



**HAL**  
open science

# Quelques contributions aux observateurs non linéaires à horizon glissant

Luis Antonio Calvillo Corona

► **To cite this version:**

Luis Antonio Calvillo Corona. Quelques contributions aux observateurs non linéaires à horizon glissant. Automatique / Robotique. Institut National Polytechnique de Grenoble - INPG, 2002. Français. NNT: . tel-00198310

**HAL Id: tel-00198310**

**<https://theses.hal.science/tel-00198310>**

Submitted on 17 Dec 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

--	--	--	--	--	--	--	--	--	--

**THESE**

*pour obtenir le grade de*

**DOCTEUR DE L'INPG**

***Spécialité : «AUTOMATIQUE-PRODUCTIQUE»***

préparée au Laboratoire d'Automatique de Grenoble

dans le cadre de l'École Doctorale «**Électronique, Électrotechnique, Automatique,  
Télécommunication et Signal**»

*présentée et soutenue publiquement*

*par*

**Luis Antonio Calvillo Corona**

*le 19 Décembre 2002*

**Titre :**

**QUELQUES CONTRIBUTIONS AUX OBSERVATEURS NON  
LINÉAIRES À HORIZON GLISSANT**

*Directeurs de thèse :*

M. Mazen Alamir (LAG)

**JURY :**

M. Jean-Michel Dion Président

M. Hassan Hammouri Rapporteur

M. Mohamed Darouach Rapporteur

M. Mazen Alamir Directeur de thèse

M. Guy Bornard Examineur



*À mes parents*



# Remerciements

Je voudrais avant tout remercier les membres du jury, à commencer par Monsieur Jean-Michel Dion, Directeur de Recherche au CNRS pour en avoir accepté la présidence et pour avoir donné une lecture méticuleuse de ce travail, malgré ses nombreuses responsabilités.

Je tiens également à adresser ma sincère et profonde reconnaissance aux Messieurs Hassan Hammouri, Professeur à l'Université Claude Bernard Lyon 1 et Mohamed Darouach, Professeur à l'Université Henry Poincaré Nancy 1, pour leur lecture approfondie de ce travail, ce qui m'a permis de l'améliorer considérablement. Je suis également reconnaissant à Monsieur Guy Bornard, Directeur de recherche au CNRS, pour les excellentes remarques qu'il a fait sur ce travail.

Enfin, je remercie tout particulièrement à Monsieur Mazen Alamir, Chargé de Recherche au CNRS et directeur de cette thèse pour avoir accompagné mon travail pendant ces trois années de thèse. C'est certainement grâce à ses grandes qualités scientifiques, pédagogiques et humaines, ainsi qu'à sa grande patience, que j'ai pu mener à son terme ce travail, qui a été pour moi une grande aventure scientifique, dans un domaine qui m'était au départ peu familier.

Je voudrais également remercier CONACYT et SFERE pour leur soutien.

Je voudrais aussi adresser mes remerciements aux personnes du Laboratoire d'Automatique de Grenoble, dont j'ai reçu un soutien et une amitié constants, les chercheurs de tous les groupes et les équipes, le service technique, Daniel, Philippe, Olivier, le service administratif, Marie-Thé, Marie-Rose, Virginie, Patricia et à tous les thésards anciens et nouveaux et tout particulièrement à Héctor M., Héctor V., Costin, Ali et Fadi.

Finalement, je remercie infiniment le soutien inconditionnel de mes parents, Luis et Silvia, de mes frères, Beba et Toy, de tous mes amis à Grenoble et de celle qui m'a supporté, encouragé et qui a changé ma vie pour toujours ... merci Paloma.



# Table des matières

Table des matières	7
<b>1 Introduction Générale</b>	<b>13</b>
<b>2 La synthèse d'Observateurs Non Linéaires</b>	<b>21</b>
2.1 Observabilité . . . . .	24
2.1.1 Observabilité des systèmes linéaires . . . . .	24
2.1.2 Observabilité des systèmes non linéaires . . . . .	25
2.2 Les Observateurs pour les Systèmes Linéaires . . . . .	27
2.2.1 Observateur de Luenberger . . . . .	27
2.2.2 Observateurs du type Kalman . . . . .	28
2.3 La synthèse d'observateurs pour les systèmes non linéaires . . . . .	29
2.3.1 Le filtre de Kalman étendu . . . . .	29
2.3.2 Méthodes basées sur les fonctions de Lyapunov . . . . .	30
2.3.3 Observateurs pour systèmes sous une forme canonique . . . . .	33
2.3.4 Les observateurs dits " <i>à grand gain</i> " . . . . .	40
2.3.5 Observateurs adaptatifs . . . . .	45
2.3.6 Linéarisation étendue . . . . .	48
2.3.7 Observateurs par modes glissants . . . . .	50
2.3.8 Les observateurs basés sur l'optimisation . . . . .	55
2.4 Conclusion . . . . .	56
<b>3 Vers la Synthèse d'Observateurs Optimaux Robustes</b>	<b>61</b>
3.1 Le problème d'observation . . . . .	64
3.1.1 La classe des systèmes considérés . . . . .	64
3.1.2 Formulation du problème . . . . .	65
3.2 Le principe de l'algorithme . . . . .	66
3.3 Le problème élémentaire . . . . .	69
3.3.1 Définition du problème élémentaire . . . . .	69
3.3.2 Résolution générale du problème élémentaire . . . . .	70
3.3.3 Les équations de Hamilton-Jacobi . . . . .	71



3.3.4	La transformation du problème élémentaire en un problème linéaire quadratique . . . . .	74
3.4	Points stationnaires et calcul des pondérations . . . . .	77
3.4.1	Points stationnaires . . . . .	78
3.4.2	Calcul des pondérations . . . . .	79
3.5	Discussion . . . . .	80
3.6	Exemple d'illustration . . . . .	82
3.6.1	Réacteur continûment agité <b>RCA</b> . . . . .	82
3.6.1.1	Description réacteur continûment agité . . . . .	84
3.6.1.2	Résultats . . . . .	91
3.7	Conclusion . . . . .	114
<b>4</b>	<b>Résultats Supplémentaires sur les Observateurs Non Linéaires à Horizon Glissant</b>	<b>119</b>
4.1	Définitions et Notations . . . . .	120
4.1.1	Les définitions associées au système . . . . .	120
4.1.2	Définitions associées à l'observabilité . . . . .	122
4.1.3	Les définitions associées à la conception de l'observateur . . . . .	123
4.2	Le rayon d'observabilité . . . . .	127
4.3	Résultats de convergence . . . . .	132
4.4	Exemple d'illustration . . . . .	133
4.5	Conclusion . . . . .	134
<b>5</b>	<b>Conclusion Générale</b>	<b>139</b>
<b>A</b>	<b>Objets et opérateurs différentiels</b>	<b>143</b>
<b>B</b>	<b>Modules Scalaires Parallèles</b>	<b>145</b>
B.1	Module Paramètre . . . . .	145
B.2	Module FLH . . . . .	150
B.3	Module Initial . . . . .	155
B.4	Module Résolution . . . . .	156
B.5	Module Traitement . . . . .	163
B.6	Liste des modules . . . . .	165
B.6.1	Module Paramètre . . . . .	165
B.6.2	Module FLH . . . . .	166
B.6.3	Module Initial . . . . .	166
B.6.4	Module de résolution . . . . .	167
B.6.5	Procédures Externes . . . . .	167
B.6.6	Module Traitement . . . . .	167
B.6.7	Programme Principal . . . . .	168

---

<b>C Preuves</b>	<b>170</b>
C.1 Idée de la preuve du théorème de vérification 3.3.2 . . . . .	170
C.2 Preuve du Corollaire 4.2.2 . . . . .	172
C.3 Preuve de la Proposition 4.2.3 . . . . .	172
C.4 Preuve de la Proposition 4.3.1 . . . . .	172
 <b>Bibliographie</b>	 <b>177</b>



# Introduction Générale



# Chapitre 1

## Introduction Générale

La théorie des systèmes non linéaires a connu des développements significatifs pendant les deux dernières décennies. Basées sur la linéarisation par retour d'état, des méthodologies de commande ont été conçues pour contourner les non linéarités dans les modèles des processus physiques. Ainsi, la théorie de la commande par retour d'état des systèmes non linéaire est aujourd'hui relativement bien avancée.

Cependant, dans beaucoup de situations pratiques, pour des raisons techniques ou économiques (construction, positionnement et/ou coût des capteurs), il n'est pas possible d'accéder à toutes les variables d'état d'un système pour appliquer le retour d'état. Une alternative intéressante est alors de produire une estimée de la grandeur recherchée en utilisant la connaissance du processus (Le modèle, la commande et les mesures). Le dispositif utilisé pour estimer le vecteur d'état est appelé **observateur**.

Il peut y avoir une multitude de raisons pour lesquelles une mesure ou une estimée est désirée, mais pour l'automaticien, l'utilisation d'un observateur peut être envisagée pour répondre à trois catégories d'objectifs, à savoir, la surveillance, la détection de défaillances et la commande.

### Surveillance

L'utilisateur peut avoir besoin d'une estimée des variables d'état d'un certain processus pour prendre l'action appropriée. Un pilote peut, par exemple, vouloir connaître l'angle de tangage de l'avion afin de savoir s'il est près de l'angle de décrochage. Autre exemple est un opérateur qui nécessite une estimée de l'état d'un procédé discontinu afin de savoir quand l'arrêter. Cette situation est illustrée sur la Figure 1.1. L'observateur prend les entrées et les sorties pertinentes du processus et fournit une estimée de l'état du processus. Quand l'estimé est seulement utilisé comme signal de surveillance, il n'existe pas de boucle fermée dynamique impliquant cette estimation.

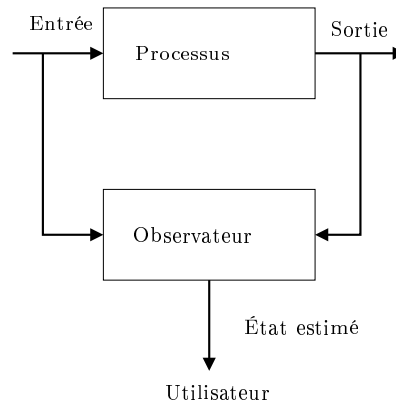


FIG. 1.1: *Un observateur utilisé pour la surveillance de processus*

## Détection de défaillances

Les défaillances dans des processus techniques sont presque inévitables. Par exemple, la fuite dans une soupape, un palier cassé, la défaillance d'un capteur. Ces défaillances peuvent causer des ruptures sévères, aboutissant à la perte de capacité de production voire à des accidents. La capacité de détecter des défaillances avant qu'elles n'aient des conséquences sérieuses est donc fortement désirable.

Les techniques de détection de défaillances basées sur la redondance analytique emploient souvent un observateur pour déterminer l'état du processus, et ainsi pouvoir obtenir des conclusions sur des défaillances possibles. La Figure 1.2 montre la structure de principe d'un système de détection des défaillances. Le bloc de détection des défaillances prend la décision si une défaillance est arrivée ou pas et peut également déterminer son amplitude et son emplacement, on parle alors de localisation des défauts.

L'idée est de générer un signal appelé résidu  $\epsilon(t)$  qui a la propriété ( $\epsilon(t) \neq 0$ ) en cas de défaillance et la propriété ( $\epsilon(t) = 0$ ) en l'absence de défaillance. Il existe deux principales manières de représenter les défauts en utilisant un observateur pour la détection des défaillances.

D'une part, la défaillance peut être représentée comme un signal externe non mesuré. L'occurrence d'une défaillance cause alors un résidu non nul, qui peut ainsi être employé comme un signal de défaillance. Une autre manière de représenter un défaut s'obtient en ajoutant une variable d'état qui est alors estimée par l'observateur avec les variables d'état du processus.

De toute façon, en raison des entrées non mesurées telles que les erreurs de modèle ou le bruit de mesure, le résidu ne satisfera jamais exactement  $\epsilon(t) = 0$  en l'absence de défaut, et doit donc être traité par une logique de décision représentant une couche supérieure. Dans la pratique, ceci signifie qu'un seuil  $E(t)$  est placé pour  $\epsilon(t)$  et si  $\epsilon(t) > E(t)$  alors une défaillance est déclarée présente.

Les perturbations peuvent aussi être considérées comme des signaux non mesurés

qui sont bornés dans un certain sens. Le seuil est alors calculé de sorte que  $\epsilon(t) < E(t)$  soit satisfaite tant que les perturbations satisfont leurs bornes indiquées et qu'il n'y ait aucune défaillance. La détection des défaillances ne sera pas abordée dans ce mémoire, le lecteur intéressé se reportera à [7; 49; 50]

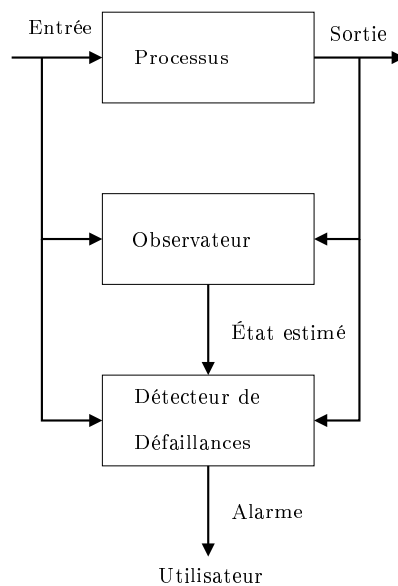


FIG. 1.2: *Un observateur dans un système de détection des défaillances*

## Commande

Puisque le vecteur d'état est, en général, non mesuré, il doit être estimé en utilisant un observateur. Cette situation est illustrée dans la Figure 1.3. Les objectifs sont dans ce cas radicalement différents des deux applications précédentes. L'exactitude de l'estimation de l'état n'est plus une fin en soi, mais à la place, la commande du comportement de l'état est le but global. Un problème complexe de stabilité surgit puisqu'il existe une boucle fermée impliquant le contrôleur, l'observateur et le processus à commander. Pour des systèmes linéaires le principe de séparation permet au vecteur estimé de l'état d'être employé au lieu du vecteur des états du processus dans la loi de commande, sans affecter la stabilité de la boucle fermée, mais ce n'est pas généralement le cas pour des systèmes non linéaires. Cette question de l'interaction entre la commande et l'observation ne sera pas abordée ici. Pour des études concernant cette question, voir par exemple [101]. En général, une saturation explicite de la commande basée sur l'état estimé est à utiliser aboutissant à des résultats de stabilisation semi-globaux [47; 130; 131].



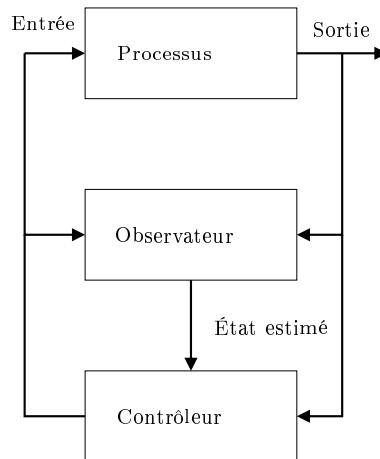


FIG. 1.3: *Un observateur dans une boucle de commande*

Des travaux menés ces dernières années [48; 67; 100], il résulte à l’heure actuelle toute une panoplie d’outils théoriques d’analyse, de commande et d’observation des systèmes non linéaires. Pourtant, le problème de la synthèse d’observateurs pour les systèmes non linéaires est encore largement ouvert, car malgré la grande variété de résultats pour la synthèse d’observateurs pour systèmes non linéaires, il n’existe pas de solutions systématiques. En fait, il n’existe qu’une gamme de méthodologies de synthèses disponibles pour des systèmes sous des formes particulières, ou bien vérifiant des conditions structurelles particulières.

Les principales techniques de synthèse d’observateurs non linéaires sont les suivantes

- Observateurs basés sur les fonctions de Lyapunov.
- Observateurs pour systèmes sous forme canonique.
- Observateurs à grand gain.
- Observateurs adaptatifs.
- Observateurs par linéarisation étendue.
- Observateurs par modes glissants.
- Observateurs par optimisation

Les aspects techniques de ces approches sont décrits dans le Chapitre 2, mais de cette étude, il peut être déjà énoncé que les observateurs obtenus des différentes méthodologies de synthèse souffrent au moins de l’un des inconvénients suivants

- La synthèse nécessite des conditions structurelles difficiles à satisfaire ou à mettre à jour.

- Les erreurs de modèle et les perturbations ne sont pas explicitement prises en considération. Elles sont implicitement prises en compte par la marge de robustesse de l'observateur en tant que système, en quelque sorte, bouclé. En d'autres termes, la caractérisation des amplitudes possibles de ces incertitudes ne conduit pas à une action spécifique sur la synthèse de certains observateurs.
- La synthèse nécessite des calculs lourds.

Compte tenu des difficultés et des inconvénients rencontrés pour la synthèse des observateurs non linéaires, le premier objectif de ce travail est d'aller dans le sens d'obtenir une nouvelle méthodologie pour construire des observateurs optimaux robustes qui tentent de respecter les objectifs suivants :

- Indépendance vis à vis des conditions structurelles du système de manière à obtenir une approche plus systématique et générique.
- Prise en compte explicite des incertitudes et/ou des perturbations.
- Temps de calcul réduit.

Sur la base de la discussion précédente, la méthodologie proposée dans le Chapitre 3 est une méthode itérative qui consiste d'abord à diviser le problème d'optimisation principal en  $n$  problèmes d'optimisation scalaires, dénommés *Problèmes Élémentaires*, puis, à résoudre chaque problème scalaire en utilisant un solveur numérique de l'équation de Hamilton-Jacobi scalaire (en la variable de l'espace). Les solutions des différents problèmes scalaires sont alors "analysées" pour définir un nouvel ensemble de problèmes scalaires élémentaires pour l'itération suivante. Les itérations continuent jusqu'à ce que toutes les solutions soient mutuellement compatibles. Cette méthodologie a été présentée dans [36]

L'autre partie de ce travail est constituée par les observateurs à horizon glissant plus classiques [4; 95; 115; 116; 152]. L'avantage principal dans l'utilisation de tels observateurs est l'absence de toute hypothèse sur l'existence de formes canoniques pour que l'observateur correspondant soit conçu. Cependant, les calculs associés sont généralement très lourds et peuvent empêcher l'utilisation de ces observateurs pour des systèmes avec une dynamique "rapide".

Dans [4], des solutions techniques ont été proposées pour partiellement surmonter ces inconvénients à l'aide de la technique de post-stabilisation. Cependant, une hypothèse de régularité uniforme globale a été employée afin d'obtenir les résultats de convergence. Le deuxième objectif de ce travail est alors de traiter le cas où de telles hypothèses ne sont pas vérifiées. Ces résultats ont été présentés dans [5; 6]

## Organisation du mémoire

Le mémoire est organisé de la façon suivante

1. Le chapitre 2 est consacré aux rappels des notions de base de l'observabilité des systèmes linéaires et non linéaires, ainsi qu'à la description des différentes techniques

de synthèse d'observateurs où les avantages et les inconvénients de chaque méthode sont énoncés. Les méthodologies de synthèse présentées dans ce chapitre sont les plus utilisées dans le domaine.

2. L'étude des différentes techniques de synthèse a permis de développer une nouvelle méthodologie de synthèse d'observateur pour des systèmes non linéaires qui est présentée dans le chapitre 3. Le travail développé dans ce chapitre est plutôt exploratoire et à ce titre, il devrait être vu et jugé comme un premier pas vers la construction d'un observateur à horizon glissant permettant un traitement explicite des incertitudes.
3. Dans le chapitre 4, l'intérêt est concentré sur les observateurs à horizon glissant plus classiques. Le cas où les hypothèses de régularité uniforme globale ne sont pas supposées comme dans [4] est traité. Les résultats de ce chapitre sont essentiellement de nature théorique. Ils permettent une meilleure compréhension de ce que coûterait l'absence de l'hypothèse de régularité globale utilisée dans [4]. Les notions intuitives de rayon de régularité et de rayon de convergence sont rigoureusement définies.

# La Synthèse d'Observateurs Non Linéaires



## Chapitre 2

# La synthèse d'Observateurs Non Linéaires

La disponibilité de toutes les variables d'état pour la mesure directe est rarement vérifiée dans la pratique. Il existe dans la plupart des cas un vrai besoin d'une estimation fiable des variables non mesurées, particulièrement quand elles sont employées pour la synthèse de lois de commande ou pour la surveillance des processus. Pour cette tâche particulière, un observateur d'état est alors nécessaire pour reconstruire les variables d'état du processus.

La construction d'un observateur exige la propriété d'observabilité. Initialement formalisée par [62], cette propriété peut être définie d'une façon générale à partir de la notion d'indistinguabilité d'une paire d'états, au sens où un système est dit observable si toute paire d'états du système peut être distinguée grâce aux sorties correspondantes, pour au moins une entrée.

Quand un système linéaire est observable, il est observable indépendamment de l'entrée et l'observabilité se trouve complètement caractérisée par la paire  $(C, A)$ . De plus, contrairement au cas non linéaire général, elle suffit pour garantir l'existence d'un observateur, à vitesse de convergence exponentielle et arbitrairement rapide.

Pour les systèmes non linéaires, ceci n'est plus vrai. En général les systèmes non linéaires ont des entrées singulières qui les rendent non observables. Ainsi, d'une façon générale, l'observabilité d'un système non linéaire dépend de l'entrée appliquée. Cependant si le système n'admet que des entrées permettant de distinguer toute paire d'états (entrée universelle ou non singulière [129]) alors l'observabilité du système est uniforme [53].

Compte tenu de l'abondance et de la diversité des observateurs disponibles pour diverses classes de systèmes, il est souhaitable de formaliser une approche "générale" de la question permettant théoriquement d'exploiter au mieux les résultats existants et d'en obtenir de nouveaux. Ainsi, d'une façon générale, les systèmes non linéaires considérés

sont décrits par une représentation d'état explicite de la forme suivante

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t)) \\ y &= h(x(t))\end{aligned}\tag{2.0.1}$$

où  $x$  représente l'état interne du système,  $u$  représente la commande ou entrée et  $y$  représente la sortie mesurée ou simplement la mesure. De plus, l'état appartient à l'espace  $\mathbb{R}^n$  ou à une sous variété ouverte de  $\mathbb{R}^n$  ( $x \in \mathcal{X} \subset \mathbb{R}^n$ ), la commande appartient à l'espace  $\mathbb{R}^m$  ou à un sous ensemble de  $\mathbb{R}^m$  ( $u \in \mathcal{U} \subset \mathbb{R}^m$ ) et la sortie appartient à l'espace  $\mathbb{R}^p$  ou à un sous espace de  $\mathbb{R}^p$  ( $y \in \mathcal{Y} \subset \mathbb{R}^p$ ). Enfin, les entrées en tant que fonctions du temps sont en général supposées choisies dans un ensemble d'entrées dites admissibles de fonctions mesurables et bornées.

