient, alors :

$$J = \sum_{i=t}^{t+N_c} \tilde{\mathbf{x}}(i)^T \mathbf{Q}\tilde{\mathbf{x}}(i) + \tilde{\mathbf{u}}(i)^T \mathbf{R}\tilde{\mathbf{u}}(i) \quad (6.8)$$

$$\mathbf{Q} > \mathbf{0}, \mathbf{R} > \mathbf{0}$$

Où $\tilde{\mathbf{x}} \in \tilde{\mathbf{X}}$ et $\tilde{\mathbf{u}} \in \tilde{\mathbf{U}}$. Les domaines $\tilde{\mathbf{X}}$ et $\tilde{\mathbf{U}}$ étant des sous-ensembles compacts de \mathbb{R}^n et \mathbb{R}^m , respectivement, contenant l'origine comme point intérieur. Le problème de poursuite optimal dans l'espace d'état définie par (6.1) et (6.2) est, donc, transformé en un simple problème de régulation représenté par les équations (6.7) et (6.8).

La résolution de ce problème d'optimisation linéaire pour un horizon infini $N_c = \infty$ (Méthode de Rawling-Muske ; sous-section 2.4.2) sous les contraintes $\tilde{\mathbf{x}} \in \tilde{\mathbf{X}}$ et $\tilde{\mathbf{u}} \in \tilde{\mathbf{U}}$ aboutit à une commande prédictive asymptotiquement stabilisante $\tilde{\mathbf{u}}(t)$ pour le système (6.7) et, nous aurons nécessairement

$$\lim_{t \rightarrow \infty} |\tilde{\mathbf{x}}(t)| = \lim_{t \rightarrow \infty} |\mathbf{x}(t) - \mathbf{x}_d(t)| = 0 \quad (6.9)$$

et la loi de commande pour le système linéaire original (6.3) est donnée, en tenant compte de (6.6), par :

$$\begin{aligned} \mathbf{u}(t) &= \mathbf{u}_r(t) \\ &= \tilde{\mathbf{u}}(\tilde{\mathbf{x}}(t)) + \mathbf{u}_d(t) \end{aligned} \quad (6.10)$$

Ceci démontre que la commande en poursuite d'une trajectoire $\mathbf{x}_d(t) \in \mathbf{X}$ dans l'espace d'état du système (6.3) est composée de deux éléments : un premier composant prédictif $\tilde{\mathbf{u}}(\tilde{\mathbf{x}}(t))$ exerçant une action stabilisatrice par retour

d'état et, un deuxième composant à action directe $u_d(t)$ pour le maintien du système dans l'état désiré $x_d(t)$. La projection de cette technique vers le domaine non linéaire se heurte à la difficulté que constitue la transformation du problème de poursuite de trajectoire (trajectory tracking) en un problème de régulation. Dans [137], Park et al. ont proposé de réaliser cette stratégie de commande en entraînant deux réseaux de neurones multicouches pour approximer la composante prédictive $\tilde{u}(\cdot)$ et la composante directe $u_d(\cdot)$ pour le cas d'une poursuite de trajectoire dans l'espace de sortie. Cependant, cette approche n'apporte pas une solution formelle et convaincante au problème de la stabilité car, l'entraînement du réseau approximant la composante $\tilde{u}(\cdot)$ suit une méthode *ad hoc* basée sur le principe du *teste et de l'erreur* (trial and error) en utilisant une procédure d'apprentissage complexe conçu autour de l'algorithme de rétro propagation généralisée (generalised-backpropagation-through-time, GBTT). Dans leur technique, Park et al. n'abordent pas la question de transformer le problème de poursuite en un problème de régulation ce qui ne permet pas, en conséquence, de bénéficier des propriétés stabilisantes des régulateurs prédictifs non linéaires existants.

6.7 Une architecture générale pour une commande prédictive stabilisante référencée modèle GRBF

Dans [50], Ahmida et al. adoptent une philosophie similaire à celle de Park et al. pour une classe de systèmes non linéaires identifiés par une structure de réseaux GRBF dans l'espace d'état (voire sous-section 5.3.3) et, proposent une commande à deux composantes : l'une garantissant l'action stabilisatrice est calculée, en-ligne, sous forme d'une régulation non linéaire prédictive à horizon fini avec fonction de coût terminale et l'autre, sous forme d'un réseau GRBF inverse capable de générer l'action $u_d(t)$ pour toute valeur $x_d(t)$ de la trajectoire de référence dans l'espace d'état (voire section 5.4). Le résultat est une nouvelle structure de commande intégrant les avantages d'une régulation non linéaire prédictive stabilisante avec ceux d'une commande neuronique (neural controller) pour atteindre de bonnes performances de stabilité et de poursuite.

6.7.1 Génération des déviations des variables d'état pour une classe de systèmes non linéaires commandés en poursuite de trajectoire

Dans le chapitre 5 (sous-section 5.3.3), nous avons présenté une structure de réseaux de neurones GRBF entraînée pour émuler une représentation d'espace d'état d'un système non linéaire discret tel celui décrit par (6.1). Nous considérons, donc, dans ce qui suit que les n éléments du de la fonction-vecteur $f(.,.): \mathfrak{R}^{n \times m} \rightarrow \mathfrak{R}^n$ sont les sorties de n réseaux GRBF ayant le vecteur d'entrée $(\mathbf{x}(t), \mathbf{u}(t))$ et, dont la sortie pour le i ème élément est:

$$\mathbf{x}_i(t+1) = f_i(\mathbf{x}(t), \mathbf{u}(t)) = w_{i0} + \sum_{k=1}^{K_i} w_{ik} \Phi_{ik}(\|\mathbf{x}(t), \mathbf{u}(t) - \mathbf{c}_{ik}\|, \sigma_{ik}), \quad t = 0, 1, \dots \quad (6.11)$$

avec $i = 1, 2, \dots, n$, x_i le i ème élément du prédicteur, Φ_{ik} sont des fonctions gaussiennes à base radiale, K_i est le nombre des centres du i ème élément du prédicteur, w_{i0} un biais et, les w_{ik} sont les poids synaptiques du réseau. Le vecteur d'entrée de chaque réseau à l'instant t est :

$$(\mathbf{x}(t), \mathbf{u}(t)) = [x_1(t), x_2(t), \dots, x_n(t), u_1(t), \dots, u_m(t)]^T \quad (6.12)$$

Les fonction gaussiennes Φ_{ik} sont données par :

$$\Phi_{ik}(\mathbf{x}(t), \mathbf{u}(t)) = \exp\left(-\frac{\|\mathbf{x}(t), \mathbf{u}(t) - \mathbf{c}_{ik}\|^2}{\sigma_{ik}^2}\right) \quad (6.13)$$

où \mathbf{c}_{ik} et σ_{ik} représentent les centres et les largeurs du i ème réseau, respectivement. Les contraintes $\mathbf{x}(t) \in \mathbf{X} \subset \mathfrak{R}^n$ et $\mathbf{u}(t) \in \mathbf{U} \subset \mathfrak{R}^m$ sur l'état et la commande, ainsi que les hypothèses 1, 2, et 3 de la section 6.2 restent valables.

En utilisant l'équation (6.11) et la définition de la fonction gaussienne à base radiale (6.13), nous pouvons réécrire le i ème élément de l'équation (6.7) de la déviation du vecteur d'état par rapport à la trajectoire de référence ainsi :

$$\begin{aligned} \tilde{\mathbf{x}}_i(t+1) &= f_i(\mathbf{x}(t), \mathbf{u}(t)) - f_i(\mathbf{x}_d(t), \mathbf{u}_d(t)) \\ &= \sum_{k=1}^{K_i} q_{ik} \left[\exp\left(-\frac{\|\mathbf{x}(t), \mathbf{u}(t) - \mathbf{c}_{ik}\|^2}{\sigma_{ik}^2}\right) - \exp\left(-\frac{\|\mathbf{x}_d(t), \mathbf{u}_d(t) - \mathbf{c}_{ik}\|^2}{\sigma_{ik}^2}\right) \right] \end{aligned} \quad (6.14)$$

Par manipulation du terme $\|(\mathbf{x}(t), \mathbf{u}(t)) - \mathbf{c}_{ik}\|^2$ dans l'équation (6.14) par addition et soustraction du vecteur $(\mathbf{x}_d(t), \mathbf{u}_d(t))$, on obtient :

$$\begin{aligned} \|(\mathbf{x}(t), \mathbf{u}(t)) - \mathbf{c}_{ik}\|^2 &= \|(\mathbf{x}(t), \mathbf{u}(t)) - (\mathbf{x}_d(t), \mathbf{u}_d(t)) + (\mathbf{x}_d(t), \mathbf{u}_d(t)) - \mathbf{c}_{ik}\|^2 \\ &= \|(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) - (\mathbf{c}_{ik} - (\mathbf{x}_d(t), \mathbf{u}_d(t)))\|^2 \end{aligned} \quad (6.15)$$

En définissant $\tilde{\mathbf{c}}_{ik}(t)$ comme la déviation du centre de la gaussienne :

$$\tilde{\mathbf{c}}_{ik}(t) = \mathbf{c}_{ik} - (\mathbf{x}_d(t), \mathbf{u}_d(t)) \quad (6.16)$$

La dynamique du i ème élément de la déviation du vecteur d'état ou de l'erreur d'état peut s'écrire :

$$\begin{aligned} \tilde{x}_i(t+1) &= \tilde{f}_i(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) \\ &= \sum_{k=1}^{K_i} q_{ik} \left[\exp\left(-\frac{\|(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) - \tilde{\mathbf{c}}_{ik}(t)\|^2}{\sigma_{ik}^2}\right) - \exp\left(-\frac{\|\tilde{\mathbf{c}}_{ik}(t)\|^2}{\sigma_{ik}^2}\right) \right] \end{aligned} \quad (6.17)$$

pour $i = 1, 2, \dots, n$. On conclut que pour toute valeur $\mathbf{x}_d(t)$ de la trajectoire de référence dans l'espace d'état à un instant t , un modèle non linéaire de l'erreur d'état représenté par une fonction-vecteur :

$$\tilde{\mathbf{x}}(t+1) = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) \quad (6.18)$$

est explicitement exprimé en fonctions des paramètres du réseau GRBF.

A partir de (6.17), il est aussi possible de conclure que le modèle explicite dérivé pour l'erreur d'état est lui-même un réseau GRBF avec des centres $\tilde{\mathbf{c}}_{ik}(t)$, des largeurs σ_{ik} et un biais :

$$\tilde{q}_{i0} = -\sum_{k=1}^{K_i} q_{ik} \exp\left(-\frac{\|\tilde{\mathbf{c}}_{ik}(t)\|^2}{\sigma_{ik}^2}\right) \quad (6.19)$$

6.7.2 Propriétés dynamiques du modèle de l'erreur d'état et reformulation du problème optimal de poursuite de trajectoire dans l'espace d'état

Pour être exploitable dans une stratégie de commande en poursuite de trajectoire dans l'espace d'état avec stabilité garantie, le modèle de l'erreur d'état ou déviation d'état (6.18), dérivé dans la sous-section précédente et exprimé explicitement par les équations (6.17) et (6.19), doit nécessairement satisfaire aux conditions :

1. $\tilde{f}(0,0) = 0$.
2. Le système linéaire $\tilde{\mathbf{x}}_l(t+1) = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_l(t) + \tilde{\mathbf{B}}\tilde{\mathbf{u}}(t)$ obtenu par linéarisation de (6.18) au voisinage de $(0,0) \in \tilde{\mathbf{X}} \times \tilde{\mathbf{U}}$;

$$\tilde{\mathbf{A}} = \left. \frac{\partial \tilde{f}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})}{\partial \tilde{\mathbf{x}}} \right|_{\substack{\tilde{\mathbf{x}}=0 \\ \tilde{\mathbf{u}}=0}} \quad \text{et} \quad \tilde{\mathbf{B}} = \left. \frac{\partial \tilde{f}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}} \right|_{\substack{\tilde{\mathbf{x}}=0 \\ \tilde{\mathbf{u}}=0}}$$

est stabilisable.

La première condition est facilement vérifiée en remplaçant par $(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) = (0,0)$ dans (6.17). Pour la deuxième condition, nous commençons par écrire la déviation d'état, à un instant quelconque t :

$$\begin{aligned} \tilde{\mathbf{x}}(t+1) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) - \mathbf{f}(\mathbf{x}_d(t), \mathbf{u}_d(t)) \\ &= \mathbf{f}(\tilde{\mathbf{x}}(t) + \mathbf{x}_d(t), \tilde{\mathbf{u}}(t) + \mathbf{u}_d(t)) - \mathbf{f}(\mathbf{x}_d(t), \mathbf{u}_d(t)) \\ &= \tilde{f}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) \end{aligned} \quad (6.20)$$

La linéarisation de cette erreur au voisinage de l'origine conduit

à $\tilde{\mathbf{x}}_l(t+1) = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_l(t) + \tilde{\mathbf{B}}\tilde{\mathbf{u}}(t)$ avec, $\tilde{\mathbf{A}} = \left. \frac{\partial \tilde{f}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})}{\partial \tilde{\mathbf{x}}} \right|_{\substack{\tilde{\mathbf{x}}=0 \\ \tilde{\mathbf{u}}=0}}$ et $\tilde{\mathbf{B}} = \left. \frac{\partial \tilde{f}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}} \right|_{\substack{\tilde{\mathbf{x}}=0 \\ \tilde{\mathbf{u}}=0}}$.

En posant :

$$\tilde{f}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) - \mathbf{f}(\mathbf{x}_d(t), \mathbf{u}_d(t)) \quad (6.21)$$

avec $\mathbf{f}(\mathbf{x}_d(t), \mathbf{u}_d(t))$ traité comme un offset (constante), le calcul des dérivées partielles des deux membres de l'équation (6.21) par rapport à $\mathbf{x}(t)$ et $\mathbf{u}(t)$ au point de référence $(\mathbf{x}_d, \mathbf{u}_d)$ résulte en les égalités suivantes :

$$\left. \frac{\partial \tilde{f}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})}{\partial \tilde{\mathbf{x}}} \right|_{\substack{\tilde{\mathbf{x}}=0 \\ \tilde{\mathbf{u}}=0}} = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x}=\mathbf{x}_d \\ \mathbf{u}=\mathbf{u}_d}} \quad (6.22)$$

$$\left. \frac{\partial \tilde{f}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}} \right|_{\substack{\tilde{\mathbf{x}}=0 \\ \tilde{\mathbf{u}}=0}} = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \right|_{\substack{\mathbf{x}=\mathbf{x}_d \\ \mathbf{u}=\mathbf{u}_d}} \quad (6.23)$$

Donc, nous pouvons conclure à partir de l'hypothèse 3 que la stabilisabilité du système linéarisé (\mathbf{A}, \mathbf{B}) au point de référence $(\mathbf{x}_d, \mathbf{u}_d)$ implique la stabilisabilité du modèle linéarisé $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$ de l'erreur d'état à l'origine.

Nous sommes maintenant en mesure de reformuler le problème optimal de commande non linéaire stabilisante en poursuite d'une trajectoire dans l'espace d'état par :

- a) Résoudre le problème optimal à horizon fini avec fonction de coût terminale, défini par l'équation (6.18), les contraintes $\tilde{\mathbf{x}} \in \tilde{\mathbf{X}}$, $\tilde{\mathbf{u}} \in \tilde{\mathbf{U}}$ et en minimisant la fonction coût :

$$J(\tilde{\mathbf{x}}(t), t, \tilde{\mathbf{u}}_{t+N_c+1}, N_c, a, P) = \sum_{i=t}^{t+N_c-1} \{ \tilde{\mathbf{x}}(i)^T \mathbf{Q} \tilde{\mathbf{x}}(i) + \tilde{\mathbf{u}}(i)^T \mathbf{R} \tilde{\mathbf{u}}(i) \} + a \| \tilde{\mathbf{x}}(t+N_c) \|_{\mathbf{P}}^2 \quad (6.24)$$

et déduire la composante prédictive stabilisante $\kappa_{pb}(\tilde{\mathbf{x}}(t)) = \tilde{\mathbf{u}}(t)$ par retour d'état.

- b) Générer une commande directe $\mathbf{u}_d(t)$ correspondant à l'état de référence $\mathbf{x}_d(t)$ pour maintenir le système dans cet état statique au moment où la composante prédictive tend asymptotiquement vers zéro.

Pour le point (a), la composante prédictive de la commande est calculée en résolvant le problème optimal (6.24) par l'application de l'algorithme de Parisini et al. [76-77] pour la régulation prédictive stabilisante à horizon fini HF avec fonction de coût terminale (sous-section 3.2.4). Quant au point (b), Cette commande directe est générée par un réseau GRBF inverse entraîné pour approximer la fonction non linéaire :

$$\mathbf{u}_d(t) = \mathbf{f}^{-1}(\mathbf{x}_d(t), \mathbf{x}_d(t+1)) \quad (6.25)$$

La procédure de l'identification non linéaire inverse par l'algorithme d'apprentissage RANEKF modifié est décrite, en détail, dans la section 5.4 .

6.7.3 Synthèse d'une loi de commande prédictive stabilisante référencée modèle GRBF

L'architecture générale de la loi de commande stabilisante, hybride prédictive neuronique pour la poursuite de trajectoire dans l'espace d'état et, basée sur des modèles GRBF directe et inverse du processus est illustrée dans la figure 6.1.