### Le problème d'observation

Étant donné un système décrit par les équations (2.0.1), le rôle d'un observateur consiste à estimer l'état courant  $x(t)$  du système à partir des grandeurs supposées connues que sont ses entrées  $u$  (issues d'un algorithme de commande) et ses sorties  $y$  (mesurées par des capteurs).

Un observateur est donc un système dynamique auxiliaire dont les entrées sont les entrées/sorties du système (2.0.1), et les sorties sont les états estimés  $\hat{x}$  comme est montrée par la Figure 1.1.

Un tel système peut être représenté d'une façon générale par

$$\begin{aligned}\dot{z}(t) &= \hat{f}(z(t), u(t), y(t)) \\ \hat{x} &= \hat{h}(z(t), u(t), y(t))\end{aligned}\tag{2.0.2}$$

et doit vérifier la propriété que l'état estimé converge vers l'état réel, c'est à dire

$$\|\hat{x}(t) - x(t)\| \rightarrow 0 \quad \text{quand } t \rightarrow \infty$$

Bien souvent, la forme de l'observateur se résume à une copie de la dynamique du système, à laquelle est ajoutée un terme de correction basé sur l'écart entre la sortie mesurée, et la sortie estimée du système (avec éventuellement une extension dynamique)

$$\begin{aligned}\dot{\hat{x}}(t) &= f(\hat{x}(t), u(t)) - k(\hat{z}(t), y(t) - h(\hat{x}(t), \hat{z}(t))) \\ \dot{\hat{z}} &= Z(z(t), u(t), y(t)) \quad \text{avec } k(z(t), 0) = 0\end{aligned}\tag{2.0.3}$$

En toute généralité, et en considérant les représentations ci-avant, un observateur correspond à un système dynamique d'ordre supérieur à celui du système observé (en raison de l'équation de gain).

Dans le cas des systèmes linéaires, une solution bien connue est donnée par l'observateur de Luenberger [89]. Cet observateur est tout simplement une copie de la

dynamique du système à laquelle est ajoutée un terme de correction sous la forme d'un gain constant multiplié par l'erreur d'estimation de la sortie.

Dans le domaine des systèmes non linéaires, la synthèse d'observateurs est plus difficile et beaucoup moins systématique. C'est la raison pour laquelle elle a suscité une quantité considérable de travaux dans la littérature. Malgré l'abondance des résultats pour la synthèse d'observateurs non linéaires, il n'existe pas de méthode systématique de synthèse d'observateurs pour un système sous une forme générale (2.0.1).

Pour cette raison, les efforts se sont centrés sur la synthèse d'observateurs pour différentes classes spécifiques de systèmes non linéaires. Les formes de systèmes les plus étudiées sont les suivantes

- Les systèmes bilinéaires

$$\begin{aligned}\dot{x}(t) &= Ax(t) + \sum_{i=1}^m u_i(t)D_i x(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}\tag{2.0.4}$$

- Les systèmes affines en l'état

$$\begin{aligned}\dot{x}(t) &= A(u(t))x(t) + B(u(t)) \\ y(t) &= C(u(t))x(t)\end{aligned}\tag{2.0.5}$$

- Les systèmes affines modulo injection d'entrées/sorties

$$\begin{aligned}\dot{x}(t) &= A(u(t))x(t) + B(u(t), y(t)) \\ y(t) &= Cx(t)\end{aligned}\tag{2.0.6}$$

- Les systèmes affines en la commande

$$\begin{aligned}\dot{x}(t) &= f(x(t)) + \sum_{i=1}^m g(x(t))u_i(t) \\ y(t) &= h(x(t))\end{aligned}\tag{2.0.7}$$

- Les systèmes linéaires perturbés

$$\begin{aligned}\dot{x}(t) &= Ax(t) + f(x(t), u(t)) \\ y(t) &= Cx(t)\end{aligned}\tag{2.0.8}$$

- Les systèmes autonomes

$$\begin{aligned}\dot{x} &= f(x) \\ y &= h(x)\end{aligned}\tag{2.0.9}$$



Ce chapitre est dédié à un survol, forcément non exhaustif des techniques de synthèse d'observateurs pour les systèmes non linéaires. S'il ne réussira jamais à rendre compte de toutes les subtilités des approches proposées, il permettra néanmoins de souligner l'étendue du sujet et la distance qui nous sépare encore d'une solution systématique satisfaisante.

Ce chapitre est divisé en trois parties

- La première partie est dédiée à quelques rappels des notions de base d'observabilité des systèmes linéaires et non linéaires.
- La deuxième partie rappelle de manière brève le principe de l'observateur de Luenberger et le filtre de Kalman. La raison en est que certaines approches de synthèse d'observateurs non linéaires utilisent une extension de l'observateur de Luenberger ou du filtre de Kalman.
- La dernière partie est consacrée aux principales (ou plus utilisées) techniques de synthèse d'observateurs non linéaires.

## 2.1 Observabilité

Dans ce paragraphe, les notions d'observabilité des systèmes linéaires et non linéaires sont rappelées. Ces rappels sont donnés pour montrer que dans le cas linéaire, la condition d'observabilité est indépendante de l'entrée, et elle est également suffisante pour garantir l'existence d'un observateur à vitesse de convergence exponentielle et arbitrairement rapide [89]. Pour les systèmes non linéaires, la synthèse dépend des entrées, puisque leur observabilité en dépend. Ainsi, pour le cas de systèmes non linéaires une classification des différents types d'entrées est donnée.

### 2.1.1 Observabilité des systèmes linéaires

Les résultats classiques d'observabilité que l'on peut trouver dans [72], [145] sont rappelés dans cette partie.

#### Observabilité

La classe de systèmes considérés est de la forme

$$\begin{aligned} \dot{x} &= Ax + Bu \quad ; \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m \\ y &= Cx \quad ; \quad y \in \mathbb{R}^p \end{aligned} \tag{2.1.1}$$

La propriété d'observabilité d'un tel système peut être formalisée de la façon suivante

**Théorème 2.1.1.** *Le système (2.1.1) est observable si et seulement si*

$$\text{rang} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} = n \quad (2.1.2)$$



En fait, ce résultat signifie que le sous espace non observable défini par

$$\mathcal{N} = \text{Ker} \left( \bigcap_{i=1}^n CA^{i-1} \right)$$

(le plus grand sous espace vectoriel A-invariant contenu dans  $\text{Ker } C$ ) est réduit à  $\{0\}$ .

## 2.1.2 Observabilité des systèmes non linéaires

Pour un système non linéaire de la forme

$$\begin{aligned} \dot{x} &= f(x, u) & ; & \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m \\ y &= h(x) & ; & \quad y \in \mathbb{R}^p \end{aligned} \quad (2.1.3)$$

le problème d'observation se complique, dans la mesure où contrairement au cas linéaire, l'observabilité dans ce cas dépend de l'entrée appliquée. Soit  $\mathcal{X}_u(t, x_0)$  la solution à l'instant  $t$  du système sous la condition initiale  $x_0$  à l'instant 0 et soumis à la commande  $u$ . L'observabilité peut alors être définie à partir de la relation d'indistinguabilité (ou indiscernabilité)[62]. Une bonne synthèse sur la question est donnée dans [30].

### Observabilité, condition de rang et problème des entrées

**Définition 2.1.1.** *[Indiscernabilité][30]*

*Une paire  $(x_0, \tilde{x}_0)$  est dit indiscernable par tout  $u$  si*

$$\forall t \geq 0 \quad h(\mathcal{X}_u(t, x_0)) \equiv h(\mathcal{X}_u(t, \tilde{x}_0))$$



**Définition 2.1.2.** *[Observabilité][30]*

*Un système non linéaire (2.1.3) est observable s'il n'admet pas de paires indiscernables.*



Contrairement au cas linéaire, il n'existe pas de conditions géométriques globales garantissant l'observabilité d'un système non linéaire. Cependant, en localisant la relation d'indiscernabilité, une notion d'observabilité locale faible est introduite dans [62], garantissant que tout point est discernable par rapport aux points appartenant à son voisinage. Cette propriété est caractérisable par une condition de rang analogue au cas linéaire.

En notant  $I_U(x_0)$ , pour un ouvert  $U \subset \mathbb{R}^n$  et  $x_0 \in U$ , l'ensemble de points indiscernables de  $x_0$  tant que les trajectoires qui en sont issues restent dans  $U$ , on a la définition suivante

**Définition 2.1.3.** [*Observabilité locale faible*][62]

Un système (2.1.3) est localement faiblement observable en  $x_0$  s'il existe un voisinage  $U$  de  $x_0$  tel que  $I_U(x_0) \cap U = \{x_0\}$ . ♠

Pour les systèmes linéaires stationnaires (2.1.1), l'observabilité est caractérisée par la condition de rang (2.1.2). Si l'espace vectoriel  $\mathcal{O}$  des fonctions de  $\mathbb{R}^n$  dans  $\mathbb{R}^p$  engendré par les fonctions  $Cx, CAx, \dots, CA^{n-1}x$  et l'espace  $d\mathcal{O}$  des différentielles (constantes) de chacune de ces fonctions sont considérées, alors, en chaque point  $x$ , l'évaluation en  $x$  de  $d\mathcal{O}$  est précisément engendrée par les colonnes de la matrice (2.1.2). Ceci amène à définir, pour les systèmes non linéaires, l'espace d'observabilité qui permet d'exprimer une condition de rang.

**Définition 2.1.4.** [*Espace d'observabilité*][30]

L'espace d'observabilité de (2.1.3) est le plus petit sous espace vectoriel  $\mathcal{O}(h)$  de fonctions de  $\mathbb{R}^n$  à valeurs dans l'espace de sortie, contenant les sorties  $h = (h_1, \dots, h_p)$  et invariant sous l'action de la dérivée de Lie par rapport à tous les champs de vecteurs du type  $f_u(x) = f(x, u)$ ,  $u \in \mathbb{R}^n$ , fixé. ♠

En notant  $d\mathcal{O}(h)$  l'espace des différentielles de  $\mathcal{O}(h)$  et  $d\mathcal{O}(h)(x_0)$  l'espace des différentielles évaluées en  $x_0$ , la condition du rang en  $x_0$  est alors donnée par

$$\dim d\mathcal{O}(h)(x_0) = n \quad (2.1.4)$$

Le système est dit satisfaire la condition de rang d'observabilité, s'il la satisfait pour tout  $x_0$ .

**Théorème 2.1.2.** [62]

Si le système (2.1.3) satisfait la condition de rang en  $x_0$  alors il est localement faiblement observable en  $x_0$ . ♠

Contrairement au cas linéaire, pour un système non linéaire la condition de rang (2.1.4) n'est en général pas suffisante pour la synthèse d'un observateur.

**Remarque 2.1.1.** La Définition 2.1.1 et la Définition 2.1.2 n'excluent pas le fait qu'un système observable puisse admettre des entrées, pour lesquelles il existe des états indiscernables par  $u$ . □

Ceci revient à rappeler le fait qu'en général, contrairement au cas linéaire, l'observabilité d'un système non linéaire ne suffit pas pour la synthèse d'un observateur, et qu'une telle synthèse doit prendre en compte le problème des entrées. Il en résulte que l'étude des propriétés des entrées pour un système donné est d'une grande importance pour la synthèse d'observateurs; un cas particulier est celui d'une entrée pour laquelle il n'existe pas de paire indiscernable.

**Définition 2.1.5.** [*Entrées universelles*][30]

Une entrée  $u$  est universelle sur  $[0, t]$  si pour tout couple d'états initiaux distincts  $x_0 \neq \tilde{x}_0$ , il existe  $\tau \in [0, t]$  tel que  $h(\mathcal{X}_u(\tau, x_0)) \neq h(\mathcal{X}_u(\tau, \tilde{x}_0))$ . De plus

- Une entrée universelle sur  $\mathbb{R}^+$  est dite universelle.
- Une entrée non universelle est dite singulière.



**Définition 2.1.6.** [30] Un système dont toutes les entrées sont universelles sur  $[0, t]$  est dit uniformément observable. Si pour tout  $t > 0$ , toutes les entrées sont universelles sur  $[0, t]$ , le système est dit uniformément localement observable.

Il existe un autre type d'entrées qui a été largement étudié pour la classe des systèmes affines en l'état du type

$$\dot{x} = A(u)x + B(u) \tag{2.1.5}$$

$$y = Cx \tag{2.1.6}$$

ces entrées connues comme entrées régulièrement persistantes ne sont pas abordées dans ce mémoire, une étude complète de ce sujet ainsi que celle de la synthèse d'observateurs pour la classe de systèmes (2.1.5) est donnée dans [24; 30].

## 2.2 Les Observateurs pour les Systèmes Linéaires

### 2.2.1 Observateur de Luenberger

La classe de systèmes considérée est de la forme

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.2.1}$$

$$y(t) = Cx(t)$$

L'équation dynamique de l'observateur est

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) \tag{2.2.2}$$

L'erreur est dénotée par  $e = x - \hat{x}$  qui satisfait l'équation différentielle

$$\dot{e} = (A - KC)e \quad (2.2.3)$$

Si la paire  $(A, C)$  est observable, alors la matrice de gain  $K$  de l'observateur peut être choisie (par exemple en utilisant la méthode du placement de pôles) de telle manière que les valeurs propres de  $(A - KC)$  soient strictement négatives. En conséquence, l'erreur converge à zéro quand  $t \rightarrow \infty$  à un taux exponentiel désirable. L'observateur (2.2.2) est connu comme l'observateur de Luenberger [89].

## 2.2.2 Observateurs du type Kalman

Le problème d'estimation de l'état des systèmes dynamiques linéaires perturbés par des bruits Gaussiens a été résolu par [73].

Plus spécifiquement, la classe de systèmes considérée est celle des systèmes linéaires temps-variant uniformément observables

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) + \eta(t), & \eta(t) &\sim \mathcal{N}(0, Q(t)) \\ y(t) &= C(t)x(t) + \varsigma(t), & \varsigma(t) &\sim \mathcal{N}(0, R(t)) \end{aligned} \quad (2.2.4)$$

où les matrices  $A(t)$ ,  $C(t)$  sont supposées bornées et  $\eta$ ,  $\varsigma$  sont des bruits Gaussiens de moyenne nulle et de matrice de covariance  $Q$  et  $R$  respectivement. L'observateur pour (2.2.4) est donné par le théorème suivant

### **Théorème 2.2.1.** [30]

*Soit le système (2.2.4) complètement uniformément observable, et le système*

$$\begin{aligned} \dot{\hat{x}}(t) &= A(t)\hat{x}(t) - S^{-1}(t)C^T(t)Q(C(t)\hat{x}(t) - y(t)) \\ \dot{S}(t) &= -\theta S(t) - A(t)^T S(t) - S(t)A(t) + C^T(t)QC(t) - S(t)RS(t) \\ S(0) &= S_0 \quad \text{où } S_0 \text{ (est définie positive)} \end{aligned} \quad (2.2.5)$$

Si l'une des deux conditions suivantes est satisfaite

1. La paire  $(A(t), R(t))$  est complètement uniformément commandable.
2.  $\theta > 0$

alors le système (2.2.5) est un observateur pour le système (2.2.4), dont la norme de l'erreur d'estimation est bornée par une exponentielle décroissante. De plus, le choix de la valeur de  $\theta$  permet d'obtenir une vitesse de convergence arbitraire. ♠

L'interprétation de l'observateur de Kalman en terme de minimisation d'une erreur de poursuite de la sortie (au sens des moindres carrés) est donnée par le théorème suivant

**Théorème 2.2.2.** [40]

L'état estimé fourni par l'observateur (2.2.5) avec  $R = 0$ , est solution du problème de minimisation suivant

$$\hat{x}(t) \quad : \quad \min_{\hat{x}} J(\hat{x}) = \left\{ \|\Phi(0, t)\hat{x} - \hat{x}_0\|_{S_0}^2 + \int_0^t \|C(\tau)\Phi(\tau, t)\hat{x} - y(\tau)\|_Q^2 d\tau \right\} \quad (2.2.6)$$

où  $\Phi(\cdot, \cdot)$  est la matrice de transition de la partie autonome du système (2.2.4). ♠

Quelques résultats de convergence pour le filtre de Kalman sont présentés dans [32].

## 2.3 La synthèse d'observateurs pour les systèmes non linéaires

Comme il a été déjà mentionné, il n'existe pas de méthode systématique de synthèse d'observateurs pour un système sous une forme générale (2.0.1). Le but de ce paragraphe est de donner une vue d'ensemble des approches qui sont appliquées à la conception des observateurs non linéaires. La liste n'est pas exhaustive et la classification n'est pas unique puisque des combinaisons des différentes approches ont été envisagées.

### 2.3.1 Le filtre de Kalman étendu

Le filtre de Kalman Étendu (FKE) [70] est une extension du filtre de Kalman linéaire au cas où le système est décrit dans l'espace d'état par une équation différentielle non linéaire. Sa conception est basée sur une linéarisation locale du système autour d'une trajectoire de référence. La classe de systèmes considérée est de la forme

$$\begin{aligned} \dot{x}(t) &= f(x(t), t) + \eta(t), & \eta(t) &\sim \mathcal{N}(0, Q(t)) \\ y(t) &= h(x(t), t) + \varsigma(t), & \varsigma(t) &\sim \mathcal{N}(0, R(t)) \end{aligned} \quad (2.3.1)$$

où  $\eta, \varsigma$  sont des bruits Gaussiens de moyenne nulle et de matrice de covariance  $Q$  et  $R$  respectivement. Pour le système (2.3.1), le filtre est donné par

$$\begin{aligned} \dot{\hat{x}}(t) &= f(\hat{x}(t), t) + S(t)H(\hat{x}(t), t)^T R^{-1}(t)(y(t) - h(\hat{x}(t), t)) \\ \dot{S}(t) &= F(\hat{x}(t), t)S(t) + S(t)F(\hat{x}(t), t)^T + Q(t) - S(t)H(\hat{x}(t), t)^T R^{-1}(t)H(\hat{x}(t), t)S(t) \end{aligned} \quad (2.3.2)$$

où

$$\begin{aligned} F(x(t), t) &= \frac{\partial f(x(t), t)}{\partial x(t)} \\ H(x(t), t) &= \frac{\partial h(x(t), t)}{\partial x(t)} \end{aligned}$$

sont évaluées en  $x(t) = \hat{x}(t)$ .

## Commentaires sur l'approche

### *Avantages*

- Sa grande ressemblance avec le filtre de Kalman linéaire. Ceci n'est pas à minimiser dans la mesure où cette ressemblance permet de mieux assimiler cet observateur par des praticiens ayant par la force des choses plus l'habitude des techniques de l'automatique linéaire.
- L'approche permet de traiter d'une manière relativement objective le cas des bruits dont on est capable de caractériser les propriétés statistiques (même si le caractère optimal est perdu par la linéarisation).

### *Inconvénients*

- La performance du FKE ne peut pas être garantie et sa stabilité ne peut être prouvée.
- Cette approche reste une approche basée sur la linéarisation qui n'est valable que dans un domaine réduit autour de la trajectoire nominale.
- Aucune robustesse n'est garantie contre les erreurs de modélisation.
- La mise en oeuvre exige beaucoup de calculs en ligne. Des variantes, comme le FKE à gain constant [118], ont été conçues pour éviter les longs calculs liés à la mise à jour des estimations de l'état et des matrices de covariance.

### 2.3.2 Méthodes basées sur les fonctions de Lyapunov

Une première approche pour construire des observateurs pour les systèmes non linéaires utilise des arguments du type Lyapunov appliqués à l'équation de l'évolution de l'erreur. Cette méthode, proposée dans [132] et généralisée dans [77], mène aux résultats donnant des conditions suffisantes pour l'existence des observateurs, mais ne donne malheureusement pas les outils pour construire un tel observateur.

Les systèmes considérés sont de la forme

$$\begin{aligned} \dot{x} &= Ax + f(x, u) \\ y &= Cx \end{aligned} \tag{2.3.3}$$

où  $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  est continue et  $A \in \mathbb{R}^{n \times n}$ ,  $C \in \mathbb{R}^{p \times n}$ . La paire  $(A, C)$  est supposée complètement observable, un gain  $K \in \mathbb{R}^{p \times n}$  tel que  $A - KC$  soit Hurwitz (les valeurs propres de  $A - KC$  soient dans la moitié gauche du plan complexe) peut alors être trouvée.

L'observateur est donné par

$$\dot{\hat{x}} = (A - KC)\hat{x} + f(\hat{x}, u) + Ky \quad (2.3.4)$$

L'erreur est dénotée par  $e = x - \hat{x}$  qui satisfait l'équation différentielle

$$\dot{e} = (A - KC)e + f(x, u) - f(\hat{x}, u) = (A - KC)e + f(x, u) - f(x - e, u) \quad (2.3.5)$$

Comme les valeurs propres de  $A - KC$  sont strictement négatives, pour n'importe quelle matrice définie positive  $Q \in \mathbb{R}^{n \times n}$  il existe une matrice unique  $P \in \mathbb{R}^{n \times n}$  définie positive telle que  $(A - KC)^T P + P(A - KC) = -Q$ .

La fonction  $f$  est supposée localement Lipschitz autour de l'origine, c'est à dire, il existe une constante positive  $L$  telle que

$$\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|$$

pour tout  $x_1, x_2$  dans un voisinage  $W$  de l'origine. En utilisant la fonction de Lyapunov candidate

$$V(e) = e^T P e$$

il est montré dans [132] que l'estimation de l'erreur  $e$  converge asymptotiquement à l'origine si

$$\frac{\lambda_{\min}(Q)}{2\lambda_{\max}(P)} > L \quad (2.3.6)$$

Dans [111] un algorithme est utilisé pour choisir la matrice de gain  $K$  qui assure la stabilité. L'algorithme proposé est le suivant:

S'il existe un  $\epsilon > 0$  suffisamment petit tel que l'équation de Riccati

$$AP + PA^T + P\left(L^2 I - \frac{1}{\epsilon} C^T C\right)P + I + \epsilon I = 0 \quad (2.3.7)$$

possède une solution symétrique, définie positive  $P$ , alors le gain de l'observateur

$$K = \frac{1}{2\epsilon} P C^T \quad (2.3.8)$$

stabilise la dynamique de l'erreur (2.3.5) pour tout  $f$  avec une constante de Lipschitz  $L$ . Cependant l'algorithme ne réussit pas pour toute paire  $(A, C)$  observable. Dans [52], cet algorithme est utilisé pour la détection des défaillances.

Un autre algorithme pour trouver la matrice de gain  $K$  pour assurer la stabilité asymptotique de l'observateur est donné dans [112] basé en termes des valeurs propres et des vecteurs propres de la matrice  $A - KC$ , où les valeurs propres doivent être suffisamment négatives et les vecteurs propres doivent être suffisamment bien conditionnés pour assurer la stabilité. L'algorithme essaye de réduire au minimum le rapport  $[\lambda_{\min}(A - KC)/M_2(T)]$  où  $M_2(T)$  est le conditionnement de la matrice des vecteurs propres de  $A - KC$ . Malheureusement l'algorithme peut seulement trouver des minima



locaux.

Dans [113] le concept de *distance à la non-observabilité* de la paire  $(A, C)$  donnée par l'équation

$$\delta(A, C) = \min_{\omega} \sigma_{\min} \begin{bmatrix} j\omega I - A \\ C \end{bmatrix} ; \quad \omega \in \mathbb{R} \quad (2.3.9)$$

(où  $\sigma_{\min}$  est la valeur singulière la plus petite), est utilisée pour indiquer qu'il existe une matrice de gain  $K$  telle que l'équation de l'erreur (2.3.5) soit asymptotiquement stable si  $\delta(A, C) > L$ , alors la matrice de gain  $K$  est calculée à partir de

$$K = \frac{P^{-1}C^T}{2L^2} \quad (2.3.10)$$

où  $P$  est la solution de l'équation de Riccati

$$A^T P + PA + L^2 P P + I - \frac{C^T C}{L^2} + \epsilon I = 0 \quad (2.3.11)$$

avec  $\epsilon > 0$  suffisamment petit. Cependant, dans [1], il est montré que  $\delta(A, C)$  n'est pas égale à la distance de la non-observabilité, mais plutôt à la distance à la paire la plus proche avec un mode purement imaginaire non observable. De nouvelles hypothèses sur la convergence sont données ainsi qu'un algorithme pour trouver  $\delta(A, C)$ , pour assurer la stabilité. La matrice de gain  $K$  est calculée à partir de

$$K = \frac{1}{2} P^{-1} C^T \quad (2.3.12)$$

où  $P$  est la solution de l'équation de Riccati

$$A^T P + PA + PP + (L^2 + \epsilon)I - C^T C = 0 \quad (2.3.13)$$

avec  $\epsilon > 0$  suffisamment petit.

Dans [136], les systèmes considérés sont de la forme (2.0.1), l'observateur est donné par l'équation dynamique

$$\dot{\hat{x}} = f(\hat{x}, u) + K(u)(y - C\hat{x}) \quad (2.3.14)$$

et l'équation dynamique de l'erreur est

$$\dot{e} = f(x, u) - f(\hat{x}, u) - K(u)Ce \quad (2.3.15)$$

La matrice de gain  $K(u)$  donnée par

$$K(u) = \alpha \beta(u) P^{-1} C^T \quad (2.3.16)$$

où  $\alpha$  est une constante positive suffisamment grande, et  $\beta$  et  $P$  sont choisis pour satisfaire

$$\mu^T P \frac{\partial f(x, u)}{\partial x} \mu \leq -s_1 \|\mu\|^2 \quad (2.3.17)$$

$$\begin{aligned} \left| \nu^T P \frac{\partial f(x, u)}{\partial x} \nu \right| &\leq \beta(u) \|\nu\|^2 \\ s_2 &\leq \beta(u) \end{aligned}$$

pour certaines constantes positives  $s_1, s_2$  et pour tout  $x, u, \nu \in \mathbb{R}^n$  et tout  $\mu \in \mathbb{R}^n$  dans un voisinage de tout vecteur dans le noyau de  $C$ , assure la stabilité asymptotique de l'observateur. La généralisation est présentée dans [137], où le gain est de la forme  $K(\hat{x}, u)$  et cet algorithme est utilisé dans [2] pour la détection de défaillances. Cependant, en général, la construction de la fonction de Lyapunov proposée dans [136], [137] est assez difficile.

## Commentaires sur l'approche

### *Avantages*

- La mise en œuvre de cet observateur est simple quand on peut trouver une matrice de gain  $K$  qui garantit que la dynamique de l'erreur est stable.

### *Inconvénients*

- Le système considéré doit avoir une structure particulière. Choisir la matrice de gain  $K$  pour satisfaire la condition (2.3.6) est un processus d'essai-erreur qui devient, pour des systèmes d'ordre élevé pratiquement impossible à réaliser.
- Cette méthode ne permet pas de traiter explicitement le bruit ou les incertitudes de modèle. Sa robustesse n'est qu'une conséquence directe de l'existence d'une fonction de Lyapunov.

### 2.3.3 Observateurs pour systèmes sous une forme canonique

La synthèse d'observateurs non linéaires s'est beaucoup inspirée des méthodes issues des travaux sur les systèmes linéaires. En particulier ont été proposés des observateurs pour les systèmes présentant une forme canonique. Cette forme est basée sur une structure linéaire modulo injection d'Entrées/Sorties et d'un certain nombre de leurs dérivées [27; 76; 80; 106].

Le problème est que souvent, un système non linéaire est donné sous une forme peu appropriée à une synthèse facile d'observateur. Pour cette raison, beaucoup de techniques de synthèse appliquent *une transformation à travers un changement de coordonnées* qui fait apparaître une forme particulière répondant aux conditions de synthèse. Ceci permet de définir une relation d'équivalence entre systèmes

**Définition 2.3.1.** [*Relation d'équivalence*][24]

Soit  $x_0 \in \mathbb{R}^n$ . Un système décrit par les équations

$$\begin{aligned}\dot{x} &= f(x, u) = f_u(x) \\ y &= h(x)\end{aligned}$$

est dit équivalent en  $x_0$  au système suivant

$$\begin{aligned}\dot{x}^* &= F(x^*, u) = F_u(x^*) \\ y &= H(x^*)\end{aligned}$$

s'il existe un difféomorphisme  $x^* = \Phi(x)$  sur un voisinage de  $x_0$  tel que

$$\forall u \in \mathbb{R}^m, \quad \left. \frac{\partial \Phi}{\partial x} f_u(x) \right|_{x=\Phi^{-1}(x^*)} = F_u(x^*) \quad \text{et} \quad h \circ \Phi^{-1} = H$$



Ainsi, un système peut se montrer plus aisé à observer sous une représentation équivalente. Aussi, si pour un système donné l'observateur est déterminé, alors pour tout système équivalent un observateur peut être déduit. Ce résultat est présenté dans la proposition suivante

**Proposition 2.3.1.** [24]

Étant donnés deux systèmes respectivement définis par

$$\begin{aligned}\dot{x} &= X(x, u) \\ y &= h(x)\end{aligned} \tag{2.3.18}$$

et

$$\begin{aligned}\dot{x}^* &= X^*(x^*, u) \\ y &= h(x^*)\end{aligned} \tag{2.3.19}$$

et équivalents par le changement de coordonnées  $x^* = \Phi(x)$ , si

$$\begin{aligned}\dot{\hat{x}}^* &= X^*(\hat{x}^*, u) - k(y - H(\hat{x}^*), z) \\ \dot{z} &= Z(z, u, y)\end{aligned} \tag{2.3.20}$$

est un observateur pour (2.3.19), alors

$$\begin{aligned}\dot{\hat{x}} &= X(\hat{x}, u) - \left( \frac{\partial \Phi}{\partial x} \right)_{\hat{x}}^{-1} k(y - h(\hat{x}), z) \\ \dot{z} &= Z(z, u, y)\end{aligned} \tag{2.3.21}$$

est un observateur pour (2.3.18). Si  $\Phi$  et  $\Phi^{-1}$  préservent les métriques de  $\mathbb{R}^n$



En outre, une telle équivalence de systèmes peut être étendue par la prise en compte d'injections de sortie, voire de transformations de sortie, qui permettent de plus d'augmenter les classes de systèmes auxquelles peuvent s'appliquer des observateurs connus. La difficulté provient alors de la détection des systèmes équivalents ayant une telle forme, et c'est pourquoi leurs caractérisations ont été largement explorées [24; 81; 87; 107; 146; 147].

Le problème de la caractérisation des classes de systèmes pour lesquelles un observateur peut être construit est apparu pour la première fois en 1983, sous forme de conditions de linéarisation par difféomorphisme et injection de sortie pour l'observation, comme problème dual à celui de linéarisation par difféomorphisme et retour d'état de la commande. Dans cette même année, deux résultats dans ce domaine ont émergé : l'un proposé dans [80] utilise une linéarisation exacte de l'estimation de la dynamique de l'erreur. Cette voie est basée sur une étude de la structure interne du problème en utilisant des outils géométriques. L'autre proposée dans [27] utilise une linéarisation approchée de l'estimation de la dynamique de l'erreur, dans lequel les non-linéarités du système dépendent seulement de l'entrée (ou de l'entrée avec un certain nombre de ses dérivées) et de la sortie du système original.

Plus spécifiquement, le système non linéaire de la forme

$$\begin{aligned} \dot{x} &= f(x) \\ y &= h(x) \end{aligned} \quad (2.3.22)$$

est considéré comme équivalent modulo une injection de sortie à un système linéaire autonome s'il existe une transformation de coordonnées

$$x = T(x^*) \quad (2.3.23)$$

telle que le système dans les nouvelles coordonnées soit donné sous la forme suivante

$$\begin{aligned} \dot{x}^* &= A_c x^* + \phi(y) \\ y &= C_c x^* \end{aligned} \quad (2.3.24)$$

cette transformation est intéressante lorsque la paire  $(C_c, A_c)$  est une paire observable alors que  $\phi$  est une fonction arbitraire de  $y$ .

Si la paire  $(C_c, A_c)$  est sous forme canonique (duale de la forme de commandabilité de Brunovsky) alors le système (2.3.24) est sous la forme canonique d'observabilité

$$\begin{aligned} \dot{x}^* &= \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ 1 & & & \vdots \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix} x^* + \begin{bmatrix} a_1(y) \\ \vdots \\ \vdots \\ a_n(y) \end{bmatrix} \\ y &= \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} x^* \end{aligned} \quad (2.3.25)$$

Si  $\phi(y)$  et la transformation  $x = T(x^*)$  peuvent être trouvées, alors un observateur pour le système (2.3.24) peut être construit presque aussi facilement que pour un système linéaire autonome, c'est à dire, un observateur de la forme suivante est obtenu

$$\begin{aligned}\dot{\hat{x}}^* &= A_c \hat{x}^* + \phi(y) + K(y - \hat{y}) \\ \hat{y} &= C_c \hat{x}^*\end{aligned}\quad (2.3.26)$$

avec la dynamique linéaire de l'erreur  $\dot{e} = (A - KC)e$ . De cette façon, si (2.3.26) est réécrite dans les coordonnées d'origine, un observateur pour le système (2.3.22) est obtenu en temps fini.

Dans [80], des conditions nécessaires et suffisantes, basées sur l'algèbre de Lie (pour les rappels des définitions des objets et opérateurs différentiels voir l'Annexe A), sont données pour l'observabilité des systèmes non linéaires et pour trouver des observateurs. Ces conditions montrent quand un système non linéaire autonome (2.0.9) peut être transformé en un système linéaire en utilisant une transformation de coordonnées modulo une injection de sortie. Le résultat suivant caractérise cette classe de systèmes non linéaires.

**Théorème 2.3.2.** [80] *Il existe un changement de coordonnées qui transforme le système non linéaire (2.3.22) localement autour de  $x^0$  dans le système linéaire (2.3.25) si et seulement si les conditions suivantes sont satisfaites dans un voisinage  $V$  de  $x^0$*

1.  $\dim[\text{span}\{dh(x), \dots, dL_f^{n-1}h(x)\}] = n, \quad \forall x \in V$
2. *Le champ de vecteur  $v(x)$  défini sur  $V$  par*

$$L_v L_f^j h(x) = \begin{cases} 0, & j = 0, \dots, n-2 \\ 1, & j = n-1 \end{cases} \quad (2.3.27)$$

*satisfait la relation*

$$[v, ad_f^i v](x) = 0 \quad ; \quad i = 1, \dots, 2n-1 \quad \forall x \in V \quad (2.3.28)$$

♠

Le Théorème 2.3.2 indique que la linéarisation de l'observateur est possible si et seulement si

$$\dim[\text{span}\{dh(x), \dots, dL_f^{n-1}h(x)\}] = n$$

ainsi que si un vecteur unique  $v(x)$  peut être trouvé dans le voisinage  $V$  de  $x^0$  tel que (2.3.27) et

$$\frac{\partial T}{\partial x^*} = [v(x) \quad -ad_f v(x) \quad \dots \quad (-1)^{n-1} ad_f^{n-1} v(x)]$$

alors la transformation de coordonnées existe si et seulement si

$$[v, ad_f^i v](x) = 0 \quad ; \quad i \leq 2n-1$$

La synthèse de l'observateur est identique à celle pour le cas d'un système linéaire, si la transformation peut être trouvée. Cependant l'équation (2.3.28) est une condition restrictive. Donc la classe de systèmes non linéaires qui peuvent être ramenés à une forme canonique en utilisant cette technique est plutôt réduite. Le cas multi-sortie a été étudié dans [81] et étendu dans [64; 147].

Dans [140], en employant les arguments donnés par [27], il est montré que la transformation  $x = T(x^*)$  dans le cas mono-sortie peut être déterminée, sous quelques conditions, de la manière suivante :

En utilisant l'équation de la sortie

$$y = h(x) = x_n^* \quad (2.3.29)$$

et en prenant la dérivée partielle de (2.3.29) par rapport à  $x^*$

$$\frac{\partial h(x)}{\partial x} \frac{\partial T}{\partial x^*} = [0 \ 0 \ \dots \ 0 \ 1] \quad (2.3.30)$$

qui s'écrit en termes du vecteur  $\partial T / \partial x_1^*$  par

$$\begin{bmatrix} L_f^0(dh)(x) \\ L_f^1(dh)(x) \\ \vdots \\ L_f^{n-1}(dh)(x) \end{bmatrix} \frac{\partial T}{\partial x_1^*} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (2.3.31)$$

où la matrice

$$O(x) = \begin{bmatrix} L_f^0(dh)(x) \\ L_f^1(dh)(x) \\ \vdots \\ L_f^{n-1}(dh)(x) \end{bmatrix}$$

est la matrice d'observabilité (qui est non singulière) du système (2.3.22). Donc  $\partial T / \partial x_1^*$  peut être déterminé à partir de la matrice  $O^{-1}(x)$ .

La dérivée de  $x = T(x^*)$  par rapport au temps est

$$\dot{x} = \frac{\partial T}{\partial x^*} f^*(x^*) \quad (2.3.32)$$

Toutes les colonnes de  $\partial T / \partial x^*$  peuvent être exprimées en termes du vecteur  $\partial T / \partial x_1^*$  de la manière suivante

$$\frac{\partial T}{\partial x^*} = \left[ \left( ad^0 f, \frac{\partial T}{\partial x_1^*} \right) \left( ad^1 f, \frac{\partial T}{\partial x_1^*} \right) \dots \left( ad^{n-1} f, \frac{\partial T}{\partial x_1^*} \right) \right] \quad (2.3.33)$$

Ainsi, la transformation  $T(x^*)$  peut être déterminée par intégration de la matrice Jacobienne (2.3.33)<sup>1</sup>.

La généralisation de [27] au cas multi-sortie est faite dans [28], [149]. Le cas multi-sortie est traité d'une façon similaire, la différence est l'utilisation de blocs de matrices, chacune correspondant à une sortie particulière. Cependant dans ce cas, la matrice de gain  $K$  pour construire l'observateur n'est pas unique.

Dans [105], il est montré que pour des systèmes autonomes (2.0.9) ayant une dimension supérieure à 2, calculer la transformation pour amener le système non linéaire à une forme canonique observable est de plus en plus compliquée. La difficulté particulière des approches proposées dans [80] et [27] est leur exigence de calculs lourds de crochets de Lie et surtout, le problème de la résolution des équations résultantes.

Dans [76] ces résultats ont été étendus par transformation d'état généralisé, à la linéarisation modulo injection de sortie, de l'entrée et de ses dérivées. Pour ce cas, la forme canonique généralisée de l'observateur est définie par

$$\begin{aligned} \dot{x}^* &= A_c x^* + \phi(y, \tilde{u}) \\ y^* &= C_c x^* \end{aligned} \quad (2.3.34)$$

où

$$\tilde{u} = [u \quad \dot{u} \quad \ddot{u} \quad \cdots \quad u^{(n)}]$$

la transformation est généralisée à

$$\begin{aligned} x^* &= T(x, \bar{u}) \\ y^* &= V(y, u) \end{aligned}$$

où

$$\bar{u} = [u \quad \dot{u} \quad \ddot{u} \quad \cdots \quad u^{(n-1)}]$$

et la structure de l'observateur est

$$\begin{aligned} \dot{\hat{x}}^* &= A_c \hat{x}^* + \phi(y, \tilde{u}) + K(y^* - \hat{y}^*) \\ \hat{y}^* &= C_c \hat{x}^* \end{aligned} \quad (2.3.35)$$

L'inconvénient de cette approche est la nécessité de résoudre un certain nombre d'équations différentielles aux dérivées partielles pour trouver la transformation. Pour un système du deuxième ordre, il est nécessaire de résoudre trois équations différentielles aux dérivées partielles, pour un système du troisième ordre les équations à résoudre sont six et ainsi de suite. Avec l'augmentation de l'ordre du système, le nombre d'équations différentielles aux dérivées partielles augmente rapidement, laissant cette approche envisageable seulement pour des systèmes de dimension faible.

Le problème de la réduction de l'ordre de dérivation des entrées a ensuite été

---

1. L'intégration d'une matrice  $A(t) = [a_{ij}(t)]$  est définie par  $\int A(t)dt = [\int a_{ij}(t)dt]$

abordé dans [43], [110]. Des conditions nécessaires et suffisantes pour l'existence de la transformation de coordonnées de l'état et de la sortie pour amener un système de la forme (2.0.1) à la forme canonique (2.3.34) pour les systèmes mono-sortie sont données dans [58] en termes de formes différentielles ainsi qu'un algorithme pour accomplir les transformations de l'état et de la sortie. Dans [108] les résultats sont généralisés pour des systèmes multi-sorties.

Dans [75], l'observateur pour un système autonome (2.0.9) est conçu par une transformation de coordonnées appropriée sans l'exigence de la linéarité de la sortie. En effet, l'application de la sortie transformée est en général non linéaire. Le système (2.0.9) est transformé en utilisant la transformation (2.3.23) dans le système

$$\begin{aligned}\dot{x}^* &= Ax^* + by \\ y &= h(T^{-1}(x^*))\end{aligned}\tag{2.3.36}$$

et l'observateur est

$$\dot{\hat{x}}^* = A\hat{x}^* + by\tag{2.3.37}$$

Bien que l'observateur n'inclue pas de bouclage explicite de l'erreur de la sortie, la dynamique de l'erreur  $\dot{e} = Ae$  peut être rendue arbitrairement rapide en choisissant la transformation  $T(x)$  de sorte que la matrice  $A$  ait des propriétés appropriées.

L'inconvénient principal de cette approche est la difficulté de satisfaire l'hypothèse sur la matrice Jacobienne de  $f(x)$  en  $x = 0$ , à savoir

$$F = \frac{\partial f}{\partial x}(0)$$

doit posséder toutes les valeurs propres strictement dans la moitié gauche du plan ou dans la moitié droite du plan. Cette condition est allégée dans [82] et la méthode est généralisée aux systèmes multi-sorties.

Pour élargir la classe de systèmes qui peuvent être traités par transformation dans la forme canonique observable, plusieurs méthodes pour résoudre approximativement le problème de la transformation ont été proposées.

Dans [98], [99], [117] une forme canonique approchée (appelée *pseudo-linéarisation*) qui combine linéarisation autour d'un point de fonctionnement et transformation d'état est présentée. L'idée est que quand cette transformation ne peut pas être accomplie exactement, elle est obtenue en négligeant les termes du deuxième ordre et supérieurs de leurs développements de Taylor autour des points de fonctionnement. Les systèmes non linéaires considérés sont de la forme (2.0.1) et l'équation dynamique de l'observateur est donnée par

$$\begin{aligned}\dot{\hat{x}}^* &= A_c\hat{x}^* + \phi(y, u) + K(y - \hat{y}) \\ \hat{y} &= C_c\hat{x}^*\end{aligned}\tag{2.3.38}$$

qui est égale à l'observateur (2.3.26) pour des systèmes autonomes (2.0.9) obtenu par [80].



L'inconvénient principal de cet approche est la nécessité d'une connaissance parfaite du système afin d'établir la transformation.

En transformant un système autonome (2.0.9) qui ne remplit pas les conditions de la transformation dans la forme canonique observable (la condition d'intégrabilité peut échouer), le système transformé est

$$\begin{aligned}\dot{x}^* &= A_c x^* + \phi(y) + \eta(x^*) \\ y &= C_c x^*\end{aligned}\tag{2.3.39}$$

où le terme  $\eta(x^*)$  représente les non linéarités qui ne peuvent pas être annulées. Dans [90] la transformation est réalisée afin de minimiser ce terme des non linéarités. L'extension au cas multi-sortie pour des systèmes affines en l'entrée (2.0.7) est développée dans [91]. D'autres méthodes pour se rapprocher de la forme canonique observable peuvent être trouvées dans [45], [97].

Une généralisation de la forme canonique observable est la forme affine en l'état. Dans [59], des conditions pour l'existence d'une transformation du système (2.0.1) à la forme affine en l'état sont données. Il est également proposé qu'un filtre de Kalman soit utilisé comme un observateur pour cette forme puisque celle-ci peut être considérée comme un système linéaire, temps-variant. La forme affine de l'état est généralisée dans [57].

## Commentaires sur l'approche

### *Avantages*

- Cette synthèse d'observateur se réduit à un simple observateur linéaire dès que la transformation est effectuée.

### *Inconvénients*

- Il n'est pas toujours possible de trouver une transformation qui met le système dans une forme canonique.
- Difficulté en général de détecter des systèmes équivalents avec une telle forme canonique.
- L'approche de linéarisation est basée sur un ensemble de conditions extrêmement restrictives, qui sont rarement satisfaites dans la pratique.
- La robustesse n'est pas assurée à cause des dérivations successives de  $f$  et de  $h$  pour trouver les transformations nécessaires.

### 2.3.4 Les observateurs dits "*à grand gain*"

Comme il a été dit précédemment, l'observabilité des systèmes non linéaires, en général, est dépendante de l'entrée. Cependant, il existe des systèmes qui sont observables

quelque soit l'entrée. Une telle classe de systèmes est appelée uniformément observable. Dans [53], il est montré que les systèmes mono-sortie affines en la commande qui sont uniformément observables peuvent être transformés localement par un changement approprié de coordonnées dans une forme canonique qui est composée d'une partie observable linéaire et d'une partie commandée non linéaire qui possède une structure triangulaire.