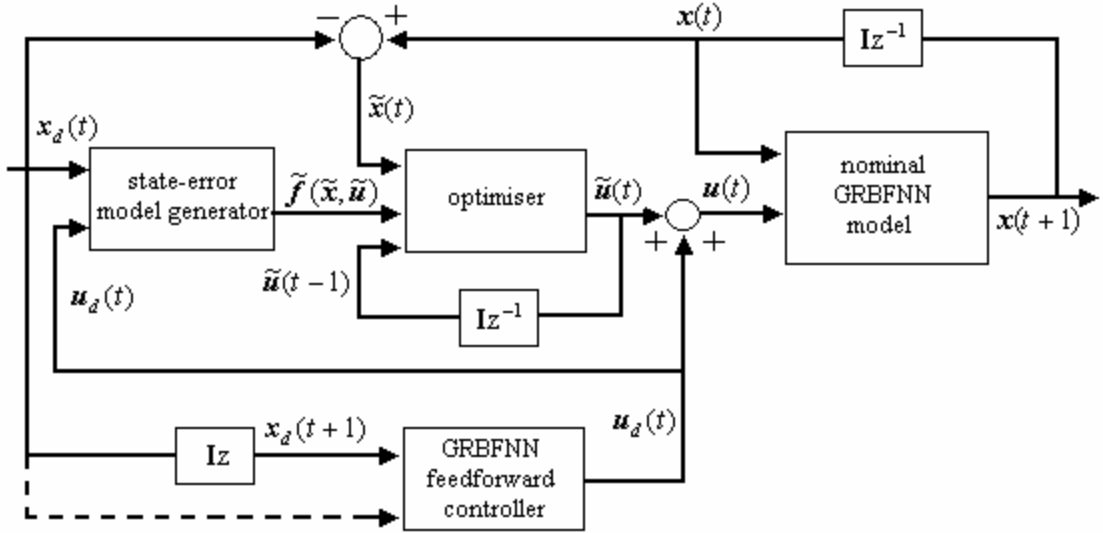


Figure 6.1 : Structure générale d'une commande hybride prédictive-neuronique pour la poursuite de trajectoire dans l'espace d'état.

Dans cette architecture, le block générateur du modèle d'erreur d'état prend les valeurs courantes de l'état de référence $x_d(t)$ et de la composante $u_d(t)$; générée par le contrôleur neuronique ou modèle inverse, pour produire les paramètres du modèle d'erreur d'état, spécifiquement les centres \tilde{c}_{ik} . Ensuite, le modèle d'erreur d'état est utilisé par la fonction d'optimisation pour calculer la composante optimale prédictive et stabilisante $\tilde{u}(t)$ qui assure la stabilité de la commande en boucle fermée.

La minimisation de la fonction coût $J(\tilde{x}(t), \tilde{u}_{t+N_c-1}, N_c, a, \mathbf{P})$ à chaque instant d'échantillonnage t , implique l'utilisation des prédictions de l'erreur d'état le long de l'horizon de prédiction N_c données par la séquence :

$$\tilde{x}_{t,t+N_c} = \{\tilde{x}(t/t), \tilde{x}(t+1/t), \dots, \tilde{x}(t+N_c/t)\} \quad (6.26)$$

pour produire la séquence des commandes optimales :

$$\tilde{u}_{t,t+N_c-1} = \{\tilde{u}(t/t), \tilde{u}(t+1/t), \dots, \tilde{u}(t+N_c-1/t)\} \quad (6.27)$$

Le vecteur $\tilde{x}(t/t)$ représente l'état courant de la déviation d'état et, est donné par :

$$\tilde{x}(t/t) = x(t) - x_d(t) \quad (6.28)$$

et la composante prédictive stabilisante par retour d'état n'est autre que le premier élément de la séquence $\tilde{\mathbf{u}}_{t, t+N_c-1}$, et on écrit :

$$\begin{aligned}\kappa_{fb}(\tilde{\mathbf{x}}(t)) &= \tilde{\mathbf{u}}(t/t) \\ &= \tilde{\mathbf{u}}(t)\end{aligned}\quad (6.29)$$

Les éléments de la séquence (6.26) sont calculés en utilisant le modèle (6.17) de l'erreur d'état comme prédicteur récursif à d -étapes-futures (d -step-ahead recursive predictor) :

$$\tilde{x}_i(t+d) = \sum_{k=1}^{K_i} q_{ik} \left\{ \exp\left(-\frac{\|(\tilde{\mathbf{x}}(t+d-1|t), \tilde{\mathbf{u}}(t+d-1|t)) - \tilde{\mathbf{c}}_{ik}(t+d-1|t)\|^2}{\sigma_{ik}^2}\right) - \exp\left(-\frac{\|\tilde{\mathbf{c}}_{ik}(t+d-1|t)\|^2}{\sigma_{ik}^2}\right) \right\} \quad (6.30)$$

Pour $d=1, 2, \dots, N_c$ et $i=1, 2, \dots, n$. Les centres des gaussiennes $\tilde{\mathbf{c}}_{ik}(t+d-1|t)$ qui apparaissent dans les prédictions sont donnés en transformant (6.16) :

$$\tilde{\mathbf{c}}_{ik}(t+d-1|t) = \mathbf{c}_{ik} - (\mathbf{x}_d(t+d-1|t), \mathbf{u}_d(t+d-1|t)) \quad (6.31)$$

L'équation (6.31) montre clairement que les centres futures dépendent, à la fois, des états futures de la trajectoire de référence et des entrées de référence futures générées par le contrôleur neuronique ou modèle inverse du processus. Notons aussi que, les équations (6.30) et (6.31) montrent que l'information sur l'état courant et les états futures de la trajectoire de référence est entièrement intégrée dans les prédicteurs de l'erreur d'état à travers les déviations opérées sur les centres des gaussiennes dans l'espace d'entrée.

Contrairement aux structures classiques de la commande prédictive référencée modèle où, le modèle du processus est directement utilisé pour le calcul d'un profil de commande optimal basé sur des prédictions de l'état le long d'un horizon de prédiction, dans cette nouvelle architecture le modèle nominal du processus est utilisé pour générer un autre modèle qui décrit explicitement la dynamique de l'erreur sur l'état pour une valeur spécifique de l'état de référence. C'est précisément ce modèle de l'erreur de poursuite (tracking error model) qui est utilisé pour prédire le comportement future de l'erreur de poursuite et minimiser la fonction de coût pour, enfin, produire une action de commande qui entraîne l'erreur sur l'état vers zéro ($\lim_{t \rightarrow \infty} |\mathbf{x}(t) - \mathbf{x}_d(t)| = 0$) et assure, ainsi, la stabilité de la commande.

L'algorithme de cette commande hybride prédictive-neuronique pour la stabilisation en poursuite de trajectoire dans l'espace d'état peut être résumé ainsi :

Etape 0

- 0.1 Charger le modèle nominal du processus et le modèle inverse
- 0.2 Fixer la trajectoire d'état de référence
- 0.3 Fixer N_c , \mathbf{x}_o
- 0.4 Fixer la séquence initiale de la composante prédictive à zéro

Etape 1

- 1.1 Obtenir la séquence des valeurs futures de l'état de référence

$$\mathbf{x}_{d/t,t+N_c} = \{\mathbf{x}_d(t/t), \mathbf{x}_d(t+1/t), \dots, \mathbf{x}_d(t+N_c/t)\}$$

- 1.2 Utiliser le modèle inverse (6.25) pour calculer la séquence des commandes de référence futures

$$\mathbf{u}_{d/t,t+N_c} = \{\mathbf{u}_d(t/t), \mathbf{u}_d(t+1/t), \dots, \mathbf{u}_d(t+N_c-1/t)\}$$

- 1.3 Fixer $d=1$, utiliser l'équation (6.31) pour calculer les centres $\tilde{\mathbf{c}}_{ik}(t/t)$, et générer le modèle prédictif à *une*-étape-future de l'erreur d'état, telle que donnée par (6.30).

Etape 2

- 2.1 Linéariser le modèle de l'erreur d'état au voisinage de l'origine.
- 2.2 Appliquer l'algorithme de Parisini et al. [21] et déterminer la matrice P et le scalaire $\tilde{\alpha}$ pour la fonction coût (6.24).

Etape 3

- 3.1 Calculer la déviation d'état courante

$$\tilde{\mathbf{x}}(t/t) = \mathbf{x}(t) - \mathbf{x}_d(t).$$

- 3.2 Utiliser l'équation (6.31) pour le calcul de la séquence

$$\{\tilde{\mathbf{c}}_{ik}(t+1/t), \tilde{\mathbf{c}}_{ik}(t+2/t), \dots, \tilde{\mathbf{c}}_{ik}(t+N_c-1/t)\},$$

ensuite utiliser (6.30) pour calculer les prédictions sur l'erreur d'état $\tilde{\mathbf{x}}(t+j/t)$

pour $1 \leq j \leq N_c$.

Etape 4

Formuler la fonction coût (6.24).

Etape 5

- 5.1 Utiliser l'algorithme d'optimisation pour obtenir la séquence de

commande $\{\tilde{\mathbf{u}}(t/t), \tilde{\mathbf{u}}(t+1/t), \dots, \tilde{\mathbf{u}}(t+N_c-1/t)\}$.

5.2 Appliquer la commande hybride $\mathbf{u}(t) = \tilde{\mathbf{u}}(t/t) + \mathbf{u}_d(t)$ au système nominal, et mesurer l'état courant $\mathbf{x}(t)$.

Etape 6

6.1 Fixer la future séquence de commande initiale à

$\{\tilde{\mathbf{u}}(t/t), \tilde{\mathbf{u}}(t+1/t), \dots, \tilde{\mathbf{u}}(t+N_c-1/t)\}$.

6.2 Incrémenter $t=t+1$ et aller à l'étape 1.

6.8 Simulations et résultats

Dans cette section nous tâcherons d'illustrer comment notre approche de commande hybride prédictive-neuronique stabilise un système non linéaire en poursuite de trajectoire dans l'espace d'état.

Le système nominal considéré dans cette simulation est le modèle GRBF du manipulateur à articulation unique déjà identifié et testé dans la sous-section 5.3.3 du chapitre précédent. Nous rappelons que ce système est décrit par un vecteur d'état $\mathbf{x}(t) = [x_1(t), x_2(t)]^T$ où, $x_1(t)$ et $x_2(t)$ sont la position et la vitesse angulaires, respectivement. La variable d'état est $u(t) \in \mathbb{R}^1$. Les contraintes sur l'état et la commande sont exprimées par :

$$\mathbf{X} = \{ \mathbf{x} = [x_1, x_2]^T ; |x_1| \leq 0.7rd, |x_2| \leq 2.5rd/s \} \quad (6.32)$$

$$\mathbf{U} = \{ \mathbf{u} ; |u| \leq 7Nm \} \quad (6.33)$$

Les valeurs des matrices **A** et **B** données dans (5.59), (5.60), (5.61) et le tableau (5.1) montrent les propriétés de stabilité du modèle nominal et, qu'il satisfait à l'hypothèse (3) formulée dans la section 6.2. Le modèle inverse ou contrôleur neuronique est celui identifié dans la section 5.4 selon la technique de l'inversion directe. Les performances de ce modèle sont clairement montrées dans les figures (5.21) et (5.22). Pour démontrer la stabilité asymptotique en boucle fermée de la commande hybride prédictive-neuronique, plusieurs exercices de simulation sont mis en œuvre en utilisant la fonction *fmincon* disponible sur le 'MATLAB Optimisation Toolbox' pour l'optimisation sous contrainte.

A partir de (6.6), la déviation de la variable de commande $\tilde{u}(t)$ est contrainte de varier à l'intérieur de l'ensemble $\tilde{\mathbf{U}} = \{\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}_d : |\tilde{\mathbf{u}}| \leq 14Nm\}$. Dans la première simulation, l'objectif fixé est d'utiliser le modèle GRBF dans une régulation stabilisante prédictive classique, c'est-à-dire ramener le système vers l'origine en un temps fini à partir de certains état initiaux. Dans ce cas, $x_d(t) = u_d(t) = 0$ et le modèle est directement utilisé pour fournir les prédictions nécessaire au calcul du signal de commande. Les matrices \mathbf{Q} et \mathbf{R} dans la fonction coût (6.24) sont fixées par $\mathbf{Q} = \alpha_1 \mathbf{I}$, $\mathbf{R} = \alpha_2 \mathbf{I}$ avec \mathbf{I} la matrice unité et, avec $\alpha_1 = 0.001$ et $\alpha_2 = 0.5$. L'horizon de commande N_c est fixé à 5 et, la matrice \mathbf{P} et le scalaire \tilde{a} sont calculés selon la procédure décrite par Parasini et al. dans [77], ce qui a donné :

$$\mathbf{P} = 10^3 \begin{bmatrix} 141.98 & 3.9460 \\ 3.9460 & 9.6054 \end{bmatrix} \quad \text{et} \quad \tilde{a} = 0.008982$$

Les trajectoires d'état en boucle fermée et les profils de commande pour différents états initiaux sont montrées dans les figures (6.2) et (6.3).

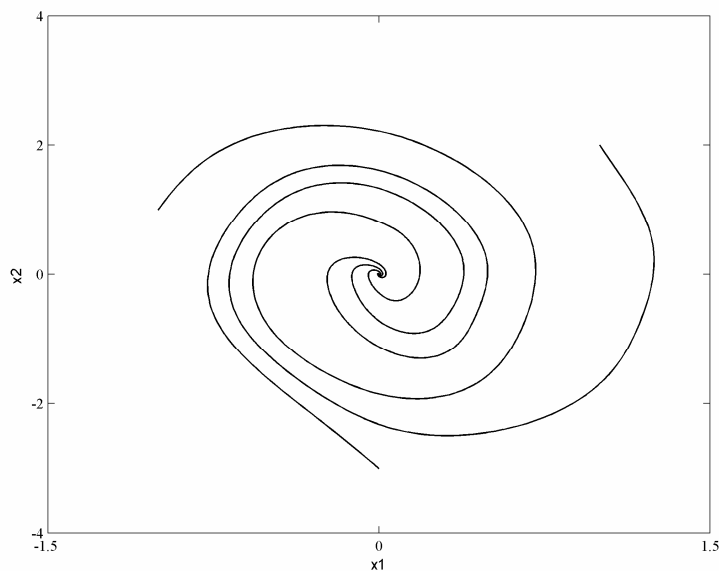


Figure 6.2 : trajectoires dans l'espace d'état en boucle fermée

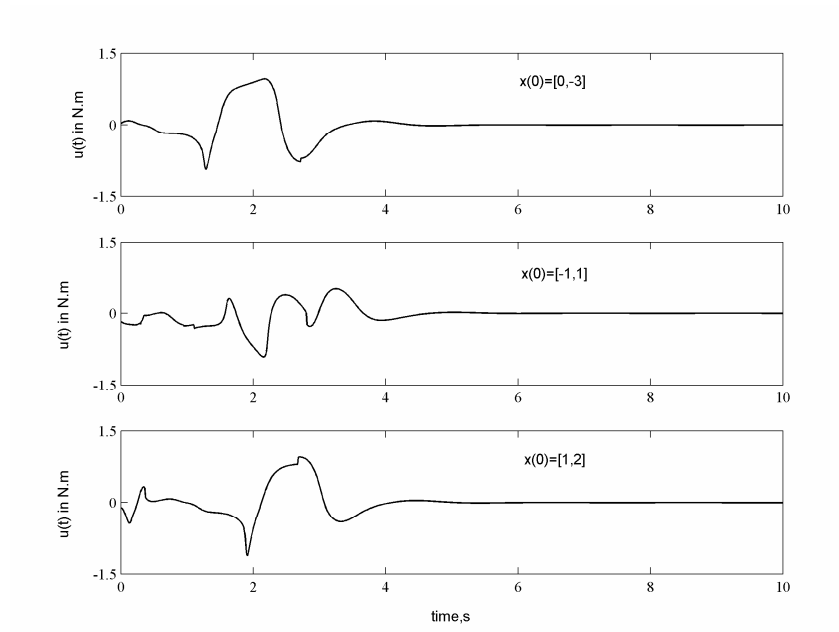


Figure 6.3 : profils de commande en régulation stabilisante pour différents états initiaux

La deuxième simulation illustre les performances de poursuite de trajectoires d'état fixes et variables. Dans ce cas les matrices $\mathbf{Q} = \alpha_1 \mathbf{I}$, $\mathbf{R} = \alpha_2 \mathbf{I}$ sont fixées en choisissant $\alpha_1 = 0.001$ et $\alpha_2 = 0.025$. Les réponses du système pour plusieurs points de référence sont illustrées dans la figure (6.4).