Dans [54] (et les travaux proposés par [44] et [134]), un observateur à grand gain a été donné pour de tels systèmes sous des hypothèses du type Lipschitz sur les non linéarités. Plus spécifiquement, un observateur à grand gain a été proposé pour la classe de systèmes de la forme

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \quad (2.3.40)$$

pour lesquelles l'application  $\Phi$  définie par

$$\Phi = \begin{pmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{pmatrix} \quad (2.3.41)$$

constitue un difféomorphisme qui transforme le système (2.3.40) en

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_n \\ \varphi(x) \end{bmatrix} + \begin{bmatrix} g_1(x_1) \\ g_2(x_1, x_2) \\ \vdots \\ g_{n-1}(x_1, \dots, x_{n-1}) \\ g_n(x_1, \dots, x_n) \end{bmatrix} u \\ &= F(x) + G(x)u \\ y &= x_1 = Cx \end{aligned} \quad (2.3.42)$$

où chaque  $g_i(\bar{x}_i)$  ( $\bar{x}_i$  dénote le vecteur  $(x_1, \dots, x_i)^T$ ) est globalement Lipschitz et  $\varphi$  peut être étendu de  $\Omega$  (sous-ensemble sur lequel le système (2.3.40) est observable) pour tout  $\mathbb{R}^n$  par une fonction  $C^\infty$  globalement Lipschitz sur  $\mathbb{R}^n$ .

Le système (2.3.42) est supposé être uniformément observable. Le système dynamique

$$\dot{\hat{x}} = f(\hat{x}) + g(\hat{x})u - S_\infty^{-1}C^T(C\hat{x} - y) \quad (2.3.43)$$

constitue un observateur pour le système (2.3.42) où  $S_\infty$  est solution de

$$0 = -\theta S_\infty - A^T S_\infty - S_\infty A + C^T C$$

avec  $C = [1 \ 0 \ \dots \ 0]$ ,  $A$  définie par

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & \ddots & \vdots \\ 0 & \cdots & & 0 & 1 & \\ 0 & \cdots & & & & 0 \end{bmatrix}$$

et pour un  $\theta$  suffisamment grand qui permet d'ajuster la vitesse de convergence, c'est à dire

$$\|\hat{x}(t) - x(t)\| \leq K(\theta) \exp(-\theta t/3) \|\hat{x}_0 - x_0\| \quad (2.3.44)$$

Le gain de l'observateur est basé seulement sur la partie linéaire du système et l'effet de la non linéarité est rendue négligeable en choisissant une valeur suffisamment grande de  $\theta$  d'où le terme "*grand gain*" utilisé pour désigner cet observateur.

Une autre façon de représenter le système (2.3.40) après l'application du changement non-linéaire de coordonnées  $\Phi$  est

$$\begin{aligned} \dot{x} &= Ax + g(x, u) \\ y &= Cx \end{aligned} \quad (2.3.45)$$

où la matrice  $A$ , le vecteur  $C$  et la fonction  $g(x, u)$  possèdent la même structure décrite ci-dessus. L'observateur pour le système (2.3.45) est donné par

$$\dot{\hat{x}} = A\hat{x} + g(\hat{x}, u) - S_\infty^{-1}C^T(C\hat{x} - y) \quad (2.3.46)$$

dans lequel  $S_\infty$  est comme dans (2.3.43).

Dans [34], le gain constant  $S_\infty^{-1}C^T$  est décomposé comme  $S_\infty^{-1}C^T = \Delta_\theta^{-1}K$  où

$$\begin{aligned} K &= \text{col}\left(C_n^1, C_n^2, \dots, C_n^n\right) \\ C_n^p &= \frac{n!}{(n-p)!p!} \\ \Delta_\theta &= \text{diag}\left[\frac{1}{\theta}, \frac{1}{\theta^2}, \dots, \frac{1}{\theta^n}\right] \end{aligned}$$

En fait, ce n'est pas nécessaire que le vecteur  $K$  soit de la forme spécifique ci-dessus, car tout vecteur  $K$  tel que la matrice  $(A - KC)$  soit stable est convenable [31]. En conséquence, l'observateur (2.3.46) peut simplement être réécrit comme

$$\dot{\hat{x}} = A\hat{x} + g(\hat{x}, u) - \Delta_\theta^{-1}K(C\hat{x} - y) \quad (2.3.47)$$

Dans [33], le gain constant est remplacé par un gain pondéré qui est fonction de l'état estimé et de l'entrée, car l'observateur proposé incorpore une partie de la non linéarité du système  $g(x, u)$  en termes de sa matrice Jacobienne. Plus précisément, l'observateur est donné par

$$\dot{\hat{x}} = A\hat{x} + g(\hat{x}, u) - M^{-1}(\hat{x}, u)[L(\hat{x}, u) + \Delta_\theta^{-1}K](C\hat{x} - y) \quad (2.3.48)$$

où la matrice  $\Delta_\theta$  et le vecteur  $K$  ont la même signification qu'auparavant et la matrice  $M(\hat{x}, u)$  de dimension  $(n \times n)$  et le vecteur  $L(\hat{x}, u)$  de dimension  $(n \times 1)$  sont fonctions de la dérivée partielle de  $g$  par rapport à  $x$  évaluées en  $\hat{x}$ .

Il a été montré que l'observateur (2.3.48) présente moins d'oscillations transitoires et semble être plus robuste vis à vis du bruit de mesure que l'observateur (2.3.43). Néanmoins, la conception de l'observateur (2.3.48) suppose une connaissance parfaite du modèle du système, car les non linéarités sont utilisées explicitement dans l'observateur. Ainsi, cet approche est très sensible aux erreurs de modèle.

La synthèse d'observateurs à grand gain pour les systèmes généraux (2.0.1) est montrée dans [55]. La transformation de ces systèmes mène à

$$\left. \begin{array}{l} \dot{x} = \psi(x, u) \\ y = \psi_0(x, u) \end{array} \right\} = F(x, u) \quad (2.3.49)$$

où  $\psi(x, u)$  possède la structure triangulaire

$$\psi(x, u) = \begin{bmatrix} \psi_1(x_1, x_2, u) \\ \psi_2(x_1, x_2, x_3, u) \\ \vdots \\ \psi_n(x, u) \end{bmatrix} \quad (2.3.50)$$

Pour le système (2.3.49), l'observateur est donné par

$$\dot{\hat{x}} = F(\hat{x}, u) - \Delta_\theta^{-1}K(\psi_0(\hat{x}, u) - y) \quad (2.3.51)$$

Un autre observateur pour cette structure est présenté dans [35] et une généralisation de la transformation aux systèmes multi-sorties est trouvée dans [63]. La détection de défaillances pour de systèmes affine dans l'entrée avec une forme triangulaire est considérée dans [60], [104].

Dans [38], un autre type d'observateur exponentiel est proposé. La différence par rapport à l'observateur proposé dans [54] est qu'après la transformation non linéaire, l'observateur dans les nouvelles coordonnées n'a pas besoin d'avoir une structure triangulaire, cependant, il faut que l'observateur possède un degré relatif  $n$ . Plus précisément, le système (2.3.40) est transformé par l'application (2.3.41) en

$$\dot{x}^* = Ax^* + B[\phi(x^*) + \gamma(x^*)u] \quad (2.3.52)$$

$$y = Cx^*$$

où  $A$  et  $C$  sont définis comme auparavant,  $B = [0 \ 0 \ \dots \ 1]^T$  et

$$\begin{aligned}\phi(x^*) &= L_f^n h(x) \Big|_{x=\Phi^{-1}(x^*)} \\ \gamma(x^*) &= L_g L_f^{n-1} h(x) \Big|_{x=\Phi^{-1}(x^*)}\end{aligned}$$

L'observateur dans ces coordonnées est

$$\dot{\hat{x}}^* = A\hat{x}^* + B[\phi(\hat{x}^*) + \gamma(\hat{x}^*)u] - K(C\hat{x}^* - y) \quad (2.3.53)$$

En retournant au système de coordonnées original, l'observateur est représenté par l'équation dynamique

$$\dot{\hat{x}} = f(\hat{x}) + g(\hat{x})u - [Q(\hat{x})]^{-1}K(C\hat{x} - y) \quad (2.3.54)$$

où

$$Q(x) = \frac{d}{dx}\Phi(x)$$

L'avantage de cette approche par rapport à l'approche proposée par [54] est qu'une structure triangulaire n'est pas nécessaire pour la synthèse de l'observateur. Cependant, si le système possède un degré relatif inférieur à  $n$ , l'observateur requiert des dérivées de l'entrée  $u$ . L'observateur est alors augmenté avec une partie qui estime ces dérivées. Aussi, des hypothèses d'existence de constantes de type Hölder associées à la matrice  $Q(x)$  doivent être satisfaites.

Dans [41] l'extension au cas multi-sortie est réalisée et le cas où le degré relatif n'est pas bien défini est traité dans [71]<sup>2</sup>.

## Commentaires sur l'approche

### *Avantages*

- Cet observateur donne une réponse exponentielle aussi rapide que souhaité en augmentant  $\theta$ .

### *Inconvénients*

- Difficulté d'obtenir une structure triangulaire même dans le cas où une telle structure est théoriquement accessible par une transformation de coordonnées.
- Sensibilité au bruit de mesure si le gain est choisi trop grand.

---

2. Ces deux extensions sont des cas très particuliers de [31]

### 2.3.5 Observateurs adaptatifs

Un objectif dans beaucoup d'applications pratiques est de combiner la connaissance a priori au sujet d'un système physique avec des données expérimentales mesurables pour fournir l'évaluation en ligne des états ou des paramètres de ce système. La combinaison de cette connaissance a priori avec les mesures en ligne de l'entrée et de la sortie a permis de concevoir des observateurs adaptatifs pour l'estimation des états ou des paramètres. Les premiers observateurs adaptatifs stables pour des systèmes linéaires temps-invariant ont été proposés dans [78; 79; 83; 84; 85].

Dans [15], un observateur adaptatif non linéaire pour des systèmes mono-entrée/mono-sortie qui peuvent être transformés dans une certaine forme canonique observable a été donné en utilisant les concepts développés dans [84]. Plus spécifiquement, un observateur adaptatif non linéaire a été proposé pour la classe de systèmes donnée par

$$\begin{aligned} \dot{x} &= f(x, u, p) \\ y &= x_1 \end{aligned} \quad (2.3.55)$$

où  $x \in \mathbb{R}^n$  est le vecteur d'état,  $u \in \mathbb{R}$  est l'entrée de commande mesurée,  $y \in \mathbb{R}$  est la sortie mesurée et  $p \in \mathbb{R}^q$  est le vecteur de paramètres inconnus, bornés et temps-variant.

L'idée est de transformer le système non linéaire (2.3.55) dans une forme canonique observable en utilisant une transformation non linéaire de  $(x, p)$  à  $(\xi, \theta)$  paramétrée par  $(n - 1)$  paramètres  $(c_2, \dots, c_n)$ , plus précisément

$$\begin{bmatrix} \xi \\ \theta \end{bmatrix} = T(x, p, c_2, \dots, c_n) \quad ; \quad T(\cdot) \in \mathbb{R}^{n+m} \quad (2.3.56)$$

de telle sorte que les équations du système dans les nouvelles coordonnées soient données par

$$\begin{aligned} \dot{\xi} &= R\xi + \Omega[\omega(t)]\theta(t) + g(t) \\ y &= \xi_1 \end{aligned} \quad (2.3.57)$$

où  $\xi \in \mathbb{R}^n$  est le vecteur d'état de la même dimension que  $x$ ,  $\theta \in \mathbb{R}^m$  est le vecteur de paramètres inconnus, temps-variant qui sont estimés en ligne,  $\omega \in \mathbb{R}^s$  est un vecteur de fonctions connues de  $u(t)$  et  $y(t)$ ,  $\Omega[\omega(t)]$  est une matrice de dimension  $(n \times m)$  dont les éléments sont tous de la forme  $\Omega_{ij}[\omega(t)] = \alpha_{ij}^T \omega(t)$  où  $\alpha_{ij} \in \mathbb{R}^s$  sont des constantes connues et  $g(t)$  est un vecteur de fonctions connues du temps. Enfin,  $R$  est une matrice constante connue de dimension  $(n \times n)$  de la forme suivante

$$R = \begin{bmatrix} 0 & | & k^T \\ \text{---} & | & \text{---} \\ 0 & | & \\ \vdots & | & F(c_2, \dots, c_n) \\ 0 & | & \end{bmatrix}$$

$$k^T = [k_2, \dots, k_n] \quad (2.3.58)$$

où  $(k_2, \dots, k_n)$  sont des constantes connues et  $F(c_2, \dots, c_n)$  est une matrice constante de dimension  $(n-1) \times (n-1)$  dont les valeurs propres peuvent être choisies librement par un choix propre des paramètres constants  $(c_2, \dots, c_n)$ . Typiquement,  $F = \text{diag}(-c_2, \dots, -c_n)$  avec  $c_i > 0$ .

La forme canonique observable "*adaptive*" (2.3.57) est linéaire dans les quantités inconnues  $\xi$  et  $\theta$  alors que  $\Omega[\omega(t)]$  est une fonction connue de  $u(t)$  et  $y(t)$  qui peut être non linéaire. L'observateur adaptatif suivant peut alors être appliqué

$$\begin{aligned} \dot{\hat{\xi}}(t) &= R\hat{\xi}(t) + \Omega[\omega(t)]\hat{\theta}(t) + g(t) + \begin{bmatrix} c_1\tilde{y}(t) \\ V(t)\dot{\hat{\theta}}(t) \end{bmatrix} \\ \hat{y}(t) &= \hat{\xi}_1 \quad ; \quad \tilde{y}(t) = y(t) - \hat{y}(t) \end{aligned} \quad (2.3.59)$$

où  $c_1$  est une constante positive. Les estimés des paramètres sont donnés par

$$\dot{\hat{\theta}}(t) = \Gamma\phi(t)\tilde{y}(t) \quad (2.3.60)$$

où  $\tilde{y}(t)$  est l'erreur d'estimation de la sortie, et  $\Gamma$  est une matrice définie positive normalement choisie telle que  $\Gamma = \text{diag}(\gamma_1, \dots, \gamma_m)$  avec  $\gamma_i > 0$ .  $\phi(t)$  est le vecteur, communément appelé, "*régresseur*" qui contient l'information mesurée sur les entrées et les sorties du processus.

De plus, un filtre auxiliaire est introduit pour générer le vecteur  $\phi(t)$  à l'issue des mesures actuelles. Les équations du filtre sont données par

$$\begin{aligned} \dot{V}(t) &= FV(t) + \bar{\Omega}[\omega(t)] \\ \phi(t) &= V^T(t)k + \Omega_1^T[\omega(t)] \\ V(0) &= 0 \end{aligned} \quad (2.3.61)$$

où  $\Omega_1$  est la première ligne de  $\Omega(\omega(t))$  et  $\bar{\Omega}$  les lignes restantes, à savoir,

$$\Omega = \begin{bmatrix} \Omega_1 \\ \bar{\Omega} \end{bmatrix}$$

$V(t)$  est une matrice de dimension  $(n-1) \times m$  et  $\phi(t)$  est un vecteur de dimension  $m$ .

L'observateur adaptatif (2.3.59) fournit, sous l'hypothèse de la persistance des excitations, des estimations asymptotiquement convergentes pour les variables d'état et aussi pour le vecteur de paramètres  $\theta$ .

D'autres structures pour la forme canonique observable *adaptive* sont données dans [92; 93; 94]. Dans [93] l'observateur adaptatif obtenu fournit des estimations asymptotiquement convergentes pour les états équivalentes même sans la persistance d'excitation. Dans [94], basé sur les résultats de [78], l'observateur adaptatif possède

la propriété de garantir un taux arbitraire de convergence exponentielle pour les erreurs des états et des paramètres si la persistance d'excitation est satisfaite. Dans [92], l'observateur adaptatif est conçu pour prendre en compte de façon explicite les perturbations. Cependant tous ces résultats sont limités aux systèmes non linéaires dont la dynamique peut être linéarisée par un changement de coordonnées avec injection de sortie. Leur applicabilité est limitée par la condition restrictive de linéarisation. Il s'agit de l'hypothèse qui rend tout le développement précédent possible. En fait, la synthèse effective de l'observateur nécessite plus que cette hypothèse. En effet, l'existence de ce changement de coordonnées linéarisante ne suffit pas à elle seule pour la synthèse; encore faut-il trouver ce changement de coordonnées.

Les approches proposées dans [25; 37; 114] pour la synthèse d'un observateur adaptatif ne nécessitent pas que le système non linéaire soit linéarisable. Ces approches supposent l'existence d'une certaine fonction de Lyapunov qui satisfait des conditions particulières. Ces approches ne sont pas des méthodes constructives dans le sens où il n'existe pas de manière systématique de concevoir la fonction de Lyapunov requise.

Une méthode constructive pour la synthèse des observateurs adaptatifs globaux pour une classe de systèmes non linéaires qui ne peuvent pas être linéarisés par un changement de coordonnées et injection de sortie est présenté dans [150]. L'idée est de combiner l'observateur adaptatif linéaire avec les observateurs à grand gain. Plus clairement, si un observateur à grand gain peut être conçu pour un système non linéaire donné, alors il est possible de trouver un observateur adaptatif pour le système obtenu en ajoutant dans l'équation d'état des termes complémentaires avec des coefficients inconnus.

La détection des défaillances a été étudiée en utilisant la forme canonique observable adaptative dans [51] et en utilisant une forme plus générale de l'observateur adaptatif dans [26].

## Commentaires sur l'approche

### *Avantages*

- L'observateur permet une estimation simultanée de l'état et des paramètres. Dans ce sens, il est en quelque sorte robuste vis à vis des variations paramétriques par rapport à un observateur d'état prenant les valeurs nominales des paramètres comme données.

### *Inconvénients*

- Les hypothèses sont difficiles à vérifier.
- Il est difficile de trouver la transformation linéarisante.



### 2.3.6 Linéarisation étendue

Cette méthodologie de synthèse développée par [17; 18; 19; 20] est une autre technique qui exploite les outils disponibles pour les systèmes linéaires. Cette technique tombe dans la catégorie de méthodes du type "*gain scheduling*". Plus précisément, les gains de l'observateur sont calculés de telle sorte que les valeurs propres de l'équation linéarisée de l'erreur soient placées à des valeurs spécifiques qui sont localement invariantes par rapport au point de fonctionnement. Cette approche a été proposée pour la classe des systèmes dynamiques décrits par l'équation

$$\begin{aligned}\dot{x} &= f(x, u) \\ y &= h(x)\end{aligned}\tag{2.3.62}$$

où  $f(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$  et  $h(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^p$  sont analytiques dans un voisinage de l'origine, avec  $f(0, 0) = 0$  et  $h(0) = 0$ .

Le système (2.3.62) est supposé avoir une famille de points d'équilibre dénotés par  $u = \epsilon$  et  $x = x_\epsilon = x(\epsilon)$ . On suppose de plus que l'inverse  $\left[\partial f / \partial x(0, 0)\right]^{-1}$  existe dans un voisinage de l'origine. Cette technique construit un observateur qui place les valeurs propres de l'équation différentielle de l'erreur linéarisée autour des points d'équilibre.

L'observateur obéit à l'équation différentielle

$$\dot{\hat{x}} = f(\hat{x}, u) + g(y) - g(\hat{y})\tag{2.3.63}$$

où  $\hat{y} = h(\hat{x})$  et  $g(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^n$  est analytique avec  $g(0) = 0$ .

En dénotant l'erreur entre l'état actuel et l'estimé par  $e = x - \hat{x}$ , l'équation dynamique de l'erreur devient

$$\dot{e} = f(x, u) - f(x - e, u) - g(y) + g(\hat{y})\tag{2.3.64}$$

Il peut être noté que si  $u = \epsilon$  alors  $x = x_\epsilon$  et  $\hat{y}_\epsilon = y_\epsilon = h(x_\epsilon)$  est un point d'équilibre pour (2.3.62) et (2.3.64). En faisant une linéarisation de (2.3.64) autour de ce point d'équilibre, l'équation linéarisée de l'erreur est

$$\dot{e} = [D_1 f(x_\epsilon, \epsilon) - Dg(y_\epsilon) Dh(x_\epsilon)]e\tag{2.3.65}$$

où

$$Dh(x) = \frac{\partial h(x)}{\partial x}$$

$$Dg(y) = \frac{\partial g(y)}{\partial y}$$

et

$$D_1 f(x, u) = \frac{\partial f(x, u)}{\partial x}$$

De plus, on suppose que

- Le système est complètement observable autour de zéro, c'est à dire, la paire  $[D_1f(0, 0), Dh(0)]$  est une paire observable.
- La condition suivante est satisfaite

$$Dy_\epsilon|_{\epsilon=0} = \frac{\partial y_\epsilon}{\partial \epsilon} \Big|_{\epsilon=0} = Dh(0)Dx_\epsilon(0) \neq 0$$

Le but pour la synthèse de cet observateur est de chercher une fonction analytique  $g(\cdot)$  telle que l'équation (2.3.65) possède des valeurs propres invariantes par rapport à  $\epsilon$ . La démarche pour trouver cette fonction analytique  $g(\cdot)$  est la suivante :

Il peut être remarqué que les valeurs propres de (2.3.65) sont les mêmes que les valeurs propres de

$$[D_1f(x_\epsilon, \epsilon)^T - Dh(x_\epsilon)^T Dg(y_\epsilon)^T]$$

La paire  $[D_1f(0, 0), Dh(0)]$  est supposée être complètement observable, donc la paire  $[D_1f(0, 0)^T, Dh(0)^T]$  est complètement contrôlable.

Soient  $A = D_1f(x_\epsilon, \epsilon)^T$ ,  $B = Dh(x_\epsilon)^T$  et  $K(s) = k_0 + k_1s + \dots + s^n$  le polynôme caractéristique désiré de (2.3.65). Une formule de placement de pôles est utilisée pour trouver une  $C(\epsilon)$  où  $C(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{p \times n}$  tel que  $[D_1f(x_\epsilon, \epsilon)^T - Dh(x_\epsilon)^T C(\epsilon)]$  ait des valeurs propres spécifiques qui soient invariantes par rapport à  $\epsilon$ . Ceci est possible s'il existe une fonction analytique  $g$  tel que l'équation aux dérivées partielles suivante soit satisfaite

$$Dg(y_\epsilon)^T = C(\epsilon) = l_1 \cdot K(A)$$

où  $l_1$  représente la dernière ligne de l'inverse de la matrice de contrôlabilité  $[B \ AB \ \dots \ A^{n-1}]$  et  $K(A) = k_0I + k_1A + \dots + A^n$  alors les valeurs propres de l'équation de l'erreur linéarisée (2.3.65) sont indépendantes de  $\epsilon$ .