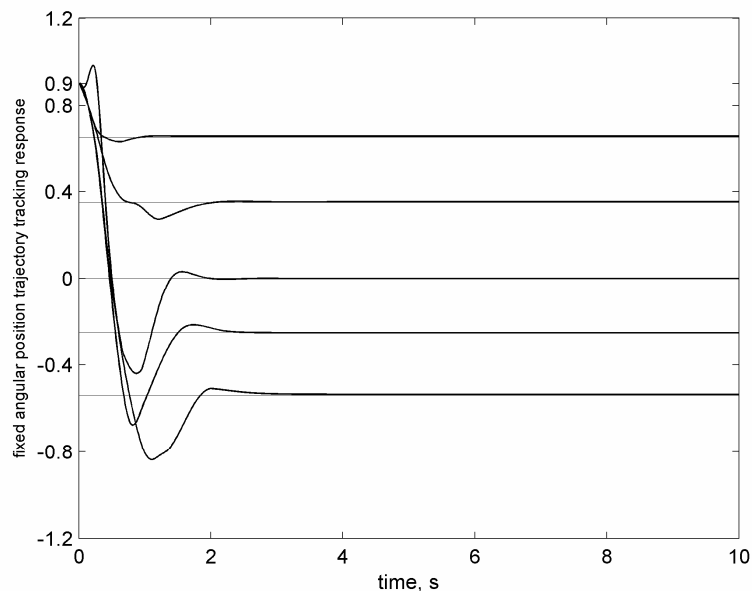


Figure 6.4 : Réponse du système pour plusieurs consignes

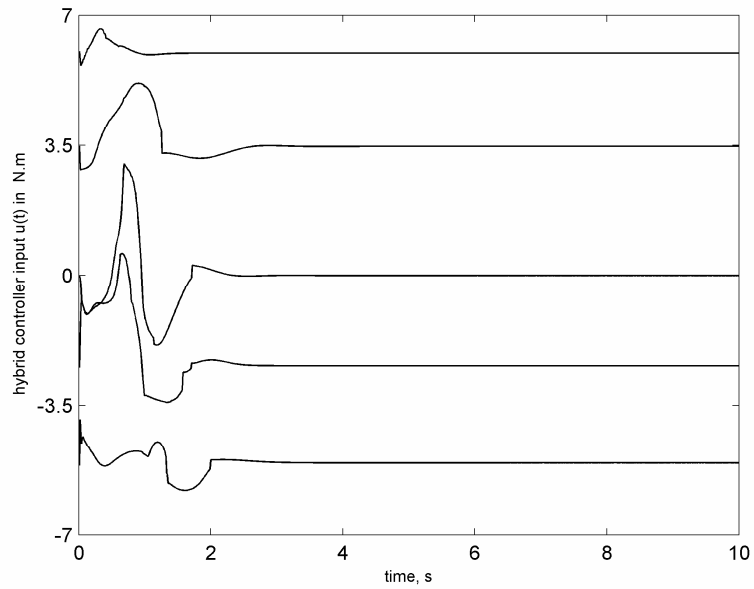


Figure 6.5 : Profils des signaux de commande

Point de réf. $\mathbf{x}_d = [x_1, x_2]^T$	Matrice P	Scalaire \tilde{a}
$[0.65, 0]^T$	$10^7 \begin{bmatrix} 1.4671 & 0.0561 \\ 0.0561 & 0.1554 \end{bmatrix}$	5.4414×10^{-6}
$[0.35, 0]^T$	$10^7 \begin{bmatrix} 2.0465 & 0.0909 \\ 0.0909 & 0.0988 \end{bmatrix}$	9.2380×10^{-6}
$[0, 0]^T$	$10^9 \begin{bmatrix} 1.8984 & -0.0334 \\ -0.0334 & 0.2407 \end{bmatrix}$	4.2685×10^{-8}
$[-0.25, 0]^T$	$10^6 \begin{bmatrix} 5.1442 & 0.0083 \\ 0.0083 & 1.7891 \end{bmatrix}$	1.3407×10^{-5}
$[-0.55, 0]^T$	$10^7 \begin{bmatrix} 1.6600 & -0.0710 \\ -0.0710 & 0.2404 \end{bmatrix}$	2.1231×10^{-6}

Tableau 6.1: Valeurs de la matrice **P** et du scalaire \tilde{a} pour différents Points de référence

L'état initial est sélectionné hors du domaine d'opération pour mieux démontrer l'action stabilisante du contrôleur. L'horizon de commande N_c est fixé à 9 et, les valeurs calculées pour la matrice \mathbf{P} et le scalaire \tilde{a} sont présentées dans le Tableau (6.1) pour les différentes valeurs de l'état de référence. Les profils de la commande hybride stabilisante sont montrés dans la figure (6.5).

Le cas d'une trajectoire de référence variable est illustré par la figure (6.6) pour les performances de poursuite. Les profils des composantes neuronique et prédictive (gras) sont montrés dans la figure (6.7). L'horizon de commande N_c pour cette dernière simulation est fixé à 9 et, les valeurs de la matrice \mathbf{P} et du scalaire \tilde{a} sont calculées hors-ligne (off-line) en exploitant la connaissance *a priori* de la trajectoire de référence et en utilisant la même procédure de Parisini et al. [77].

Ces résultats montrent les bonnes performances de stabilité tant en mode régulation qu'en mode poursuite de trajectoire dans l'espace d'état de cette nouvelle structure de commande hybride prédictive-neuronique où, le problème de

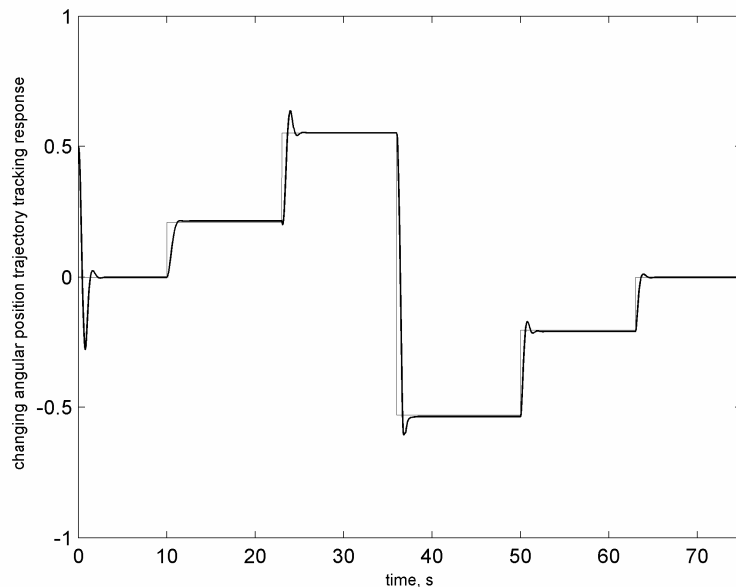


Figure 6.6 : performance d'une commande en poursuite

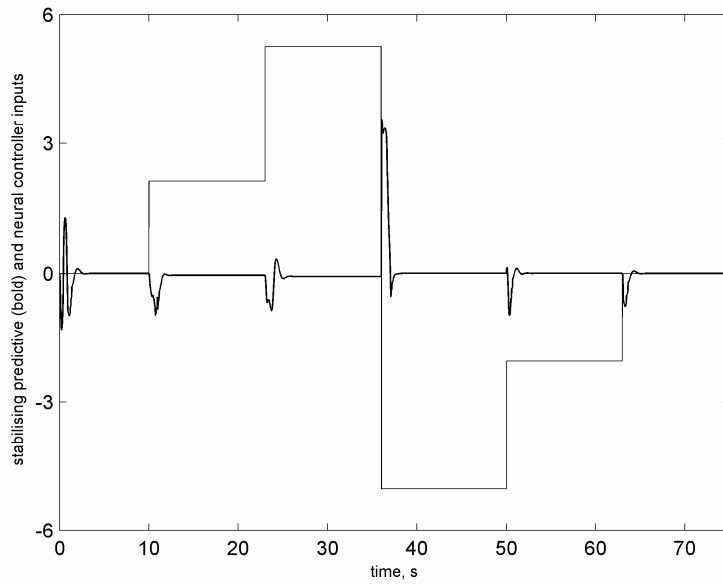


Figure 6.7 : Composantes prédictive (gras) et neuronique du contrôleur

poursuite est transformé en un problème de régulation permettant, ainsi, de bénéficier des propriétés de stabilisation qu'assurent les algorithmes de régulation prédictive non linéaire disponibles.

Conclusion

Dans ce travail, une approche pratique de l'identification des systèmes non linéaires par les réseaux de neurones à fonctions Gaussiennes à base radiale est avancée.

Un algorithme, le RANEKF, traditionnellement utilisé en apprentissage séquentiel, est modifié pour être exploité dans des applications d'identification hors-ligne. Ces modifications comportent :

- (3) Une permutation aléatoire des données de l'identification.
- (4) L'introduction de l'erreur relative parmi les critères de '*nouveauté*' de l'algorithme. Cette nouvelle condition assure une meilleure approximation sur tout l'espace d'opération.

Les performances de l'algorithme RANEKF modifié sont testées par des simulations d'identification de modèles NARX. Les résultats de ces simulations montrent la rapidité de l'apprentissage, la précision des modèles identifiés et leurs bonnes aptitudes à l'interpolation avec des topologies réduites.

Dans une deuxième phase et, afin de bénéficier directement de la théorie et des algorithmes disponibles pour la stabilisation de la commande prédictive non linéaire, une structure de réseaux de neurones GRBF est proposée pour la description, dans l'espace d'état, d'un système dynamique non linéaire discret. Cette structure parallèle et récursive est composée de prédicteurs d'état identifiés par des réseaux GRBF entraînés en utilisant l'algorithme RANEKF modifié. Les travaux de simulation ont montré que cette classe de modèles neuroniques possède, non seulement, des aptitudes d'approximation mais aussi des capacités d'apprentissage et de restitution des propriétés dynamiques et de stabilité du système identifié. Les caractéristiques dynamiques de cette structure de réseaux GRBF ouvrent le chemin de son application dans des schémas de commande prédictive nonlinéaire assurant la stabilité asymptotique en boucle fermée.

Une contribution majeure de ce travail concerne la synthèse d'une architecture de commande en poursuite à caractère hybride : prédictive et neuronique. Cette architecture est basée sur une propriété analytique propre à la structure GRBF décrite dans le chapitre 5 et, qui rend possible l'extraction d'un modèle dynamique explicite de la déviation des états pour une trajectoire donnée. Le résultat est une transformation du problème de poursuite optimale dans l'espace d'état en, un simple problème de régulation de la déviation vers l'origine. La résolution du problème de régulation dans le cadre d'une commande prédictive stabilisante garantit la stabilité asymptotique de la loi de commande en poursuite. Le maintien du système dans un état stationnaire désiré est, quant à lui, assuré par un contrôleur neuronique modélisant la dynamique inverse du système sous commande. Les exercices de simulation effectués ont montré les bonnes performances de stabilité et de poursuite de la commande proposée.

Ce travail soulève, néanmoins, certaines questions théoriques et pratiques et qui peuvent constituer une base pour des travaux de recherche futurs :

- L'applicabilité de la structure GRBF proposée dépend énormément de son aptitude, non seulement, à l'approximation d'une fonctionnelle non linéaire mais aussi à pouvoir restituer ses dérivées partielles de premier ordre par rapport aux variables d'état et de commande. Cette exigence inspire d'approfondir d'avantage l'analyse théorique de l'approximation par les réseaux RBF en l'étendant vers les dérivées partielles de premier ordre.
- Dans ce travail, la question de la stabilité n'était abordée que pour le cas nominal. Quelles sont, alors, les conditions de stabilité de la structure de commande proposée, basée sur des réseaux de neurones, en présence des erreurs de modélisation et/ou de perturbations externes ?

Bibliographie

- [1] Roberts P. D., A brief overview of model predictive control, *Computing and control engineering journal*, Vol. 10, (5), Oct. 1999, pp. 186-188
- [2] Qin J., and Badgwell T. A., An overview of industrial model predictive control technology, *Proceedings of Chemical Process control, AICHE Symposium Series 316*, California, 1997, pp. 232-256
- [3] Garcia C.E., Prett D. M., and Morari M., Model Predictive Control: Theory and Practice - a Survey, *Automatica*, Vol. 25, (3), 1989, pp. 335-348
- [4] Richalet J., Rault A., Testud J.L., and Papon J., Model predictive heuristic control: Application to industrial processes, *Automatica*, Vol. 14, 1978, pp. 413-428
- [5] Cutler C. R., and Ramaker B.L., Dynamic matrix control – a computer control algorithm, *Proceedings of the Joint Automatic Control Conference*, San Francisco, 1980.
- [6] Rawlings J.B., Tutorial : Model predictive control technology, *Proceedings of the American Control Conference*, San Diego, California, June 99, pp. 662-676
- [7] De Keyser R.M.C., and Van Cuawenberghe A.R., Extended prediction self-adaptive control, *Proceedings of IFAC Symposium on Identification and System Parameter Estimation*, York, UK, July 1985, pp. 1317-1322
- [8] Clarke D. W. , Mohtadi C. and Tuffs P.S. : Generalised predictive control – part I. The basic algorithm, *Automatica* , Vol. 23, (2) , 1987 , pp.137-148
- [9] Clarke D. W. , Mohtadi C. and Tuffs P.S. : Generalised predictive control – part II. Extensions and interpretations, *Automatica* , Vol. 23 , (2) , 1987 , pp.149-160
- [10] Åström K.J., Introduction to stochastic control theory, Academic Press, New York, 1970
- [11] Grimble M.J., LQG predictive optimal control for adaptive applications, *Automatica*, Vol. 26, (3) , 1990, pp. 949-961
- [12] Grimble M.J., Two degree of freedom linear quadratic Gaussian predictive control , *Proceedings of the IEE, Part D, Control Theory and Applications*, 142, July 1995, pp. 295-306
- [13] Ahmida Z., and Charef A., Analysis of disturbance effects on the transient response performance of an LQGPC controller, *Proceedings of the International Annual Conference IEEA'97*, Vol. 2 (Control Engineering), Batna, Algeria, Dec. 1997, pp. 131-135

- [14] Bitmead R.R., Gevers M., and Wertz V., Adaptive optimal control – The thinking man's GPC, Englewood Cliffs, Prentice-Hall, NJ, 1990
- [15] Mosca E., Lemos J.M. and Zhang J. , Stabilizing I/O receding horizon control, *Proceedings 29th IEEE Conference on Decision and Control*, Honolulu, 1990, pp. 2518-2523
- [16] Clarke D. W. , and Scattolini R., Constrained receding horizon predictive control, *Proceedings of the IEE, Part D, Control Theory and Applications*, 138, 1991, pp. 347-354
- [17] Mosca E., and Zhang J., Stable redesign of predictive control, *Automatica*, Vol. 28, (6), 1992, pp. 1229-1233
- [18] Navratil J.P., Lim K.Y., and Fisher D.G., Disturbance feedback in model predictive control systems, *Proceedings of the IFAC workshop on model-based process control*, Atlanta, 1988, pp. 63-68
- [19] Li S., Lim K.Y., and Fisher D.G., A state space formulation for model predictive control, *AIChE Journal*, Vol.35, (2), 1989, pp. 241-249
- [20] Lee J.H., Morari M., and Garcia C.E., State space interpretation of model predictive control, *Automatica*, Vol. 30, (4), 1994, pp. 707-717
- [21] Muske K.R., and Rawlings J.R., Model predictive control with linear models, *AIChE Journal*, Vol.39, (2), 1993, pp. 262-287
- [22] Becerra V.M., Abu-el-zeet Z.H., and Roberts P.D., Integrating predictive control and economic optimisation, *Computing and control engineering journal*, Vol. 10, (5), Oct. 1999, pp. 198-208
- [23] Becerra V.M., Roberts P.D., and Griffiths G.W., Novel developments in process process optimisation using predictive control, *Journal of process Control*, Vol. 8 ,(2), 1998, pp.117-138
- [24] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Sokaert, P. O. M., Constrained model predictive control: Stability and optimality, *Automatica*, Vol. 36, (6), 2000, pp. 789-814
- [25] Van Overchee P. et de Moor B., Sub-space algorithms for the identification of combined deterministic-stochastic systems, *Automatica*, Vol. 30, 1994, pp.75-93
- [26] Qin J., and Badgwell T. A., An overview of nonlinear model predictive control applications, *Proceedings of IFAC workshop: nonlinear model predictive control – Assessment and future directions*, Ascona, Switzerland, June 98

- [27] Narendra, K., S., and Parthasarathy, K., Identification and control of dynamical using neural networks, *IEEE Transactions on neural Networks*, Vol. 1, (1), 1990, pp.4-27
- [28] Hunt K. G. , Sbarbaro D., Żbikowski R. and Gawthrop P. G., Neural networks for control systems—A survey' , *Automatica* , Vol. 28 , (6), 1992, pp.1083-1112
- [29] Piché S., Rodsari B. S., Johnson D., and Gerules M., Nonlinear model predictive control using neural networks, *IEEE Control Systems Magazine*, June 2000, pp. 53-62
- [30] Byun D.G., and Kwon W.H., Predictive control: A review and some new stability results, *Proceedings of IFAC workshop on model based process control*, Pergamon Press, Oxford 88, pp. 81-87
- [31] Scattolini R, and Bittanti S., On the choice of the horizon in long range predictive control—Some simple criteria, *Automatica*, Vol. 26, (5), 1995, pp. 915-922
- [32] Kwon W.H., Bruckstein A.M., and Kailath T., Stabilising state feedback design via the moving horizon method, *International Journal of Control*, Vol. 37, (3), 1983, pp. 631-643
- [33] Scokaert P.O.M., and Clarke D.W., Stabilising properties of constrained predictive control, *Proceedings of the IEE, Part D, Control Theory and Applications*, 141, 1994, pp. 295-304
- [34] Eaton J. W., Rawlings J. B., Ungar L. H., Stability of neural net based model predictive control, *Proc.American Control Conference*, Maryland, June 1994, pp. 2481-2885
- [35] De Nicolao, G., Magni, L., and Scattolini, R., Stabilizing predictive control of nonlinear ARX models, *Automatica*, Vol. 33, (9), 1997, pp. 1691-1697
- [36] Luther J. B., Sorensen P. H., Stability of a neural predictive controller scheme on a neural model, *International Joint Conference on Neural Networks*, IJCNN99, Vol. 3, July 99, pp. 2087-2091
- [37] Nagy E., State tracking through optimised trajectory tracking, *Proceedings of the American Control Conference*, San Diego, California, June 99, pp. 1807-1811
- [38] Magni L., De Nicolao G., and Scattolini R., Output feedback and tracking of nonlinear systems with model predictive control, *Automatica*, Vol. 37, 2001, pp. 1601-1607
- [39] Magni L., On robust tracking with nonlinear model predictive control, *International journal of control*, Vol. 75, 2002, pp.399-407

- [40] De Nicolao, G., Magni, L., and Scattolini, R., On the robustness of receding horizon control with terminal constraints, *IEEE Transactions on Automatic Control*, Vol. 41, (3), 1996, pp. 451-453
- [41] Michalska, H., and Mayne, D. Q., Robust receding horizon control of constrained nonlinear systems, *IEEE Transactions on Automatic Control*, Vol. 38, 1993, pp. 1623-1633
- [42] Mayne, D. Q., and Langson W., Robustifying model predictive control of constrained linear systems, *Electronics letters*, Vol. 37, (23), Nov. 2001, pp. 1422-1423
- [43] Canon, M.D., Cullum C.D., Polak E., Theory of optimal control and mathematical programming, McGraw-Hill, New York, 1970
- [44] Coleman, T.F. and Y. Li, A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on some of the Variables, *SIAM Journal on Optimization*, Vol. 6, (4), 1996, pp 1040-1058
- [45] Gill, P.E., W. Murray, and M.H. Wright, Numerical Linear Algebra and Optimization, Vol. 1, Addison Wesley, 1991
- [46] Gill, P.E., W. Murray, and M.H. Wright, Practical Optimization, London, Academic Press, 1981
- [47] Becerra V.M., Roberts P.D., Dynamic integrated system optimisation and parameter estimation for discrete time optimal control of nonlinear systems, *International Journal of Control*, Vol. 63, 1996, pp.257-281
- [48] Biegler L.T., Advances in nonlinear programming concepts for process control, *Journal of Process Control*, Vol. 8, (5-6), 1998, pp. 301-319
- [49] Ahmida Z. and Charef A., Nonlinear systems modelling using RBF Neural networks : A random learning approach to the resource allocating network algorithm, *Proceedings of the 10th Mediterranean Conference on Control and Automation*, MED2002, Lisbon, Portugal, July 9-12, 2002
- [50] Ahmida, Z., Charef, A., and Becerra, V. M., Stabilising solution to a class of nonlinear optimal state tracking problem using radial basis function networks, *International Journal of Applied Mathematics and Computer Science*, Vol. 15, (3), 2005, pp.369-381
- [51] Vidyasagar M., Nonlinear systems analysis, Prentice-Hall, Englewood Cliffs, NJ, 1993
- [52] Kuo B.C., Digital control systems, Holt-Saunders International Editions, Tokyo, 1981