Soit  $\hat{g}(\epsilon) = g(y_\epsilon)$  ce qui implique que

$$D\hat{g}(\epsilon) = \frac{\partial g}{\partial y_\epsilon} \frac{\partial y_\epsilon}{\partial \epsilon} = C^T(\epsilon)Dy_\epsilon \quad (2.3.66)$$

où

$$C^T(\epsilon) = [c_1(\epsilon) \ \dots \ c_p(\epsilon)]$$

et  $c_i(\epsilon)$  sont  $(n \times 1)$  vecteurs.

Comme le côté droit de (2.3.66) est connu ( $Dy_\epsilon$  peut être calculé), et en faisant l'intégration de (2.3.66) par rapport à  $\epsilon$ ,  $\hat{g}(\epsilon)$  peut être trouvé. Sans perte de généralité, le premier élément du vecteur  $Dy_\epsilon$  est supposé différent de zéro, c'est à dire  $Dy_{\epsilon 1} \neq 0$ , donc il existe une fonction inverse  $y_{\epsilon 1}^{-1}(\cdot)$  telle que  $y_{\epsilon 1}^{-1}(y_{\epsilon 1}) = \epsilon$ .

Ainsi, la fonction  $g(y)$  peut être résolue de la façon suivante :

$$g(y) = \hat{g}(y_{\epsilon 1}^{-1}(y_1)) + \sum_{j=2}^p c_j(y_{\epsilon 1}^{-1}(y_1)) [y_j - y_{\epsilon j}(y_{\epsilon 1}^{-1}(y_1))] \quad (2.3.67)$$

L'équation (2.3.67) implique que l'expression suivante est vraie

$$Dg(y) |_{y=y_\epsilon} = C^T(\epsilon) \quad (2.3.68)$$

La généralisation pour le cas de multiples entrées n'est pas complètement directe et la synthèse de l'observateur est plus compliquée car  $\epsilon$  est maintenant un vecteur de dimension  $m$  ( $m$  entrées sont supposées) ce qui implique la nécessité d'intégrer les  $m$  équations aux dérivées partielles (2.3.66) pour trouver la fonction  $g$ .

### Commentaires sur l'approche

#### *Avantages*

- Cette technique de synthèse donne une forme d'observateur simple si la fonction  $g$  peut être trouvée.

#### *Inconvénients*

- La synthèse de l'observateur pour le cas de multiples entrées est plus compliquée parce qu'on doit résoudre  $m$  équations aux dérivées partielles ( $m$  entrées) pour trouver la fonction  $g$ .
- Même pour le cas d'une entrée scalaire, trouver la fonction  $g$  est une tâche difficile, car on doit d'abord trouver tous les points d'équilibre (qui n'existent pas toujours), ensuite on doit intégrer  $n$  équations différentielles par rapport à  $\epsilon$  généralement difficiles à calculer (un calcul analytique lourd).
- D'autre part, même en présence des points d'équilibre, rien ne dit que les transitoires pendant lesquelles l'estimation de l'état est effectuée passent dans un voisinage de ces points d'équilibre. Cet observateur est en conséquence conçu pour opérer localement autour d'un point d'équilibre du système original.

### 2.3.7 Observateurs par modes glissants

Toutes les méthodes décrites jusqu'ici supposent que le processus est connu parfaitement. Une façon d'augmenter la robustesse contre les erreurs de modélisation et des incertitudes est d'utiliser des idées basées sur la théorie des systèmes à structure variable ou la théorie des modes glissants [68; 123; 138; 139].

Les premiers travaux pour obtenir des observateurs pour des systèmes dynamiques

incertains non linéaires utilisant la théorie des systèmes à structure variable ont été réalisés par [140; 141; 142]. La classe de systèmes étudiés est de la forme

$$\begin{aligned}\dot{x} &= Ax + f(x, t) \\ y &= Cx\end{aligned}\tag{2.3.69}$$

où  $A$  est une matrice connue constante de dimension  $(n \times n)$  et  $C$  est une matrice connue constante de dimension  $(p \times n)$  de rang plein. La fonction  $f(\cdot)$ , supposée continue en  $x$  est utilisée pour décrire les incertitudes non linéaires du système. Le système (2.3.69) doit satisfaire les hypothèses suivantes afin d'obtenir l'observateur.

1. La paire  $(A, C)$  est observable, donc il existe une matrice  $K$  de dimension  $(n \times p)$  telle que les valeurs propres de la matrice  $(A - KC)$  sont strictement négatives.
2. Il existe une fonction  $h(x, t)$  qui satisfait

$$f(x, t) = P^{-1}C^T h(x, t)\tag{2.3.70}$$

où  $P$  est une matrice constante symétrique définie positive de dimension  $(n \times n)$  solution de l'équation de Lyapunov

$$(A - KC)^T P + P(A - KC) = -Q\tag{2.3.71}$$

pour une matrice  $Q$  constante symétrique définie positive de dimension  $(n \times n)$ .

3. Il existe une fonction positive  $\rho(t)$  telle que

$$\|h(x, t)\| \leq \rho(t)\tag{2.3.72}$$

Sous les hypothèses précédentes, l'observateur proposé est donné par

$$\dot{\hat{x}} = (A - KC)\hat{x} + \kappa(\hat{x}, y, t) + Ky\tag{2.3.73}$$

où

$$\kappa(\hat{x}, y, t) = \begin{cases} \frac{P^{-1}C^T C(x - \hat{x})}{\|C(x - \hat{x})\|} \rho(t) & C(x - \hat{x}) \neq 0 \\ 0 & C(x - \hat{x}) = 0 \end{cases}$$

Le terme de discontinuité  $\kappa(\hat{x}, y, t)$  peut mener au phénomène connu en anglais sous le nom de "*chattering*" (oscillations haute fréquence). Des solutions existent pour éviter ce phénomène au prix de l'obtention d'une erreur d'estimation qui ne tend pas vers zéro.

Dans [42], l'observateur proposé pour le système (2.3.69) donné par

$$\dot{\hat{x}} = (A - KC)\hat{x} + Ky + \frac{P^{-1}C^T C(x - \hat{x})\rho^2}{\|C(x - \hat{x})\|\rho(t) + \epsilon \cdot \exp^{-\beta t}}\tag{2.3.74}$$

où  $\epsilon$  et  $\beta$  sont constantes scalaires positives, est continu et il est montré que l'erreur  $e = (x - \hat{x})$  est globalement exponentiellement stable. Cependant, ces deux approches

doivent respecter la condition très restrictive (2.3.70), connue comme "matching condition" [39] qui n'est pas souvent satisfaite dans beaucoup d'applications.

Une approche relative est l'observateur par modes glissants proposés par [124; 125; 126]. Dans [126], une structure similaire à l'observateur de Luenberger a été utilisée en ajoutant une fonction de changement avec de gains constants comme partie des corrections en boucle fermée.

Plus spécifiquement, un observateur par modes glissants a été proposé pour la classe de systèmes en forme compagnon

$$\begin{aligned} \dot{x}_1^{(n)} &= f(x, t) \\ y &= x_1 \end{aligned} \quad (2.3.75)$$

L'observateur glissant pour le système (2.3.75) est

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 + k_1(x_1 - \hat{x}_1) + \bar{k}_1 \operatorname{sgn}(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 &= \hat{x}_3 + k_2(x_1 - \hat{x}_1) + \bar{k}_2 \operatorname{sgn}(x_1 - \hat{x}_1) \\ &\vdots \\ \dot{\hat{x}}_n &= f(\hat{x}, t) + k_n(x_1 - \hat{x}_1) + \bar{k}_n \operatorname{sgn}(x_1 - \hat{x}_1) \end{aligned} \quad (2.3.76)$$

Si l'erreur est définie par  $e_i = x_i - \hat{x}_i$ ,  $i = 1, 2, \dots, n$ , alors l'équation dynamique de l'erreur est

$$\begin{aligned} \dot{e}_1 &= e_2 - k_1 e_1 - \bar{k}_1 \operatorname{sgn}(e_1) \\ \dot{e}_2 &= e_3 - k_2 e_1 - \bar{k}_2 \operatorname{sgn}(e_1) \\ &\vdots \\ \dot{e}_n &= \Delta f - k_n(e_1) - \bar{k}_n \operatorname{sgn}(e_1) \end{aligned} \quad (2.3.77)$$

où  $\Delta f = f(x, t) - f(\hat{x}, t)$ , les valeurs constantes  $k_i$  sont choisies comme dans un observateur Luenberger (qui correspond à  $\bar{k}_i = 0$ ,  $i = 1, \dots, n$ ) afin de placer les pôles aux endroits désirés et la fonction *signe* est définie par

$$\operatorname{sgn}(t) = \begin{cases} 1 & \text{si } t > 0 \\ -1 & \text{si } t < 0 \end{cases}$$

**Remarque 2.3.1.** La valeur de  $\Delta f$  dépend de l'effort de modélisation et de la complexité informatique permise dans l'observateur lui-même.  $\square$

La fonction de glissement peut être définie en termes de  $e_1$  par  $s = e_1$  et une condition de glissement par

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s|$$

ou par

$$\frac{d}{dt} s = -\eta \operatorname{sgn}(s)$$

où  $\eta$  est une constante positive.

Pour conduire  $e_1$  à sa surface de glissement,  $\dot{e}_1$  doit avoir le signe contraire à celui-ci de  $e_1$ . Ce qui est fait, si la valeur de  $\bar{k}_1$  est donnée par

$$\bar{k}_1 \geq \eta + F \quad (2.3.78)$$

où  $F$  est une constante positive obtenue de l'hypothèse que la dynamique  $\Delta f$  est explicitement bornée, c'est à dire  $|\Delta f| \leq F$ . Donc, l'inégalité (2.3.78) peut faire que les trajectoires variantes pointent en direction de la surface  $s(t) = 0$  où l'erreur d'estimation  $e_1$  est zéro.

En appliquant le concept de dynamique équivalente [127], la combinaison convexe (une moyenne) de la dynamique sur les deux côtes de la surface  $s(t)$  mène à

$$\begin{aligned} \dot{e}_1 &= \gamma(e_2 + k_1 e_1 + \bar{k}_1) + (1 - \gamma)(e_2 + k_1 e_1 - \bar{k}_1) \\ \dot{e}_2 &= \gamma(k_2 e_1 + \bar{k}_2) + (1 - \gamma)(k_2 e_1 - \bar{k}_2) \\ &\vdots \\ \dot{e}_n &= \gamma(k_n e_n + \bar{k}_n) + (1 - \gamma)(k_n e_n - \bar{k}_n) \end{aligned} \quad (2.3.79)$$

Dès conditions de glissement et de l'équation (2.3.78), la dynamique de la surface de glissement est nulle ( $\dot{s} = \dot{e}_1 = 0$ ), ce qui donne

$$\gamma(e_2 + k_1 e_1 + \bar{k}_1) + (1 - \gamma)(e_2 + k_1 e_1 - \bar{k}_1) = 0$$

d'où la valeur de  $\gamma$  (calculée avec  $s = e_1 = 0$ ),

$$\gamma = \frac{\bar{k}_1 - e_2}{2\bar{k}_1} \quad (2.3.80)$$

En utilisant l'équation (2.3.80) dans l'équation (2.3.79), l'observateur glissant d'ordre réduit est obtenu sous la forme suivante

$$\begin{aligned} \dot{e}_2 &= -(\bar{k}_2/\bar{k}_1)e_2 \\ \dot{e}_3 &= -(\bar{k}_3/\bar{k}_1)e_2 \\ &\vdots \\ \dot{e}_n &= -(\bar{k}_n/\bar{k}_1)e_n \end{aligned} \quad (2.3.81)$$

Les  $(n - 1)$  pôles associés avec la dynamique implicite de glissement sur la région de glissement sont donnés par

$$\det(\lambda I_{n-1} - \begin{bmatrix} -\bar{k}_2/\bar{k}_1 & 1 & 0 & \cdots & 0 \\ -\bar{k}_3/\bar{k}_1 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ \vdots & & & \cdots & 1 \\ -\bar{k}_n/\bar{k}_1 & 0 & 0 & \cdots & 0 \end{bmatrix}) = 0 \quad (2.3.82)$$

où  $I_{n-1}$  est la matrice d'identité de ordre  $(n-1)$ . Ainsi, les pôles sur la région de glissement peuvent être placés arbitrairement par le choix approprié des rapports  $(\bar{k}_i/\bar{k}_1)$ ,  $i = 2, \dots, n$ . Comme  $\bar{k}_1$  est calculé à partir de l'équation (2.3.78), les pôles restants  $k_i$ ,  $i = 2, \dots, n$  sont définis de telle sorte que la dynamique implicite associée à la région de glissement soit d'une façon critique amortie, c'est à dire, qui possède tous les pôles réels et égaux à une constante positive  $\zeta$ .

Pour le cas de systèmes multi-sortie et qui ne soient pas dans la forme compagnon, c'est à dire, systèmes de la forme

$$\begin{aligned} \dot{x} &= f(x, t) \\ y &= Cx \end{aligned} \quad (2.3.83)$$

l'observateur par modes glissants correspondant est donné par

$$\dot{\hat{x}} = f(\hat{x}, t) + K(y - C\hat{x}) + \bar{K}v_s \quad (2.3.84)$$

où  $K$  et  $\bar{K}$  sont matrices de gain de dimension  $(n \times p)$  à être spécifiées et  $v_s$  est le vecteur de dimension  $(p \times 1)$  défini par

$$v_s = [\text{sgn}(y_1 - \hat{x}_1) \quad \text{sgn}(y_2 - \hat{x}_2) \quad \cdots \quad \text{sgn}(y_p - \hat{x}_p)]$$

Afin d'obtenir les matrices de gain  $K$  et  $\bar{K}$  deux procédures différentes ont été proposées dans [96]. La première donne des éléments afin que la matrice de transfert de la linéarisation du système (2.3.83) puisse être rendue strictement positive réelle. L'autre utilise l'extension du critère de cercle pour le cas multivariable et le critère de cercle pour le cas d'une seule mesure, en remplaçant dans le vecteur  $v_s$  la fonction *signe* par la fonction *saturation* qui est définie par

$$\text{sat}((y_i - \hat{x}_i)/\epsilon_i) = \begin{cases} \frac{(y_i - \hat{x}_i)}{|(y_i - \hat{x}_i)|} & \text{si } |(y_i - \hat{x}_i)| \geq \epsilon_i \\ \frac{(y_i - \hat{x}_i)}{\epsilon_i} & \text{si } |(y_i - \hat{x}_i)| < \epsilon_i \end{cases}$$

Ces procédures nécessitent la satisfaction de certaines conditions. Ainsi, les procédures développées pour déterminer les gains de commutation sont compliquées et la question de la performance des observateurs n'est pas analysée.

Un développement supplémentaire est fait dans [143] où un observateur par modes glissants d'ordre réduit est proposé avec un gain en boucle fermée temps-variant pour faire ensuite que les valeurs propres de l'erreur linéarisée du système soient constantes.

Pour les systèmes multi-sorties, la synthèse de l'observateur peut être obtenue directement à partir du système mono-sortie. Cependant, la méthode pour trouver la matrice de gains de changement  $K$  nécessite des manipulations mathématiques plus sophistiquées.

Dans [120] une combinaison contrôleur-observateur a été conçu pour les systèmes affines en la commande. En particulier, une convergence de l'erreur d'observation qui peut être rendue aussi petite que désirée a été prouvée conformément à quelques suppositions sur le système et sur le contrôleur par retour d'état.

Dans [3], le travail de [120] est étendu à une classe de systèmes non linéaires incertains triangulaires en combinant un observateur glissant avec un mode de glissement robuste par l'intermédiaire d'une procédure de back-stepping. La détection des défaillances a été étudiée dans [46], [144].

### Commentaires sur l'approche

#### *Avantages*

- La forme de l'observateur est simple et le résultat est robuste.

#### *Inconvénients*

- Satisfaire toutes les hypothèses restrictives nécessaires pour pouvoir réaliser la synthèse de l'observateur.

### 2.3.8 Les observateurs basés sur l'optimisation

Pour surmonter toutes les difficultés rencontrées pour la synthèse d'observateurs dues aux transformations de coordonnées, aux études techniques sur les systèmes et surtout aux structures particulières, les observateurs basés sur l'optimisation ont été proposés pour la classe de systèmes généraux

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= h(x) \end{aligned} \tag{2.3.85}$$

Ceux-ci sont des observateurs dont la dynamique est définie pour diminuer dynamiquement la norme  $L_2$  de l'erreur de prédiction de sortie. En fait, ceci permet d'éviter l'étude de l'erreur d'estimation qui est généralement réalisée afin de prouver la convergence de l'observateur. Plus spécifiquement, dans les méthodes par optimisation, une fonction de coût est définie comme la norme  $L_2$  de la différence entre la sortie mesurée et la sortie estimée basée sur l'estimation courante de l'état, à savoir

$$J(z(t), x(t-T)) = \left\| y(\cdot, 0, w(t), u) - y(\cdot, 0, x(t-T), u) \right\|_{L_2}^{[0, T]} \tag{2.3.86}$$

où  $w(t)$  est l'estimation courante de  $x(t-T)$  et  $T > 0$  est la longueur de l'horizon d'observation. En effet, cette quantité représente l'erreur de prédiction de sortie associée à l'estimation  $z(t)$  de  $x(t-T)$ . L'estimation de l'état est alors basée sur l'approximation en ligne de la minimisation de (2.3.86).

Dans [151], des méthodes de gradient sont appliquées pour traiter le problème de minimisation de (2.3.86), cependant employer des méthodes de Newton résulte en une grande consommation de temps de calcul. Dans [152], la matrice Hessienne de (2.3.86) doit être calculée explicitement pour exécuter la méthode d'approximation de Newton,



cela peut être difficile à faire dans un contexte de temps réel ou impossible si les fonctions fondamentales ne sont pas deux fois différentiables. En outre, il est parfois difficile de l'appliquer à un problème réel.

Dans [95], le problème d'optimisation non linéaire doit être résolu à chaque instant d'échantillonnage, ceci peut présenter quelques inconvénients sérieux dans le sens que le temps nécessaire pour réaliser un calcul pas-à-pas avec une exactitude indiquée à l'avance peut ne pas être contrôlable. De plus, il est nécessaire que le modèle du système soit parfaitement connu.

Dans les travaux cités plus haut, le problème d'optimisation doit être entièrement résolu à chaque période d'échantillonnage. Ceci est très délicat dans un contexte de temps réel. Dans [4], Une version dynamique qui se présente sous forme d'un observateur dynamique avec copie du système et terme d'innovation a été proposé. De plus, des solutions techniques ont été proposées pour partiellement surmonter les inconvénients du temps de calcul à l'aide de la technique de post-stabilisation. De cette façon, le rapport (précision/coût de calcul) peut être sensiblement amélioré. L'utilisation d'un schéma de régularisation qui évite les singularités au prix d'un observateur qui ne converge que vers un voisinage de l'état a été aussi proposé. Cependant, une hypothèse de régularité uniforme globale a été employée afin d'obtenir les résultats de convergence.

Pour perfectionner les résultats présentés ci-avant, dans le Chapitre 4, le cas où de telles hypothèses de régularité uniforme globale ne sont pas supposées sont étudiées. Ce sera aussi l'occasion de s'attarder sur les équations de cet observateur.

### Commentaires sur l'approche

#### *Avantages*

- La synthèse de l'observateur est indépendante de toute structure particulière du système.

#### *Inconvénients*

- La grande quantité de calculs nécessaires pour résoudre le problème de minimisation.

## 2.4 Conclusion

L'étude des différentes techniques de synthèse d'observateurs a permis d'établir les avantages et les inconvénients de chacune des méthodologies, en résumé

- Certaines méthodes apparaissent très liées à une structure particulière du système.
  - ★ Observateurs basées sur les fonctions de Lyapunov.
  - ★ Observateurs à grand gain.

- Certaines ne prennent pas en compte les erreurs de modèle et les perturbations.
  - ★ Filtre de Kalman étendu
  - ★ Observateurs sous une forme canonique.
  - ★ Observateurs à grand gain.
  - ★ Observateurs par linéarisation étendue.
- Certaines nécessitent des calculs importants.
  - ★ Filtre de Kalman étendu
  - ★ Observateurs adaptatifs.
  - ★ Observateurs par modes glissants.
  - ★ Observateurs par optimisation.

Pour ces raisons, le point de départ de ce travail est de proposer un schéma de synthèse qui tente de surmonter les inconvénients décrits ci-dessus. Plus spécifiquement, l'objectif est d'aller vers l'obtention d'une nouvelle méthodologie d'optimisation qui essaie de prendre en compte les points suivants :

- *Indépendance vis à vis des conditions structurelles du système de manière à obtenir une approche plus systématique et générique.*

La synthèse de l'observateur proposé n'a pas besoin d'une structure particulière du système à observer.

- *Prise en compte explicite des incertitudes et/ou des perturbations.*

La prise en compte explicite des incertitudes présente l'avantage de tenir compte de la façon précise dont ces incertitudes apparaissent dans la dynamique du système.

- *Temps de calcul réduit.*

En comparaison aux schémas d'observation du type d'horizon glissant qui cherchent à chaque période d'échantillonnage le minimum global d'un problème d'optimisation de dimension  $n$ .

Disons tout de suite que l'objectif du temps de calcul réduit n'est pas complètement résolu à l'heure actuelle même si des résultats encourageants ont été obtenus et des pistes d'accélération des algorithmes mis en œuvre ont pu être dégagées. Le chapitre suivant est consacré à la présentation d'une nouvelle méthodologie de synthèse d'observateurs qui n'est toujours qu'à ses débuts.



Vers la synthèse d'observateurs  
optimaux robustes



## Chapitre 3

# Vers la Synthèse d'Observateurs Optimaux Robustes

L'étude bibliographique réalisée dans le chapitre précédent ainsi que les études comparatives des différentes techniques de synthèse d'observateurs trouvées dans [24; 96; 133; 140; 143] montrent que la synthèse d'observateurs pour les systèmes non linéaires reste une tâche très difficile et aucune méthodologie ne peut aujourd'hui prétendre aboutir systématiquement à un observateur asymptotiquement et globalement convergent pour tout système même uniformément observable.

L'étude a montré aussi que ces observateurs souffrent au moins d'un des inconvénients suivants :

- Certaines méthodes apparaissent très liées à une structure particulière du système.
- Certaines méthodes ne prennent pas explicitement en compte les erreurs de modèle et les perturbations.
- Certaines méthodes nécessitent des calculs lourds.

Compte tenu de ce qui précède, l'objectif initial de ce travail était d'aller vers l'obtention d'une nouvelle méthodologie pour construire des observateurs optimaux robustes qui tentent de respecter les objectifs suivants :

- *Indépendance vis à vis des conditions structurelles du système de manière à obtenir une approche plus systématique et générique.*

Ceci ne veut pas dire que la réussite de l'observateur ne nécessite pas d'hypothèses restrictives. Cependant, la nature de ces hypothèses doit être non constructive. Plus précisément, la construction, elle-même de l'algorithmique de l'observateur ne doit pas dépendre à priori d'une structure particulière du système à observer.