- [53] Canon, M.D., Cullum C.D., Polak E., Theory of optimal control and mathematical programming, McGraw-Hill, New York, 1970
- [54] Dreyfus S.E., and Law A.M., The art and theory of dynamic programming, Academic Press, New York, 1977
- [55] Bertsekas D.P., Dynamic programming, Prentice-Hall, Englewood Cliffs, NJ, 1987
- [56] Kwakernaak H., and Sivan R., Linear optimal control systems, John Wiley and Sons, New York, 1972
- [57] Chan S. W., Goodwin G. C., and Sin K. S., Convergence properties of the Riccati Difference equation in optimal filtering of nonstabilizable systems, *IEEE Transactions on Automatic Control*, Vol. 29, (2), 1984, pp. 110-118
- [58] Poubelle M. A., Petersen I. R., Gevers M. R., and Bitmead R. R., A Miscellany of results on an equation of Count J. F. Riccati , *IEEE Transactions on Automatic Control*, Vol. 31, (7), 1986, pp. 651-654
- [59] De Souza C. E., Gevers M. R., and Goodwin G. C., Riccati equation in optimal filtering of nonstabilizable systems having singular state transition matrices, *IEEE Transactions on Automatic Control*, Vol. 31, 1986, pp 831-838
- [60] De Souza C. E., On stabilizing properties of solutions of the Riccati difference equation, *IEEE Transactions on Automatic Control*, Vol. 34, (12), 1989, pp 1313-1316
- [61] Rawlings J. B. and Muske K. R., Stability of constrained receding horizon control, *IEEE Transactions on Automatic Control*, Vol. 38, 1993, pp. 1512-1516
- [62] Meadows E.S., Muske K.R. and Rawlings J.B., Implementable model predictive control in the state space, *Proceedings of the American Control Conference*, Seattle, Washington, June 1995, pp. 3699-3703
- [63] Scokaert P.O. M. and Rawlings J.B., Constrained linear quadratic regulation, *IEEE Transactions on Automatic Control*, Vol. 43, (8), August 1998, pp. 1163-1169
- [64] Sznajder M. and Damberg M.J., Suboptimal control of linear systems with state and control inequality constraints, *In Proceedings of the 26th Conference on Decision and Control*, Los Angeles, 1987, pp 761-762
- [65] Laub, A. J. Numerical linear algebra aspects of control design computations, *IEEE Transactions on Automatic Control*, Vol. 30, (2), 1985, pp 97-108
- [66] Muske K.R., Linear model predictive control of chemical processes, PhD thesis, University of Texas at Austin, May 1995

- [67] Rawlings J. B., Tutorial overview of model predictive control, *IEEE control systems magazine*, June 2000, pp. 38-52
- [68] Keerthi S. S. and Gilbert E. G., Optimal, infinite horizon feedback laws for a general class of constrained discrete-time systems : Stability and moving horizon approximations, *J. Optim. Theory Appl.*, Vol. 57, 1988, pp. 265-293
- [69] Keerthi S. S. and Gilbert E. G., An existence theorem for discrete-time infinite horizon optimal control problems, *IEEE Transactions on Automatic Control*, Vol. 30, (9), September 1985, pp.907-909
- [70] Rawlings J. B., Meadows E. S., and Muske K. R., Nonlinear model predictive control: A tutorial and survey, *ADCHEM'94 Proceedings*, Kyoto, Japan, 1994, pp. 185-197
- [71] Meadows E. S., Henson M. A., Eaton J. W., and Rawlings J. B., Receding horizon control and discontinuous state feedback stabilization, *International Journal of Control*, Vol. 62, (5), 1995, pp. 1217-1229
- [72] Chen, C. C., and Shaw, L., On receding horizon feedback control. *Automatica*, Vol. 18, 1982, 349-352
- [73] Mayne, D. Q., and Michalska, H., Receding horizon control of nonlinear systems, *IEEE Transactions on Automatic Control*, Vol. 35, 1990, pp. 814-824
- [74] Scokaert P. O. M., Mayne D. Q., and Rawlings J. B., Suboptimal model predictive control : Feasibility implies stability, *IEEE transactions on automatic control*, Vol. 44, (3), 1999, pp. 648-654
- [75] De Nicolao, G., Magni, L., and Scattolini, R., Stabilizing receding-horizon control of nonlinear time-varying systems, *IEEE Transactions on Automatic Control*, Vol. 43, (7), 1998, pp. 1030-1036
- [76] Parisini, T., and Zoppoli, R., A receding-horizon regulator for nonlinear systems and a neural approximation, *Automatica*, Vol. 31, (10), 1995, pp. 1443-1451
- [77] Parisini, T., Sanguinetti, M., and Zoppoli, R., Nonlinear stabilization by receding-horizon neural regulators, *International Journal of Control*, Vol. 70, (3), 1998, pp. 341-362
- [78] McCulloch W.S., and Pitts W., A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, Vol. 9, 1943, pp.127-147
- [79] Rosenblatt F., The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological review*, (65), 1958, pp.386-408

- [80] Rumelhart D.E., and McClelland J.L., *Parallel Distributed Processing: Exploration in the microstructures of cognition*, MIT Press, Cambridge, 1986
- [81] Anderson J.A., and Rosenfeld E., *Neurocomputing: Foundations of research*, MIT Press, Cambridge, 1988
- [82] Grossberg S., *Neural networks and natural intelligence*, MIT Press, Cambridge, 1988
- [83] Albus J.S., A new approach to manipulator control: The Cerebellar Model Controller (CMAC), *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, (97), 1975, pp.220-227
- [84] Webros, P.J., *Beyond Regression: New tools for prediction and analysis in the behavioural sciences*, PhD Thesis, Applied mathematics, Harvard University, 1974
- [85] Kohonen T., *Self-Organization and Associative Memory*, Springer-Verlag, New York, 1988
- [86] Hopfield J.J., Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences of the U.S.A*, (79), 1982, pp. 2554-2558
- [87] Pozine N., *Simulation des Structures Neuroniques*, Traduction Française Editions MIR, Moscou, 1974
- [88] Hornik K., Stinchcombe M., and White H., Multilayer feedforward networks are universal approximators, *Neural Networks*, Vol. 2, 1989, pp. 359-366
- [89] Chen S., Billings S.A, and Grant P.M., Nonlinear systems identification using neural networks, *International Journal of Control*, Vol. 51, 1990, pp. 1191-1214
- [90] Chen S., Cowan C.F.N., Billings S.A, and Grant P.M., Parallel recursive prediction error algorithm for training layered neural networks, *International Journal of Control*, Vol. 51, 1990, pp. 1215-1228
- [91] Chen S., Billings S.A, Neural networks for nonlinear system modelling and identification , *International Journal of Control*, Vol. 56, 1992, pp. 319-346
- [92] Haykin S., *Neural networks: A Comprehensive Foundation*, Macmillan College Publishing Company, New York, 1994
- [93] Minsky M. and Papert S.A., *Perceptrons*, The MIT Press, Cambridge 1988

- [94] Warwick, K., Irwin G.R., and Hunt K.J, Neural networks for control and systems: Principles and applications, Control Engineering Series, Peter Peregrinus, 1992
- [95] Broomhead D. S. and Lowe D., Multivariable functional interpolation and adaptive networks, *Complex systems*, Vol. 2, 1988, pp. 321-355
- [96] Moody J. and Darken C., Fast learning in networks of locally-tuned processing units, *Neural computation*, Vol. 1, 1989, pp. 281-294
- [97] Girosi F. and Poggio T., Neural networks and the best approximation property, *Biological Cybernetics*, (63), 1990, pp. 169-176
- [98] Hartman E., Keeler J. and Kowalski J., Layered neural networks with gaussian hidden units as universal approximators, *Neural computation*, Vol. 2, 1990, pp. 210-215
- [99] Park E., and Sandberg I.W., Universal approximation using Radial Basis-Function Networks, *Neural Computation*, Vol. 3, (2), 1991, pp. 246-257
- [100] Chen S., Billings S.A, Cowan C.F.N., and Grant P.M., Practical identification of NARMAX models using Radial Basis Functions, *International Journal of Control*, Vol. 52, 1990, pp. 1327-1350
- [101] Kantorovitch L., Akilov G., Analyse Fonctionnelle, Tome 1 : Opérateurs et fonctionnelles linéaires, Traduction Française Editions MIR, Moscou, 1981
- [102] Sandberg I.W., Approximations for nonlinear functionals, *IEEE Transactions on Circuits and Systems -1: Fundamental theory and applications*, Vol. 39, (1),1992, pp. 65-67
- [103] Sjöberg J., Zhang Q., Ljung L., Benveniste A., Deylon B., Glorennec P. Y., Hjalmarsson H. and Juditsky A., Nonlinear black box modeling in system identification : a unified overview, *Automatica*, Vol. 31, (12), 1995, PP.1691-1724
- [104] Schoukens J. and Pintelon R., Identification of linear systems : A practical guideline to accurate modeling, Pergamon press, London (U.K.), 1991
- [105] Ljung L., System identification-Theory for the user, Prentice Hall,NJ, 2nd edition, 1999
- [106] Ljung L., State of the art in linear system identification : time and frequency domain methods, *Proceedings of the 2004 American Control Conference*, Boston, Massachusetts, June 30-July 02, pp. 650-660
- [107] Ljung L., Issues in system identification, *IEEE Control Systems Magazine*, Jan. 1991, pp. 25-29