- *Prise en compte explicite des incertitudes et/ou des perturbations dans l'algorithmique de l'observateur.*

Ceci ne nie pas la robustesse que peut avoir tel ou tel observateur de part sa définition même si celle-ci n'en tient pas compte explicitement. Cependant, la prise en compte explicite des incertitudes présente génériquement l'avantage de tenir compte de la façon précise dont ces incertitudes apparaissent dans la dynamique du système. Elle devrait aussi les faire intervenir dans la limite de leurs amplitudes qui, elles aussi doivent être quantitativement considérées.

- *Temps de calcul réduit.*

Cet objectif devrait exclure des schémas d'observation du type de ceux adoptant le principe de l'horizon glissant qui cherchent, à chaque période d'échantillonnage, le minimum global d'un problème d'optimisation de dimension  $n$  (la dimension de l'état) dont la solution est l'état à estimer. En effet, bien que théoriquement imparable, cette démarche sur-estime les capacités actuelles et futures des procédures d'optimisation globale multi-variables.

La méthode proposée dans ce travail consiste d'abord à diviser le problème d'optimisation principal en  $n$  problèmes d'optimisation scalaires, où chaque problème scalaire est appelé "**Problème Élémentaire**" (PE). Puis, à résoudre chaque problème élémentaire scalaire en utilisant un solveur numérique de l'équation de Hamilton-Jacobi scalaire (en la variable de l'espace).

Les solutions des différents problèmes scalaires sont alors "analysées" pour définir un nouvel ensemble de problèmes scalaires pour l'itération suivante. Les itérations continuent jusqu'à ce que toutes les solutions soient mutuellement compatibles dans un sens qui sera clarifié dans la suite.

Il va de soi que le problème de l'estimation robuste peut être décrit par une seule équation aux dérivées partielles du type Hamilton-Jacobi-Bellman multi-variable de dimension  $n$ . L'intérêt de la décomposition en  $n$  problèmes élémentaires vient du fait que la résolution des équation de Hamilton-Jacobi dans le cas multi-variable est une tâche très difficile à réaliser de manière générale [5; 12; 21; 56; 86; 119]. Cependant le cas scalaire peut bénéficier actuellement de solveurs numériques très puissants. Un simple coup d'œil sur les catalogues des bibliothèques numériques permet de s'en convaincre.

Le principe de résolution ressemble beaucoup à la commande décentralisée [121; 122; 135; 148]. Cependant les différences suivantes peuvent être énoncées :

- Dans la commande décentralisée
  1. Le système original est transformé en sous-systèmes qui parfois doivent avoir une structure particulière.
  2. L'état initial de chaque sous-système est connu.
  3. Des études de convergence existent sous certaines hypothèses.
- Dans la méthode proposée
  1. Le problème original est divisé en systèmes scalaires. Le système ne nécessite pas d'avoir une structure particulière.
  2. L'état initial de chaque système scalaire est inconnu.
  3. Il n'existe pas pour le moment de preuves de convergence de notre approche. Tout résultat de convergence ne peut s'obtenir qu'au prix des hypothèses restrictives. Il nous paraît donc déraisonnable de tenter d'obtenir un résultat de convergence avec le niveau de généralité que nous avons considérée.

Comme il peut être noté des études plus approfondies doivent être effectuées pour déterminer les avantages d'une méthode sur l'une autre, cependant par rapport aux objectifs énoncés en début du présent chapitre, et en anticipant sur la présentation qui suit, il nous semble que l'algorithme de résolution proposé est un premier pas vers l'obtention d'un schéma d'observation non linéaire qui est indépendant des conditions particulières de structure du système, avec la capacité de traiter explicitement les perturbations et/ou les incertitudes de modèle.

Le temps de calcul reste à notre sens un point sur lequel les efforts futurs devraient se concentrer même s'il s'agit là d'une question relative au temps caractéristique de chaque problème.

Il est clair qu'on ne propose pas d'utiliser systématiquement l'approche proposée, celle-ci doit s'inscrire dans la logique suivante :

- Tout d'abord le système est analysé. Si le système possède une structure particulière, alors une méthode classique de synthèse des observateurs doit être appliquée pour résoudre le problème d'observation.
- Si le système ne possède pas de structure particulière ou une méthode classique ne peut pas être appliquée, alors le problème d'observation doit être résolu par optimisation.
- Dans ce cas, si le système possède une dimension faible, alors les méthodes classiques d'optimisation peuvent être appliquées.
- Mais si le système est de grande dimension, alors l'algorithme d'optimisation proposé pourrait être envisagé.



En d'autres termes, on va utiliser cette nouvelle méthode quand pour un système, toutes les autres techniques ne peuvent pas être appliquées.

En tout état de cause, le travail développé dans ce chapitre est plutôt exploratoire et à ce titre, il devrait être vu et jugé comme un premier pas vers la construction d'un observateur à horizon glissant répondant aux exigences énoncées plus haut.

Dans la suite de ce chapitre, le problème d'observation qui nous intéresse est d'abord clairement posé. Ensuite, le principe de l'algorithme fondé sur la décomposition du problème d'observation en problèmes élémentaires est expliqué. Le problème élémentaire est alors clairement défini et la façon dont on procède pour sa résolution est explicitée. Le chapitre se termine par un exemple d'illustration permettant d'avoir une première idée des performances de la méthodologie et peut-être aussi parfois ses limites fondamentales ainsi que les points techniques qu'il convient d'approfondir dans un futur travail.

## 3.1 Le problème d'observation

Dans ce paragraphe, le problème d'observation étudié dans ce travail est présenté. Pour ce faire, la classe des systèmes considérés est d'abord précisée, puis, le problème d'observation est clairement décrit.

### 3.1.1 La classe des systèmes considérés

On considère des systèmes non linéaires régis par les équations différentielles suivantes

$$\begin{aligned} \dot{x} &= f(x, u, \delta, t) \\ y &= h(x, \delta, t) \end{aligned} \quad (3.1.1)$$

où  $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^p$ ,  $u \in \mathbb{R}^m$  et  $\delta \in \mathbb{R}^q$  représentent respectivement l'état, la sortie mesurée, l'entrée de commande et le vecteur de perturbations non mesurées ou/et d'incertitudes de modèle.

Les entrées sont supposées appartenir à un ensemble admissible de fonctions bornées. Le vecteur d'incertitudes et de perturbations  $\delta$  est supposé vérifier la contrainte suivante:

$$\forall t \geq 0 \quad ; \quad |\delta_i(t)| \leq \delta_i^{max} \quad ; \quad i = 1, \dots, q \quad (3.1.2)$$

où  $(\delta_i^{max})_{i=1}^q$  est un vecteur de bornes supérieures connues à-priori.

Il est important de noter que le caractère robuste du schéma d'observation proposé dans ce travail est fondé sur la prise en compte explicite de ces bornes sur les amplitudes des incertitudes. Plus précisément, l'algorithme d'estimation simultanée de l'état et des incertitudes aura pour objectif la recherche d'un scénario "*plausible*" portant sur les signaux inconnus et permettant d'expliquer au mieux le comportement passé

de la sortie. Ainsi, les bornes  $\delta_i^{max}$  doivent représenter au mieux les véritables bornes des incertitudes, faute de quoi, dans certains cas, l'algorithme risque de "*pouvoir tout interpréter au moyen des incertitudes*".

Afin de construire le schéma d'observation, les mesures de la sortie et de la commande doivent être acquises et stockées sur une certaine fenêtre d'observation, il est alors nécessaire de définir:

- Un horizon d'observation donné  $T > 0$  et
- Une période d'échantillonnage  $s > 0$  telle que  $(N - 1)s = T$  où  $N \in \mathbb{N}$  est un entier non nul.  $N$  représente alors le nombre d'échantillons de mesures stockées sur la fenêtre d'observation.

### 3.1.2 Formulation du problème

L'objectif de ce travail est de proposer une méthodologie concrète permettant de trouver une bonne solution "*sous-optimale*" au problème d'observation suivant :

*Les mesures de  $y$  et de  $u$  étant acquises avec une période d'échantillonnage  $s$ , donner un algorithme orienté vers la résolution du problème d'optimisation défini par le critère suivant :*

$$\begin{aligned} \hat{x}(\cdot), \hat{\delta}(\cdot) &\leftarrow \min_{\tilde{x}(\cdot), \tilde{\delta}(\cdot)} \sum_{i=1}^N \|y(t - is) - \tilde{y}(t - is)\|^2 \\ \tilde{y}(\tau) &:= h(\tilde{x}(\tau), \tilde{\delta}(\tau), \tau) \end{aligned} \quad (3.1.3)$$

*sous les contraintes (3.1.1) et (3.1.2).*

En utilisant la notation

$$\bar{Y}(t) := (y(t - Ns), \dots, y(t - s))^T$$

la dynamique de l'observateur est donnée par

$$\bar{Y}(t) = A\bar{Y}(t - s) + By(t - s) \quad (3.1.4)$$

$$\begin{pmatrix} \hat{x}(t) \\ \hat{\delta}(t) \end{pmatrix} = \mathcal{F}(\bar{Y}(t), t) \quad (3.1.5)$$

où  $\mathcal{F}(\bar{Y}(t), t)$  est la solution optimale de (3.1.3), alors que  $A$  et  $B$  sont matrices classiques de stockage fini, plus précisément

$$A := \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \cdots & & \ddots & \vdots \\ \vdots & \cdots & & & 0 & 1 \\ 0 & & \cdots & & & 0 \end{pmatrix} \quad B := \begin{pmatrix} 0 \\ \vdots \\ 1 \end{pmatrix} \quad (3.1.6)$$

### Remarque 3.1.1.

- Noter que les équations (3.1.4)-(3.1.5) sont valables pour un système mono-sortie autonome. Dans le cas multi-sorties et/ou avec commande, des notations techniques sont à définir d'une façon analogue afin de représenter le processus de stockage de l'information mesurée.
- Les équations (3.1.4)-(3.1.5) représentent un système dynamique dont l'état interne n'est autre que le vecteur de mesures tandis que la sortie représentant les estimations de l'état et du vecteur des incertitudes sont calculées à travers un processus d'optimisation.  $\square$

Dans la suite de ce travail, nous nous concentrons essentiellement sur la proposition d'un algorithme permettant la résolution du problème d'optimisation (3.1.5). Nous ne revenons plus sur l'insertion de cette résolution dans la définition d'un observateur dynamique. Il ne faut cependant pas perdre de vue que le travail ainsi fait servira dans la construction de l'observateur à horizon glissant (3.1.4)-(3.1.5).

## 3.2 Le principe de l'algorithme

Dans cette section, le principe de l'algorithme de résolution est présenté et un exemple simple est donné pour clarifier la méthodologie exposée.

La conception de l'algorithme de résolution proposé passe par la définition d'un "**Problème Élémentaire**" (PE) dont la résolution est le rôle d'un solveur. Il se trouve autant de problèmes élémentaires que de lignes dans le système dynamique (donc, autant de problèmes élémentaires que d'états).

À la fin de cette phase de résolution des problèmes élémentaires, chaque solveur dispose d'une solution à son problème. Une phase de communication des résultats est alors entamée. Lors de cette phase, l'ensemble des solutions élémentaires est utilisé pour redéfinir les problèmes élémentaires que les solveurs résoudre à l'itération suivante.

C'est cette phase de traitement qui traduit le fait que ces problèmes élémentaires ne sont en fait pas indépendants. C'est cette phase qui oriente l'algorithme vers la définition de problèmes élémentaires dont la solution est compatible avec le système dynamique dans sa totalité. Ainsi, le rôle du bloc de traitement est de produire de nouvelles consignes entrant dans la définition des problèmes élémentaires à partir des résultats de l'ensemble des solveurs. Une phase de calcul parallèle peut ainsi recommencer dans laquelle chaque solveur résout son propre problème.

Le schéma de la Figure 3.1 résume la présentation ci-dessus. Ce schéma utilise les concepts de places et des transitions propres à la représentation des systèmes à événements discrets par les réseaux de Petri.

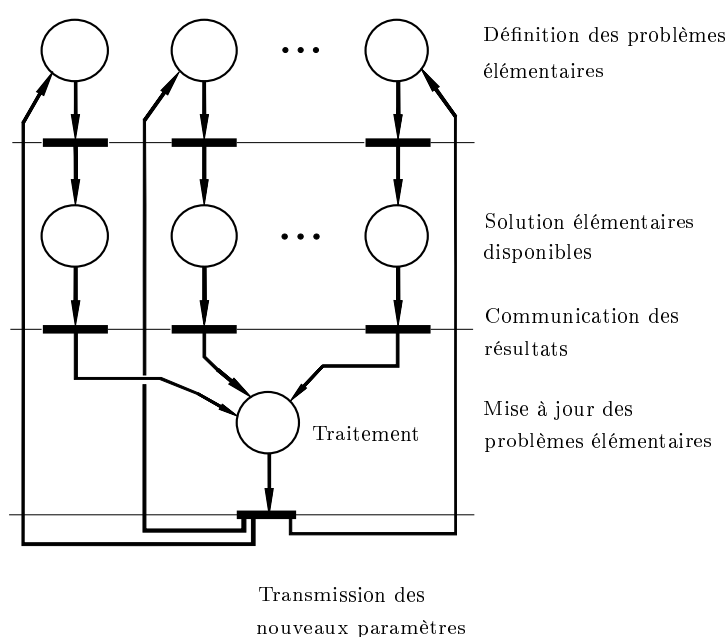


FIG. 3.1: La structure de l'algorithme

Pour clarifier le principe de l'algorithme de résolution, on l'applique au système de Van der Pol modifié suivant :

$$\dot{x}_1 = x_2 + u + v(t) \quad (3.2.1)$$

$$\dot{x}_2 = -9x_1 + \mu(t)(1 - x_1^2)x_2 \quad (3.2.2)$$

$$y = x_1 \quad (3.2.3)$$

où  $v$  est un signal externe non mesuré.

Les mesures de la sortie  $y = x_1$  sont acquises sur un intervalle donné, le premier pas c'est de définir un problème de commande optimale dans lequel (3.2.1) représente le système dynamique,  $x_1$  est l'état,  $(x_2, v)$  est le vecteur de commande,  $u$  est un signal

connu, les signaux  $x_1^{(k-1)}$ ,  $x_2^{(k-1)}$  et  $v^{(k-1)}$  représentent les références à suivre et

$$J = \int_{t-T}^t \left\{ q(x_1(\tau) - x_1^{ref}(\tau))^2 + \left\| \begin{pmatrix} x_2(\tau) \\ v(\tau) \end{pmatrix} - \begin{pmatrix} x_2^{ref}(\tau) \\ v^{ref}(\tau) \end{pmatrix} \right\|_R^2 \right\} d\tau$$

est le critère à minimiser. Les trajectoires optimales résultantes pour l'état  $x_1$  et la commande  $(x_2, v)$ , dénotés par  $x_1^{(k,1)}$  et  $(x_2^{(k,1)}, v^{(k,1)})$  sont obtenues à l'itération  $k$ .

De la même manière un deuxième problème de commande optimale scalaire peut être défini dans lequel (3.2.2) est le système dynamique,  $x_2$  est l'état,  $x_1$  est la commande,  $\mu$  est un signal connu, les signaux  $x_1^{(k-1)}$  et  $x_2^{(k-1)}$  représentent les références à suivre et

$$J = \int_{t-T}^t \left\{ q(x_2(\tau) - x_2^{ref}(\tau))^2 + \|x_1(\tau) - x_1^{ref}(\tau)\|_R^2 \right\} d\tau$$

est le critère à minimiser pour ce nouveau problème optimal. Les solutions optimales résultantes à l'itération  $k$  pour ce deuxième problème sont dénotés par  $x_1^{(k,2)}$  et  $x_2^{(k,2)}$ .

Ainsi, d'une manière générale,

$x_j^{(k,i)}$  : La solution du **PE**  $i$  à l'itération  $k$  concernant la variable  $x_j$

Comme on peut le noter, deux trajectoires pour  $x_1$  ( $x_1^{(k,i)}$  pour  $i \in \{1, 2\}$ ) à l'itération  $k$  seront ainsi obtenues et deux pour  $x_2$  ( $x_2^{(k,i)}$  ( $i \in \{1, 2\}$ )). Chacune de ces trajectoires correspond à la solution retrouvée par les solveurs de chacun des problèmes élémentaires, un traitement des résultats sera alors réalisé de façon à obtenir une seule estimation (à l'itération  $k$ ) pour  $x_1$  et  $x_2$ . Cette estimation sera notée  $x_j^k$ , plus précisément

$x_j^k$  : L'estimation adoptée à l'itération  $k$  après traitement

Ayant en main les résultats du traitement, à savoir

$$x_1^k ; x_2^k ; v^k$$

Les solveurs recommencent la résolution des nouveaux problèmes élémentaires à l'itération  $(k + 1)$ .

Il devient alors clair que le problème élémentaire qui est au cœur de l'algorithme proposé est un problème de commande optimale scalaire temps-variant qui a la particularité d'avoir un état initial inconnu. En effet, dans le problème élémentaire 2 ayant (3.2.2) comme équation dynamique,  $x_2$  comme état et  $x_1$  comme commande, la valeur initial de  $x_2$  ne peut pas être considérée comme connue dans la mesure où tout consiste à estimer l'état non mesuré  $x_2$ .

Dans la section suivante, le problème de commande optimale élémentaire utilisé jusqu'ici est étudié ainsi que la méthode proposée pour sa résolution.

### 3.3 Le problème élémentaire

Dans cette section, les caractéristiques du **PE** sont d'abord présentées. Ensuite, la méthodologie générale de résolution adoptée dans ce travail est expliquée. Pour cela, le formalisme conduisant aux équations de Hamilton-Jacobi est d'abord rappelé. Ensuite, l'application de ce formalisme à la résolution du problème élémentaire est montrée.

#### 3.3.1 Définition du problème élémentaire

Le problème élémentaire utilisé dans l'algorithme de résolution du problème (3.1.5) est un problème de commande optimale d'un système dynamique scalaire donné par

$$\dot{\xi}(\tau) = F(\xi(\tau), x^c(\tau), \nu(\tau), \tau) \quad ; \quad \tau \in [t - T, t] \quad (3.3.1)$$

où

- $\xi \in \mathbb{R}$  est l'état.
- $x^c(\cdot)$  est un signal dont l'évolution est connue sur  $[t - T, t]$ .
- $\nu$  est un vecteur de commande satisfaisant une contrainte du type

$$\nu(\tau) \in \bar{V} \subset \mathbb{R}^{n_\nu}$$

où  $n_\nu \in \mathbb{N}$  est le nombre d'entrées "de commande" impliquées dans le problème élémentaire considéré.

Le problème consiste alors à résoudre le problème de commande optimale à ÉTAT INITIAL INCONNU suivant :

$$\min_{\xi(\cdot); \nu(\cdot) \in \bar{V}^{[t-T, t]}} \int_{t-T}^t \left( q(\xi(\tau) - \xi^{ref}(\tau))^2 + \|\nu(\tau) - \nu^{ref}(\tau)\|_R^2 \right) d\tau \quad (3.3.2)$$

dans lequel  $\xi^{ref}(\cdot)$  et  $\nu^{ref}(\cdot)$  sont des trajectoires de référence données à-priori.  $q \in \mathbb{R}_+$  et  $R \in \mathbb{R}^{n_\nu}$  sont des pondérations. Lors des résolutions successives des problèmes élémentaires dans le cadre de l'algorithme dont il a été question plus haut, ces pondérations sont variées dynamiquement selon des principes qui seront explicités dans la suite.

**Remarque 3.3.1.** Il est important de souligner que le problème de commande optimale en question est à état initial inconnu. Ceci provient directement du fait que l'on se place dans le cadre de l'estimation d'état. Ceci rend délicat l'usage du principe de maximum de Pontryagin sous sa forme conduisant au problème aux deux bouts par des méthodes de tirs multiples ou de gradient.  $\square$

Dans la section suivante, la façon dont on procède pour la résolution d'un problème élémentaire défini plus haut est expliquée.

### 3.3.2 Résolution générale du problème élémentaire

Comme il a été mentionné dans la partie précédente, un problème élémentaire est un problème de commande optimale ayant les caractéristiques suivantes :

- Le système est temps variant même si la dynamique du système observé est temps invariant. Ceci provient des signaux de références et des signaux connus qui sont par nature temps variant.
- L'état initial est inconnu.

Afin de résoudre ce problème, on procède en deux étapes :

1. Le problème élémentaire représenté par les équations (3.3.1)-(3.3.2) est d'abord transformé en une équation de Hamilton-Jacobi **scalaire** (voir section 3.3.3 ci-après). C'est cette dernière propriété (scalaire) qui fait qu'il est possible de résoudre cette équation en utilisant des codes de calcul existants et efficaces. Cette résolution présente l'avantage de donner les valeurs des solutions optimales correspondantes à tous les états initiaux possibles. En effet, la solution de ces équations aboutit à une fonction  $V(0, \xi)$  représentant la valeur optimale du critère en fonction de l'état initial.
2. L'état initial est alors choisi pour être celui qui minimise  $V(0, \xi)$ , c'est à dire

$$\hat{\xi}(t - T) = \arg\left\{\min_{\xi} V(0, \xi)\right\} \quad (3.3.3)$$

Il est clair que la trajectoire de l'état  $\hat{\xi}(\cdot)$  et le profil de la commande  $\hat{v}(\cdot)$  s'obtiennent par intégration de (3.3.1) sur  $[t - T, t]$  à partir de l'état initial obtenu grâce à la résolution de (3.3.3) qui est, rappelons-le, un problème d'optimisation scalaire.

En résumé, la procédure consiste à scinder les inconnues en deux groupes

L'état initial d'une part.

La commande d'autre part.

La commande optimale est d'abord trouvée en tant qu'une fonction implicite de l'état initial (à travers l'intégration du système différentiel (3.3.1)), celui-ci est ensuite choisi pour minimiser le critère global initial (3.3.2).

Dans la section suivante on rappelle le formalisme de Hamilton-Jacobi utilisé pour la recherche de lois de commande par retour d'état minimisant un critère donné.

### 3.3.3 Les équations de Hamilton-Jacobi

Le formalisme de Hamilton-Jacobi permet de caractériser la solution d'un problème de commande optimale à travers des équations aux dérivées partielles avec conditions aux limites. Lorsqu'elle est possible, la résolution de ces équations permet de déterminer la commande optimale sous forme d'un retour d'état temps-variant et de bénéficier ainsi des avantages connus des systèmes bouclés. Dans cette section, ce formalisme est brièvement rappelé.

On considère un système non linéaire donné par

$$\dot{x} = f(x, u, t) \quad ; \quad x \in \mathbb{R}^n \quad ; \quad u \in \mathbb{R}^m \quad (3.3.4)$$

où  $x$  est l'état et  $u$  est la commande. La commande  $u$  est soumise pour tout  $t \geq 0$  aux contraintes suivantes :

$$u(t) \in \mathbb{U}(x(t)) \subseteq \mathbb{R}^m \quad (3.3.5)$$

où  $\mathbb{U}(x(t))$  est un sous-ensemble (généralement compact) de l'espace de commande. Soit le problème de commande optimale suivant:

Trouver une stratégie de commande en boucle fermée

$$u(x(\cdot), \cdot) \in \mathbb{U}(x(\cdot))$$

telle que

$$u(x(\cdot), \cdot) = \arg\left\{\min_{u(\cdot)} J_T(x(t), u(\cdot), t)\right\} \quad (3.3.6)$$

pour tout  $t \geq 0$  sous (3.3.4)

où

$$J_T(x(t), u(\cdot), t) = \Psi(x(T)) + \int_t^T L(x(\tau), u(\tau), \tau) d\tau \quad (3.3.7)$$

où  $L(\cdot)$  est une fonction de pénalité qui exprime les objectifs de commande. En connexion avec la formulation précédente, le Hamiltonien  $H$  est défini pour tout  $\lambda \in \mathbb{R}^n$  par

$$H(x, u, t, \lambda) = \lambda^T f(x, u, t) + L(x, u, t) \quad (3.3.8)$$

Pour la suite de la présentation, l'hypothèse suivante est considérée

**Hypothèse 3.3.1.** Pour tout triplet  $(x, \lambda, t) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_+$  il existe  $\hat{u}(x, \lambda, t)$  solution de

$$\hat{H}(x, \lambda, t) = \min_{u \in \mathbb{U}(x)} H(x, u, t, \lambda) \quad (3.3.9)$$



En général, des arguments basés sur la continuité des fonctions mises en jeu et le fait que le critère  $L(\cdot)$  présente une pondération sur la commande font que l'Hypothèse 3.3.1 est satisfaite pour une large classe de problèmes.

Sous l'Hypothèse 3.3.1, "le théorème de vérification" d'Isaacs rappelé ci-dessous peut être appliqué [13], [65].

**Théorème 3.3.2.** (*Théorème de vérification d'Isaacs*).

**Si**  $V(x, t)$  est une fonction de classe  $C^1$  en  $x$  et  $t$  qui satisfait l'équation suivante (dite de Hamilton-Jacobi) avec condition terminale:

$$V_t + \hat{H}(x, V_x, t) = 0 \quad ; \quad V(x, T) = \Psi(x) \quad (3.3.10)$$

où

$$V_x(x, t) = \frac{\partial V}{\partial x}(x, t) \quad ; \quad V_t(x, t) = \frac{\partial V}{\partial t}(x, t)$$

et  $\hat{H}$  est donnée par (3.3.9),

**alors** la stratégie de commande

$$u(x(\tau), \tau) = \hat{u}(x(\tau), V_x(x(\tau), \tau), \tau) \quad ; \quad \tau \in [t, T] \quad (3.3.11)$$

est une solution optimale pour le problème (3.3.6). En outre, la valeur optimale correspondante est  $V(x(t), t)$ . ♠

Pour une idée de la preuve du Théorème de vérification 3.3.2 voir l'annexe C.

En résumé, l'application de ce qui précède au problème de commande optimale scalaire associé au système

$$\dot{\xi} = F(\xi, x^c, \nu, \tau)$$

et au critère

$$J = \int_{t-T}^t L(\xi, x^c, \nu, \tau) d\tau = \int_{t-T}^t \left( q(\xi(\tau) - \xi^{ref}(\tau))^2 + \|\nu(\tau) - \nu^{ref}(\tau)\|_R^2 \right) d\tau$$

permet d'écrire:

**S'il existe** une fonction  $V(\xi, \tau)$  de classe  $C^1$  en  $\xi$  et  $\tau$  satisfaisant l'équation de Hamilton-Jacobi avec la condition finale  $V(\xi, T) = 0$  suivante :

$$V_\tau(\tau, \xi) + \hat{H}(\tau, \xi, V_\xi(\tau, \xi)) = 0 \quad (3.3.12)$$

dans laquelle

$$\begin{aligned} \hat{H}(\tau, \xi, V_\xi) &= H(\tau, \xi, x^c, \hat{\nu}(\tau, \xi, V_\xi), V_\xi) \\ H(\tau, \xi, x^c, \nu, \lambda) &= L(\tau, \xi, x^c, \nu) + \lambda^T F(\tau, \xi, x^c, \nu) \\ \hat{\nu}(\tau, \xi, V_\xi) &= \arg\{\min_{\nu \in V} H(\tau, \xi, x^c, \nu, V_\xi)\} \\ V_\xi(\xi, \tau) &= \frac{\partial V}{\partial \xi}(\xi, \tau) \quad ; \quad V_\tau(\xi, \tau) = \frac{\partial V}{\partial \tau}(\xi, \tau) \end{aligned}$$

**alors**, résoudre le **PE** défini par les équations (3.3.1)-(3.3.2) revient à intégrer l'équation aux dérivées partielles scalaire (3.3.12) en temps inverse afin d'obtenir la fonction  $V(\xi, 0)$  et la commande associée par intégration.

**Remarque 3.3.2.** La commande  $\nu(\tau, \xi)$  qui minimise le Hamiltonien  $H(\tau, \xi, x^c, \nu, \lambda)$  peut être une perturbation qui est limitée par une borne inférieure et par une borne supérieure  $\delta_{min} \leq \delta \leq \delta_{max}$ . En résolvant l'équation de Hamilton-Jacobi (3.3.12), on obtient la trajectoire optimale de la perturbation  $\hat{\delta}(\tau, \xi)$  qui est une commande qui est utilisée pour calculer l'état initial. C'est à dire, les équations de Hamilton-Jacobi tiennent compte

- De la façon dont la perturbation  $\delta$  agit sur le système.
- De l'amplitude de la perturbation.

où la perturbation  $\delta$  n'a pas de structure temporelle à priori. □

Contrairement aux équations de Hamilton-Jacobi générales dans lesquelles l'état  $x$  appartient à  $\mathbb{R}^n$  qui sont généralement très difficiles à résoudre [5; 12; 21; 56; 86; 119], les équations de Hamilton-Jacobi dont il est question ici sont définies pour un état scalaire  $\xi$ . Ceci rend leur résolution relativement aisée moyennant des codes efficaces existants. Dans le travail présenté dans cette thèse, la résolution de l'équation (3.3.12) est faite par la procédure DMOLCH de la bibliothèque IMSL du compilateur FORTRAN.

Dans la section suivante, une mise en œuvre particulière de l'approche décrite auparavant est présentée. Elle consiste à transformer l'équation non linéaire (3.3.1) en une équation linéaire temps-variant en définissant certains termes de certaines lignes du système différentiel comme des signaux de commande. Ceci n'est rigoureusement pas possible systématiquement, mais le concept de commande virtuelle utilisé dans le cœur même de la définition d'un problème élémentaire rend la classe de système pour lesquels ceci est possible relativement large. Ceci sera plus clair dans la suite de l'exposé.

### 3.3.4 La transformation du problème élémentaire en un problème linéaire quadratique

Dans beaucoup de cas et sous réserve de faire un choix particulier des vecteurs de commande dans les problèmes élémentaires, les équations de Hamilton-Jacobi peuvent prendre la forme particulière des équations de Riccati temps-variant. Ceci permet de trouver la solution optimale de chaque problème élémentaire en résolvant une équation de Riccati temps-variant [11] par les solveurs classiques performants basée sur l'algèbre linéaire.

Pour clarifier la transformation du problème d'optimisation scalaire non linéaire (3.3.1) en un problème linéaire quadratique temps-variant, l'exemple suivant est utilisé :

On considère le système non linéaire donné par les équations

$$\dot{x}_1 = x_1 + x_2^2 + u \quad (3.3.13)$$

$$\dot{x}_2 = x_1^3 + x_2 \quad (3.3.14)$$

Par rapport à la définition du **PE** (3.3.1)-(3.3.2), le système défini par l'équation (3.3.13) représente le **PE 1** où les correspondances suivantes sont définies :

$$\xi \leftrightarrow x_1 \quad ; \quad x^c \leftrightarrow u \quad ; \quad \nu \leftrightarrow x_2^2 \quad (3.3.15)$$

où  $\xi$  est l'état,  $x^c$  est un signal dont l'évolution est connu et  $\nu$  est la commande.

De la même façon, le système dynamique décrit par l'équation (3.3.14) représente le **PE 2** où les correspondances suivantes sont définies :

$$\xi \leftrightarrow x_2 \quad ; \quad x^c \leftrightarrow 0 \quad ; \quad \nu \leftrightarrow x_1^3 \quad (3.3.16)$$

Avec cette façon de définir le **PE 1** et le **PE 2**, les équations (3.3.13) et (3.3.14) deviennent

$$\dot{\xi} = \xi + \nu + x^c \quad (3.3.17)$$

$$\dot{\xi} = \xi + \nu \quad (3.3.18)$$

pour le **PE 1** et le **PE 2** respectivement. En résumé, en choisissant judicieusement la définition de la commande dans chaque problème élémentaire, il est très souvent possible de transformer le problème élémentaire en un problème linéaire quadratique.

Alors, la solution optimale pour chaque **PE** peut être trouvée de la même manière que pour le cas du régulateur linéaire quadratique, à savoir, en résolvant les équations de Riccati temps-variant correspondantes.

Ainsi, pour le système (3.3.17) la trajectoire optimale  $\xi^{opt}$  et la commande optimale  $\nu^{opt}$  sont obtenues à l'itération  $k$ . Par rapport au système de départ (3.3.13), la trajectoire optimale à l'itération  $k$  de l'état  $x_1$  (noté  $x_1^{(k,1)}$ ) est obtenue directement à partir de l'état du système (3.3.17) puisque la correspondance  $\xi \leftrightarrow x_1$  est utilisée.

La trajectoire optimale à l'itération  $k$  de l'état  $x_2$  (notée  $x_2^{(k,1)}$ ) est obtenue à partir de la commande optimale du système (3.3.17), c'est à dire, pour le calcul de  $x_2$  la correspondance  $\nu \leftrightarrow x_2^2$  est utilisée. Cette trajectoire est alors donnée par

$$x_2^{(k,1)} = (\nu^{(k,1)})^{\frac{1}{2}} \quad ; \quad \nu^{(k,1)} \geq 0 \quad (3.3.19)$$

De la même manière, pour le système (3.3.18) la trajectoire optimale  $\xi^{opt}$  et la commande optimale  $\nu^{opt}$  sont obtenues à l'itération  $k$ . Par rapport au système de départ (3.3.14), la trajectoire optimale à l'itération  $k$  de l'état  $x_1$  (noté  $x_1^{(k,2)}$ ) est obtenue à partir de la commande optimale du système (3.3.18), pour le calcul de  $x_1$  la correspondance  $\nu \leftrightarrow x_1^3$  est utilisée. Cette trajectoire est alors donnée par

$$x_1^{(k,2)} = (\nu^{(k,2)})^{\frac{1}{3}}$$

La trajectoire optimale à l'itération  $k$  de l'état  $x_2$  (noté  $x_2^{(k,2)}$ ) est obtenue directement à partir de l'état du système (3.3.18) puisque la correspondance  $\xi \leftrightarrow x_2$  est utilisée.

Par rapport à cette façon de procéder, plusieurs remarques s'imposent:

**Remarque 3.3.3.** Il va de soi que les non linéarités ne disparaissent pas pour autant par magie! En effet, ces non linéarités apparaissent au niveau de l'étape de traitement. Dans le cas de l'exemple ci-dessus en particulier, la solution  $x_2^{k,1}$  est donnée par  $\sqrt{\hat{\nu}}$  où  $\hat{\nu}(\cdot)$  est la solution optimale du problème élémentaire 1. De même que  $x_1^{k,2}$  est donné par  $\hat{\nu}^{1/3}$  où  $\hat{\nu}(\cdot)$  est la solution optimale du problème élémentaire 2.  $\square$

**Remarque 3.3.4.** La Remarque 3.3.3 montre que cette transformation d'un problème élémentaire en un problème linéaire nécessite parfois de modifier la définition des critères quadratiques. Par exemple, pour que la transformation du problème élémentaire 1 de l'exemple précédent soit valide, il faut que la solution optimale du problème linéaire quadratique correspondant soit telle que  $\hat{\nu}(\tau) > 0$  pour tout  $\tau$ . Ceci peut se faire de deux manières

1. Soit en utilisant un terme quadratique du type

$$\| \nu - \nu_0 \|_R^2 \quad ; \quad \nu_0 > 0 \quad (3.3.20)$$

avec  $\nu_0$  et  $R$  suffisamment grand pour assurer la positivité de la solution. Cette solution à l'avantage de la simplicité et l'inconvénient de l'absence de détermination a priori de  $\nu_0$  et de  $R$ . Une variation itérative en fonction des résultants de la solution des équations de Riccati peut s'avérer nécessaire. D'autre part, le problème d'origine est naturellement déformé et la solution du problème modifié diffère de celle du problème de départ.

2. Soit en utilisant explicitement une optimisation quadratique sous contraintes. Cette solution présente l'avantage de résoudre le problème exact directement et l'inconvénient de perdre l'avantage de l'efficacité des solveurs des équations de Riccati qui ne s'accommode pas avec les contraintes sur la commande. □

En résumé, le passage par des problèmes élémentaires à une dynamique linéaire permet de se passer de la résolution des équations aux dérivées partielles non linéaires, mais, il introduit des complications. Le choix d'effectuer ou non cette transformation dépend du gain en temps de calcul que cela apporte. Une conclusion générale semble difficile à prononcer.

La solution au problème du régulateur linéaire optimal est rappelée ci-après.

### Solution au problème du régulateur linéaire optimal[11]

Soit un système linéaire défini par

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (3.3.21)$$

où  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ , et les matrices  $A(t)$  et  $B(t)$  sont de dimension  $(n \times n)$  et  $(n \times m)$  respectivement. Le système est supposé être stabilisable.

Considérons un critère quadratique décrit par

$$J(x(t), u(t), t) = \int_t^{t_f} \left[ x^T(\tau)Q(\tau)x(\tau) + u^T(\tau)R(\tau)u(\tau) \right] d\tau \quad (3.3.22)$$

où  $u(\cdot)$  n'est pas contraint,  $t_f$  est spécifié,  $Q(\cdot)$  est semi-définie positive et  $R(\cdot)$  est définie positive. La solution du problème de commande optimale ci-dessus est donnée par

$$u(t) = -R^{-1}(t)B^T(t)P(t)x(t) \quad (3.3.23)$$

où la matrice symétrique  $P(t)$  de dimension  $(n \times n)$  est l'unique solution de l'équation de Riccati

$$\dot{P}(t) = -P(t)A(t) - A^T(t)P(t) + P(t)B(t)R^{-1}(t)B^T(t)P(t) - Q(t) \quad (3.3.24)$$

satisfaisant la condition terminale

$$P(t_f) = 0$$

La trajectoire optimale de l'état du système est alors la solution de l'équation différentielle linéaire

$$\dot{x}(t) = [A(t) - B(t)R^{-1}(t)B^T(t)P(t)]x(t) \quad (3.3.25)$$

□

**Remarque 3.3.5.** Il est important de noter que les pondérations utilisées dans la définition du critère doivent être choisies (voire modifiée d'une façon itérative) afin de respecter les contraintes du type de celle exprimée dans (3.3.19) qui résultent de la transformation des expressions dépendant des états en commandes<sup>1</sup>.  $\square$

Comme il peut être noté de l'exemple où les équations (3.3.13) et (3.3.14) représentent le **PE 1** et **PE 2** respectivement, deux trajectoires pour  $x_1$  ( $x_1^{(k,j)}$  ( $j \in \{1, 2\}$ ) à l'itération  $k$  seront ainsi obtenues et deux pour  $x_2$  ( $x_2^{(k,j)}$  ( $j \in \{1, 2\}$ ). Chacune de ces trajectoires correspond à la solution retrouvée par les solveurs de chacun des problèmes élémentaires, un traitement des résultats est alors réalisé de façon à obtenir une seule estimation (à l'itération  $k$ ) pour  $x_1$  et  $x_2$ .

De cette façon, le système non linéaire de départ a été transformé en un système linéaire temps-variant. Les solutions optimales sont retrouvées en utilisant l'équation de Riccati (voir les commentaires de la Remarque 3.3.3 et de la Remarque 3.3.4).

D'une manière générale, la Figure 3.2 indique un module de résolution de problèmes élémentaires.

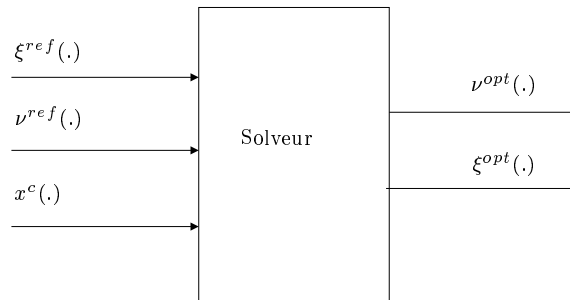


FIG. 3.2: Entrées et sorties du module de résolution de problèmes élémentaires

Les différents modules qui composent l'algorithme de résolution ainsi que les différentes procédures numériques utilisées de la bibliothèque IMSL du compilateur FORTRAN sont présentés dans l'annexe B.

Dans la section suivante, une discussion sur les points stationnaires de l'algorithme aussi comme le calcul des pondérations  $q$  et  $R$  est présentée pour mettre en évidence certaines difficultés liées à l'algorithme.

### 3.4 Points stationnaires et calcul des pondérations

Cette section est divisée en deux parties. Dans la première, une discussion à propos des points stationnaires de l'algorithme est proposée pour montrer quand ces points sont

1. Dans le problème élémentaire 1 de l'exemple précédent,  $x_2^2$  est choisie comme commande dans l'équation différentielle régissant  $x_1$ . Ceci impose une commande positive.

obtenus et comment les résultats associés doivent être interprétés. Dans l'autre partie, une discussion est entamée concernant le calcul des pondérations  $q$  et  $R$  qui explique comment faire varier dynamiquement ces pondérations afin de forcer l'algorithme à converger vers une solution "cohérente".

### 3.4.1 Points stationnaires

Il est rappelé que la méthodologie proposée est une méthode itérative où une configuration (caractérisant tous les signaux concernés) qui explique la sortie à chaque itération est recherchée. Pour ce faire, les solutions de chaque problème élémentaire sont traitées dans la phase de traitement afin d'aboutir à une solution unique concernant chacun des signaux du système à l'issue de l'itération  $k$ . Les solutions de cette phase de traitement ainsi obtenues à l'itération  $k$  se comparent avec les solutions obtenues à l'itération  $k - 1$  pour vérifier si cette différence est plus petit ou égale à un certain seuil de décision donné. Plus spécifiquement, on dira que l'algorithme a abouti à un point stationnaire du problème si à l'itération  $k$ , l'on est dans la situation suivante

$$\forall j \quad \|x_j^{(k)}(\cdot) - x_j^{(k-1)}(\cdot)\|_\infty \leq \epsilon \quad (3.4.1)$$

Ainsi, un point stationnaire est un point tel que l'itération suivante donnera un résultat qui est sensiblement identique au résultat de l'itération actuelle. Il ne servira plus de continuer les itérations. Ceci ne veut pas dire que l'algorithme converge vers une proposition cohérente concernant le problème de l'estimation de l'état. En effet, deux types de points stationnaires peuvent se présenter quand la relation (3.4.1) est satisfaite :

1. Point Stationnaire Cohérent (PSC). Ce type de point stationnaire est obtenu quand les solutions obtenues à chacun des problèmes élémentaires concernant un signal donné sont mutuellement compatibles. Plus précisément pour tout signal  $x_j$  dont les solutions sont proposées par le solveur des problèmes élémentaires  $i_1$  et  $i_2$

$$\forall j \quad \forall (i_1, i_2) \quad \|x_j^{(k, i_1)} - x_j^{(k, i_2)}\| \leq \epsilon \quad (3.4.2)$$

Ceci implique que l'algorithme a trouvé une configuration avec tous les signaux du système qui expliquent le comportement des sorties mesurées.

2. Point Stationnaire Incohérent (PSI). Ce type de point stationnaire est obtenu quand les solutions obtenues de chacun des problèmes élémentaires sont incompatibles entre elles concernant au moins un signal. Ceci correspond à la négation de la condition (3.4.2).

Dans ce cas, on ne peut pas dire que l'algorithme a proposé une configuration cohérente des signaux du système permettant d'expliquer la sortie mesurée.

Pour illustrer ces situations imaginons qu'il existe deux problèmes élémentaires (le système original possède deux états). La solution du problème élémentaire  $i$  à l'itération  $k$  concernant la variable  $x_j$  est représentée par  $x_j^{(k, i)}$  ;  $(i, j \in \{1, 2\})$ . La solution à

l'itération  $k$  après la phase de traitement est représentée par  $x_j^{(k)}$ . Ainsi une proposition cohérente est obtenue quand  $x_j^{(k,1)} = x_j^{(k,2)} = x_j^{(k)}$  et une proposition incohérente quand  $x_j^{(k,1)} \neq x_j^{(k,2)} \neq x_j^{(k)}$ .

Une représentation graphique de ces situations est montrée sur la Figure 3.3

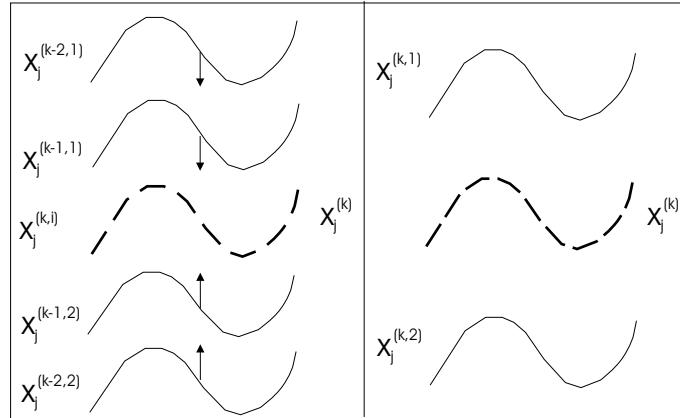


FIG. 3.3: Le schéma de gauche montre une proposition cohérente car un point stationnaire est atteint où toutes les solutions sont égales à l'itération  $k$ , il peut être noté que  $x_j^{(k,1)} = x_j^{(k,2)} = x_j^{(k)}$ . Le schéma de droite montre une proposition incohérente car quand un point stationnaire est atteint les solutions ne sont pas les mêmes à l'itération  $k$ , il peut être noté que  $x_j^{(k,1)} \neq x_j^{(k,2)} \neq x_j^{(k)}$ .