- [108] Sastry S. and Bodson M., Adaptive control : Stability, Convergence, and Robustness , Prentice Hall, NJ, 1992
- [109] Glad S.T. and Ljung L., Model structure identifiability and persistence of excitation, *Proceedings of 29th IEEE Conference on Decision & Control*, Honolulu, Hawaii, Dec. 1990, pp. 3236-3240
- [110] Ljung L. and Guo L., The role of model validation for assessing the size of the unmodeled dynamics, *IEEE Transactions on Automatic Control*, Vol. 42, (9), 1997, pp. 1230-1239
- [111] Billings S. A., Identification of nonlinear systems—a survey, *IEE Proceedings, Pt. D*, 127, (6), 1980, pp. 272-285
- [112] Willis M. J., Montague G. A., Di Massimo C., Tham M. T. and Morris A. J., Artificial neural networks in process estimation and control, *Automatica*, Vol. 28 , (6) , 1992, pp. 1181-1187
- [113] Tan Y. and De Keyser R., Neural network adaptive predictive control, *Proceedings of Advances in Model-Based Predictive Control*, 1994, Oxford University Press, pp. 358-369
- [114] Neumerkel D., Franz J., Kruger L. and Hidiroglu A., Real-time application of neural model predictive control for an induction servo drive, *Proceedings 3rd IEEE Conference on Control Applications*, 1994, pp. 433-438
- [115] Sbarbaro D., Segovia G. P., Alcozer S. and Gonzales J., Applications of radial basis network technology to process control , *IEEE Transactions on Control System Technology*, Vol. 8 ,(1), 2000, pp. 14-22
- [116] Sandoz D. J., Lennox B., Goulding P. R., Thorp P. J., Kurth T., Desforges M. J. and Woolley I. S., Innovation in industrial model predictive control, *Computing and Control Engineering Journal* , Vol.10 , (5), 1999, pp.189-197
- [117] Platt J., A resource-allocating network for function interpolation, *Neural Computation* , Vol. 3, 1991, pp. 213-225
- [118] Kadiramanathan V. and Niranjana M., A function estimation approach to sequential learning with neural networks, *Neural Computation*, Vol. 5, 1993, pp. 954-975
- [119] Rose K., Gurewitz E., and Fox G. C., A deterministic annealing approach to clustering, *Pattern Recognition Letter*, Vol. 11, Sept. 1990, pp. 589-594
- [120] Ukrainec A. and Haykin S., Signal processing with radial basis function networks using expectation maximisation algorithm clustering, *in Proc. SPIE, Adaptive signal processing*, Vol. 1565, 1991, pp. 529-539

- [121] Sutanto E. L., Mason J. D., and Warwick K., Mean-tracking clustering algorithm for radial basis function centre selection, *International Journal of Control*, Vol. 67, (6), 1997, pp.961-977
- [122] Sanner R. M. and Slotine J.J.E., Gaussian networks for direct adaptive control, *IEEE Transaction on Neural Networks*, Vol. 3, (6), Nov 1992, pp. 837-864
- [123] Yingwei L., Sundararajan N., and Saratchandran P., Identification of time-varying nonlinear systems using minimal radial basis function neural networks, *IEE Proc., Control Theory Appl.* Vol. 144, (2), 1997, pp. 202-208
- [124] Liu G. P., Kadiramanathan V. and Billings S. A., Predictive control for nonlinear systems using neural networks, *International journal of control*, Vol. 71, (6), 1996, pp. 1119-1132
- [125] Kambhampati, C., Delgado, A., Mason, J. D., and Warwick, K., Stable receding horizon control based on recurrent networks, *IEE Proceedings on Control Theory Applications*, 144, 1997, pp. 249-254
- [126] Garces, F., Becerra, V. M., Kambhampati, C., and Warwick, K., Strategies for feedback linearisation: a dynamic neural network approach, Springer-Verlag, London, 2003
- [127] Zhihong, M., Wu, H. R., and Palaniswami, M., An adaptive tracking controller using neural networks for a class of nonlinear systems, *IEEE Transactions on Neural Networks*, Vol. 9, 1998, pp. 947-954
- [128] Hunt K. J. , and Sbarbaro D. , Neural networks for nonlinear internal model control , *IEE Proceedings-D* , Vol. 138 , (5), 1991 , pp. 431-438
- [129] Craddock R. J., Kambhampati C., Tham M. and Warwick K., A practical approach for training dynamic recurrent neural networks : Use of apriori information, *UKACC International Conference on Control'98*, Sept. 1998, pp. 324-329
- [130] Becerra V.M., Garces F.R., Nasuto S.J., and Holderbaum W., An efficient Parameterization of dynamic neural networks for nonlinear system identification, *IEEE transactions on Neural Networks*, Vol. 16, (4) , 2005, pp. 983-988
- [131] Behera L., Gopal M., Chaudhury S., Inversion of RBF networks and application to adaptive control of nonlinear systems, *IEE Proc. Control Theory Appl.* Vol. 142, (6), 1995, pp. 617-624
- [132] Kambhampati C., Craddock R. J., Tham M. and Warwick K., Inverse model control using recurrent networks, *Mathematics and computers in simulation*, Vol. 51, 2000, pp. 181-199

- [133] Lee J. W. and Oh J. H., Inversion control of nonlinear systems with neural network modelling, *IEE Proc. Control Theory Appl.*, Vol. 144, (5), 1997, pp. 481-487
- [134] Economou C. G., Morari M., Palsson B. O., Internal model control: Extension to nonlinear systems, *Industrial Engineering and Chemical Process Design and Development* , Vol. 25, 1986, pp. 403-411
- [135] Psaltis D., Sideris A., and Yamamura A. A., A multilayered neural network controller, *IEEE Control Systems Magazine*, 8, 1988, pp. 17-21
- [136] Zhu K.Y., Qin X.F. ,and Chai T.Y., A new robust non-linear self-tuning predictive control using neural networks , *Proc. Instn. Mech. Engrs. :Journal of Control System engineering* , Vol. 211 ,(1), 1997 , pp.439-446
- [137] Park, Y. M., Choi, M. S., and Lee, K. W., An optimal tracking neuro-controller for nonlinear dynamic systems. *IEEE Transactions on Neural Networks*, Vol. 7, 1996 , pp.1099-1110

Abstract

This thesis is a contribution to nonlinear model predictive control in which are addressed the issues of nonlinear model identification, stability and trajectory tracking. Concerning the nonlinear system identification problem, a gaussian function neural network approach is adopted and, a practical identification technique is proposed for NARX input-output models and for more complex neural structures describing discrete dynamic systems in the state-space. This work shows that, the class of the introduced neural structures is capable of learning the dynamics of the identified system, including its stability properties. The dynamic properties of these neural structures allow their use in nonlinear model predictive control schemes with guaranteed closed-loop stability. The nonlinear optimal state-tracking control problem is solved by using a new hybrid controller architecture. In this proposed architecture the tracking problem is transformed into a nonlinear regulation problem, solved within the framework of a stabilising predictive regulator. On the other hand, the system is maintained close to the desired state trajectory through the action of a feedforward neural controller trained to model the inverse dynamics of the system under control.

Résumé

Cette thèse est une contribution à la commande prédictive nonlinéaire référencée modèle et dans laquelle sont adressées les problématiques de l'identification, de la stabilité et de la poursuite de trajectoire dans l'espace d'état. Concernant l'identification des systèmes nonlinéaires, une approche par les réseaux de neurones à fonctions gaussiennes est adoptée et une technique pratique est proposée pour l'identification des modèles NARX et d'autres structures neuroniques plus complexes décrivant un système dynamique discret dans l'espace d'état. Ce travail montre, aussi, que la classe des structures neuroniques introduites possède des aptitudes d'approximation dans l'espace d'état et, est capable de restituer les propriétés dynamiques et de stabilité du système identifié. Les caractéristiques dynamiques de ces structures neuroniques permettent leur intégration dans des schémas de commande prédictive nonlinéaire assurant la stabilité en boucle fermée. Pour la problématique de la poursuite, une nouvelle architecture de commande à caractère hybride : prédictive et neuronique est proposée. Dans cette architecture, le problème de la poursuite optimale nonlinéaire est transformé en un problème de régulation traité, ensuite, dans le cadre d'une commande prédictive stabilisante. Le maintien du système sur une trajectoire de référence dans l'espace d'état est assuré par un contrôleur neuronique modélisant la dynamique inverse du système sous commande.