Il est très important de souligner que le caractère stationnaire d'un point ainsi que le qualificatif cohérent ou incohérent d'un point stationnaire sont des propriétés accessibles et ne dépendent pas des signaux non mesurés. En effet, ces propriétés sont définies à partir des signaux de programmation, à savoir  $x_j^{(k,i)}$  ou  $x_j^{(k)}$  qui ne sont que des signaux internes à l'algorithme d'estimation proposé.

Il faut signaler le fait que les points stationnaires sont en relation directe avec le choix des pondérations. Ce fait et le calcul des pondérations sont discutés dans la section suivante.

### 3.4.2 Calcul des pondérations

Une première intuition de calcul des pondérations  $q$  et  $R$  associées au critère

$$J = \int_{t-T}^t \left( q(\xi(\tau) - \xi^{ref}(\tau))^2 + \|\nu(\tau) - \nu^{ref}(\tau)\|_R^2 \right) d\tau$$

a été de laisser les pondérations statiques pendant toutes les itérations en choisissant judicieusement des valeurs grandes pour les pondérations des sorties et en choisissant les



valeurs des pondérations des autres variables en fonction des valeurs des pondérations de la sortie.

Avec cette première démarche, comme il est montré dans la simulation 1 de l'exemple d'illustration, l'algorithme peut arriver à un point stationnaire incohérent (**PSI**) où les trajectoires optimales obtenues de chaque problème élémentaire sont différentes des signaux de référence obtenus de la phase de traitement. Et même si les itérations continuent, la solution reste dans la configuration trouvée.

Pour cette raison, comme il a été souligné dans la Remarque 3.3.5, le calcul des pondérations est réalisé dynamiquement (modifiées d'une façon itérative). Une façon de procéder est la suivante :

- Assigner des valeurs importantes pour les pondérations des sorties mesurées, puisque il s'agit de la seule information dont nous disposons pour reconstruire l'état.
- Pour chaque problème élémentaire, localiser les pondérations qui sont de nature à forcer les itérations vers un point stationnaire cohérent. Ce sont naturellement les pondérations qui pénalisent l'écart par rapport à la solution sortie de la phase de traitement. Si le problème élémentaire en question comporte une sortie, la pondération portant sur cette dernière reste prépondérante et les pondérations portant sur les variables permettant d'obtenir une solution cohérente prennent la deuxième place. En l'absence de variable de sortie, celles-ci prennent la première place.
- Lorsque l'on se trouve dans un point stationnaire incohérent, faire varier ces pondérations dans le bon sens, tout en laissant le cas échéant les pondérations sur les sorties mesurées prépondérantes.

Cette étude est à faire pour chaque problème particulier en prenant en compte ses spécificités.

## 3.5 Discussion

L'algorithme proposé devrait être vu comme un premier pas vers un observateur qui devrait posséder les propriétés suivantes

### 1. Indépendance vis à vis des conditions structurelles

À travers tout le chapitre, il a été vu que les systèmes étudiés peuvent avoir n'importe quel type de structure. Il va sans dire que lorsque le système possède une structure particulière, une mise en œuvre dédiée peut être envisagée permettant l'accélération de l'algorithme. Un cas particulier de tel traitement est celui de l'exemple illustrant la possibilité de transformer les systèmes scalaires non linéaires en des systèmes linéaires. Cependant, l'absence de telle structure particulière n'empêche

pas l'exécution de l'algorithme qui passe dans ce cas par la résolution des équations de Hamilton-Jacobi non linéaires.

## 2. Prise en compte explicite des incertitudes et/ou des perturbations

- (a) La prise en compte de l'existence des signaux inconnus affectant la dynamique du système passe par la minimisation de l'erreur de prédiction de la sortie sur l'ensemble des valeurs possibles de ces signaux. Il est alors déterminant pour la réussite de l'algorithme que ces ensembles de valeurs soient évalués correctement par le concepteur de l'observateur. Ceci passe par une bonne connaissance des niveaux d'incertitudes, des valeurs que prennent les différents paramètres du système à l'occasion de pannes affectant son fonctionnement.
- (b) D'autre part, la prise en compte de ces signaux inconnus par l'algorithme ne doit pas faire oublier que la possibilité de reconstruire l'état et les signaux les affectant n'est pas donnée dans tous les cas. En bref, le système n'est pas forcément "*observable d'une façon robuste*" quelque soit le niveau ou la structure de l'incertitude ou du bruit qui interviennent dans sa dynamique.
- (c) Cependant, il convient de noter que l'algorithme peut jouer un rôle important par rapport à ce problème d'observabilité. En effet, l'algorithme recherche une configuration de tous les signaux du système qui pourraient expliquer le comportement de la sortie. Lorsque dans une phase de simulation, l'algorithme trouve une telle configuration "**cohérente**" avec des signaux erronés, ceci indique justement que le système (avec les incertitudes et les bruits considérés) n'est pas observable. De ce fait, **l'algorithme proposé peut servir d'outil d'analyse dans la phase de conception où il est question de savoir quels sont les capteurs à utiliser, quels sont les paramètres qu'il convient de mesurer avec telle ou telle précision.**

## 3. Implémentation

L'implémentation de l'algorithme de résolution passe par la définition d'un problème élémentaire, ce qui permet de diviser le système original (de dimension  $n$ ) en  $n$  problèmes d'optimisation scalaires. L'avantage de travailler avec des problèmes scalaires est que la résolution des équations de Hamilton-Jacobi scalaires est rendue possible par des codes de calcul efficaces et éprouvés. D'autre part, cette division permet une résolution parallèle de tous les problèmes. Comme il a été décrit auparavant, la solution de chaque **PE** est le rôle d'un solveur et la solution du problème de départ est accomplie à travers une phase de traitement, c'est à dire, les solutions obtenues pour chaque solveur et transmises d'une manière synchronisée sont traitées à chaque itération afin d'obtenir une solution unique à l'issue de l'itération  $k$  (voir la Figure 3.1).

Bien qu'il soit probablement possible d'accélérer l'algorithme présent basé sur la résolution des équations de Hamilton-Jacobi, il est clair que la meilleur

manière d'accélérer le processus de calcul est d'utiliser (quand cela est possible) des équations de Riccati scalaires.

Dans la suite, seul l'algorithme basé sur la résolution des équations de Hamilton-Jacobi est utilisé. L'algorithme basé sur la résolution de l'équation scalaire de Riccati temps-variant reste comme une perspective de ce travail de thèse.

#### 4. Validation

La validation du principe de résolution passe par l'application directe de l'algorithme à un exemple concret ce qui a permis d'avoir une première notion des performances de la méthode. Une fois établie le bon fonctionnement du principe de résolution, les efforts futurs seront concentrés sur des aspects plus techniques comme les études de **convergence**.

Donc, dans la section suivante, un exemple est utilisé afin d'illustrer la méthodologie et la structure de résolution proposée.

### 3.6 Exemple d'illustration

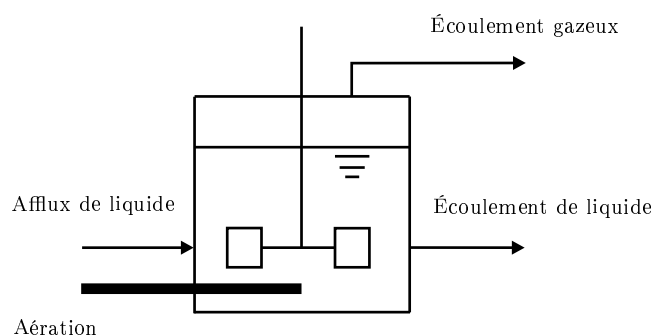
Dans cette section l'algorithme de résolution est appliqué à un bio-réacteur de 4 états, 2 sorties et sous l'influence d'une perturbation.

Tout d'abord, une petite introduction à propos des bio-réacteurs est donnée. Ensuite, la description du bio-réacteur utilisé comme exemple est présentée. Après, l'algorithme de résolution est appliqué au bio-réacteur pour finalement montrer les résultats obtenus des simulations réalisées.

#### 3.6.1 Réacteur continûment agité RCA

Les bio-procédés sont une classe importante de procédés industriels. D'une manière générale, on peut décrire le principe de fonctionnement de ces bio-procédés de la façon suivante: Une population de micro-organismes, la *biomasse* (bactéries, levures, champignons, etc) se développe en consommant certains nutriments, le *substrat* (contenant principalement du carbone et de l'azote) dans un environnement favorable, souvent contrôlé. Ces bio-procédés ont pour objet la production de biomasse ou de produits aussi divers que les protéines, les antibiotiques, l'alcool, les pesticides, etc. Ils peuvent également être utilisés à des fins de dépollution biologique, c'est à dire pour la dégradation de substances polluantes par la bio-masse.

Ainsi, les bioprocédés concernent des secteurs industriels importants, aussi divers

FIG. 3.4: *Réacteur de Réservoir Remué en Continu*

que la pharmacie (production d'antibiotiques, de vaccins, d'hormones, etc), l'agro-alimentaire (vin, bière, fromage, etc), la chimie (solvants, acides) ou l'énergie (carburants).

Le coeur d'un bio-procédé est constitué par un bio-réacteur dans lequel des micro-organismes sont mis à des fins de production ou de bio-transformation. Ainsi, à l'intérieur du bio-réacteur, on doit assurer des conditions telles que les micro-organismes soient à même de se développer et de se renouveler (croissance) et ensuite de réaliser la bio-synthèse désirée.

Un mode de fonctionnement de bio-réacteur généralement employé dans des applications industrielles est le mode continu. Un diagramme schématique d'un réacteur de réservoir remué en continu (**RCA**) est représenté sur la Figure 3.4 Le terme continu signifie que les substrats sont alimentés au réacteur continuellement et qu'un flot de sortie est constamment retiré du système de telle façon que le volume de culture est tenu constant et ce flot de sortie a la même composition que dans le réservoir. Un exemple typique de tels systèmes est celui de procédés d'épuration biologique des eaux usées pour lesquels on cherche généralement à assurer en sortie un effluent de qualité constante quelles que soient les fluctuations sur la pollution à l'entrée.

En raison de la diversité d'application des bio-procédés, la commande et la surveillance des bio-procédés sont devenus des sujets d'une importance croissante pour l'industrie. Cependant ces tâches sont rendues difficiles par le caractère extrêmement non linéaire et non stationnaire de tels processus impliquant la matière organique et par le rareté des capteurs qui peuvent faire des mesures directes en ligne.

Une méthode réaliste de conception de commande de processus est d'employer des estimateurs comme des "*capteurs logiciel*" pour identifier quelques paramètres biologiques clefs. La synthèse d'estimateurs est basée sur l'utilisation de la structure non-linéaire du modèle du processus dans l'étude et sur l'utilisation des mesures indirectes de l'état. La construction de tels estimateurs ou observateurs pour des bio-réacteurs a été largement étudiée [10; 14; 16; 22; 23; 54; 69; 74; 88; 109; 143].

Dans la section suivante le bio-réacteur utilisé comme exemple est présenté.

### 3.6.1.1 Description réacteur continûment agité

Notre objectif est d'appliquer l'algorithme d'estimation proposé précédemment au réacteur continûment agité multi-sortie étudié par [61]. Un diagramme schématique de ce système est présenté sur la Figure 3.5

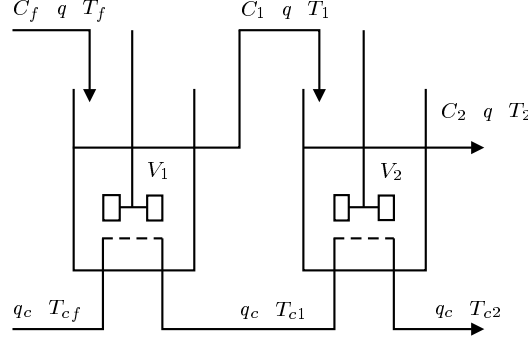


FIG. 3.5: Cascade de deux Réacteurs Continûment Agités

Le système est constitué de deux réacteurs de volumes constants rafraîchis par un flot de liquide de refroidissement simple, c'est à dire, les phases en écoulement circulent dans le même sens. Une réaction irréversible exo-thermique,  $A \rightarrow B$  prend place dans les deux réservoirs. Le modèle du processus consiste en quatre équations différentielles ordinaires non-linéaires :

$$\dot{C}_1 = \frac{q}{V_1}(C_f - C_1) - k_0 C_1 \exp\left(-\frac{E}{RT_1}\right) \quad (3.6.1)$$

$$\begin{aligned} \dot{T}_1 = & \frac{q}{V_1}(T_f - T_1) + \frac{k_0(-\Delta H)C_1}{\rho c_p} \exp\left(-\frac{E}{RT_1}\right) \\ & + \frac{\rho_c c_{pc}}{\rho c_p V_1} q_c \left[1 - \exp\left(-\frac{UA_1}{\rho_c c_{pc} q_c}\right)\right] (T_{cf} - T_1) \end{aligned} \quad (3.6.2)$$

$$\dot{C}_2 = \frac{q}{V_2}(C_1 - C_2) - k_0 C_2 \exp\left(-\frac{E}{RT_2}\right) \quad (3.6.3)$$

$$\begin{aligned} \dot{T}_2 = & \frac{q}{V_2}(T_1 - T_2) + \frac{k_0(-\Delta H)C_2}{\rho c_p} \exp\left(-\frac{E}{RT_2}\right) \\ & + \frac{\rho_c c_{pc}}{\rho c_p V_2} q_c \left[1 - \exp\left(-\frac{UA_2}{\rho_c c_{pc} q_c}\right)\right] [T_1 - T_2 \\ & + \exp\left(-\frac{UA_1}{\rho_c c_{pc} q_c}\right) (T_{cf} - T_1)] \end{aligned} \quad (3.6.4)$$

où

$A_i$	Section de transfert de chaleur pour le réservoir $i$	$m^2$
$C_i$	Concentration	mol/l
$c_p$	Capacité calorifique	J/gK
$E$	Énergie d'activation	J/mol
$\Delta H$	Chaleur de réaction	J/mol
$k_0$	Constante spécifique du taux de réaction	$\text{min}^{-1}$
$q$	Débit d'alimentation	l/min
$R$	Constante du gaz parfait	J/Kmol
$T_i$	Température du réservoir $i$	K
$U$	Coefficient de transfert de chaleur complet	J/minKm <sup>2</sup>
$V_i$	Volume du réservoir $i$	l
$\rho$	masse volumique du contenu du réacteur	g/l
<i>Indice <math>c</math></i>	Liquide de refroidissement	

Les valeurs nominales des paramètres du système sont données dans le Tableau 3.1.

<u>Variable</u>	<u>Valeur nominale</u>
$q$	100 l/min
$V_1 = V_2$	100 l
$C_f$	1 mol/l
$T_f$	350 K
$T_{cf}$	350 K
$k_0$	$7.2 \times 10^{10} \text{ min}^{-1}$
$E/R$	10,000 K
$(-\Delta H)$	$4.78 \times 10^4 \text{ J/mol}$
$\rho = \rho_c$	1000 g/l
$c_p = c_{pc}$	0.239 J/gK
$UA_1 = UA_2$	$1.67 \times 10^5 \text{ J/minK}$

TAB. 3.1: Valeurs des paramètres nominales

Les variables d'état  $x$ , la sortie du système  $y$  et l'entrée manipulée  $u$  sont définies comme suit :

$$x = [C_1 \ T_1 \ C_2 \ T_2]^T, \quad y = [T_1 \ T_2]^T, \quad u = q_c$$

Donc, les équations qui décrivent le modèle dynamique du réacteur continûment agité sont<sup>2</sup>

$$\dot{x}_1 = P_1 - x_1 - k_0 x_1 \exp\left(-\frac{E}{R x_2}\right) \quad (3.6.5)$$

$$\begin{aligned} \dot{x}_2 = & T_f - x_2 + \frac{k_0(-\Delta H)}{\rho c_p} x_1 \exp\left(-\frac{E}{R x_2}\right) \\ & + 0.01u \left[1 - \exp\left(-\frac{U A_1}{\rho c c_{pc} u}\right)\right] (T_{cf} - x_2) \end{aligned} \quad (3.6.6)$$

$$\dot{x}_3 = x_1 - x_3 - k_0 x_3 \exp\left(-\frac{E}{R x_4}\right) \quad (3.6.7)$$

$$\begin{aligned} \dot{x}_4 = & x_2 - x_4 + \frac{k_0(-\Delta H)}{\rho c_p} x_3 \exp\left(-\frac{E}{R x_4}\right) \\ & + 0.01u \left[1 - \exp\left(-\frac{U A_2}{\rho c c_{pc} u}\right)\right] [x_2 - x_4 \\ & + \exp\left(-\frac{U A_1}{\rho c c_{pc} u}\right) (T_{cf} - x_2)] \end{aligned} \quad (3.6.8)$$

où  $P_1$  représente les perturbations non mesurées dans la composition d'alimentation  $C_f$ . Le vecteur d'état  $x$  et le vecteur de sortie  $y$  en fonction des équations du modèle dynamique sont

$$x = [x_1 \ x_2 \ x_3 \ x_4]^T, \quad y = [x_2 \ x_4]^T$$

Les équations normalisées du réacteur continûment agité (3.6.5)-(3.6.8) sont

$$\dot{\bar{x}}_1 = \frac{1}{\bar{x}_1} \left[ P_1 - \bar{x}_1 x_1 - k_0 \bar{x}_1 x_1 \exp\left(-\frac{E}{R \bar{x}_2 x_2}\right) \right] \quad (3.6.9)$$

$$\begin{aligned} \dot{\bar{x}}_2 = & \frac{1}{\bar{x}_2} \left[ T_f - \bar{x}_2 x_2 + \frac{k_0(-\Delta H)}{\rho c_p} \bar{x}_1 x_1 \exp\left(-\frac{E}{R \bar{x}_2 x_2}\right) \right. \\ & \left. + 0.01u \left[1 - \exp\left(-\frac{U A_1}{\rho c c_{pc} u}\right)\right] (T_{cf} - \bar{x}_2 x_2) \right] \end{aligned} \quad (3.6.10)$$

$$\dot{\bar{x}}_3 = \frac{1}{\bar{x}_3} \left[ \bar{x}_1 x_1 - \bar{x}_3 x_3 - k_0 \bar{x}_3 x_3 \exp\left(-\frac{E}{R \bar{x}_4 x_4}\right) \right] \quad (3.6.11)$$

$$\begin{aligned} \dot{\bar{x}}_4 = & \frac{1}{\bar{x}_4} \left[ \bar{x}_2 x_2 - \bar{x}_4 x_4 + \frac{k_0(-\Delta H)}{\rho c_p} \bar{x}_3 x_3 \exp\left(-\frac{E}{R \bar{x}_4 x_4}\right) \right. \\ & \left. + 0.01u \left[1 - \exp\left(-\frac{U A_2}{\rho c c_{pc} u}\right)\right] (\bar{x}_2 x_2 - \bar{x}_4 x_4 \right. \\ & \left. + \exp\left(-\frac{U A_1}{\rho c c_{pc} u}\right) (T_{cf} - \bar{x}_2 x_2)) \right] \end{aligned} \quad (3.6.12)$$

où la variable de référence pour la normalisation est prise égale à  $\bar{x} = [0.1 \ 500 \ 0.01 \ 500]^T$ .

---

2. Après une normalisation du temps  $t' = t/(q/V_1)$

L'application de l'algorithme de résolution sur le réacteur continûment agité défini par les équations normalisées (3.6.9)-(3.6.12) est présentée ci-dessous.

Noter que par rapport à la forme générale des systèmes non linéaires considérés (3.1.1) la perturbation  $\delta$  est représentée par  $\delta(t) = P_1(t)$  et le signal  $u(\cdot)$  est une fonction connue du temps, à savoir  $x^c = u$ .

On suppose que le système fonctionne depuis au moins  $T$  secondes. Ainsi les  $N$  derniers échantillons des mesures des sorties  $y_1$  et  $y_2$  et de l'entrée  $u$  sont disponibles.

$$\begin{aligned} Y_{m1} &:= \left[ y_{m1}(t - Ns), \dots, y_{m1}(t - s) \right]^T \\ Y_{m2} &:= \left[ y_{m2}(t - Ns), \dots, y_{m2}(t - s) \right]^T \\ U &:= \left[ u(t - Ns), \dots, u(t - s) \right]^T \end{aligned}$$

Il est supposé d'autre part qu'une estimation initiale arbitraire des trajectoires de tous les signaux est disponible, plus précisément, on dispose de  $x^{(0)}(\tau)$  et  $\delta^{(0)}(\tau)$ .

Cette information est résumée par les valeurs de ces signaux aux différents instants d'échantillonnage, la fonction correspondante est alors obtenue par simple interpolation du type Spline cubique<sup>3</sup>.

La Figure 3.6 permet de comprendre la logique de l'algorithme proposé dans le cas particulier du système (3.6.9)-(3.6.12).

---

3. procédure DCSIEZ de la bibliothèque IMSL



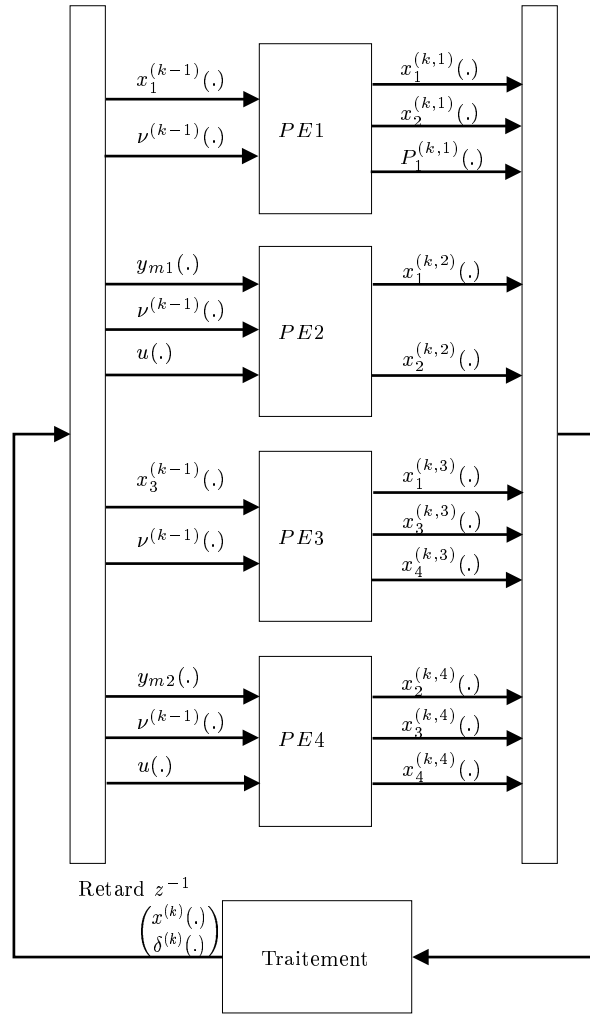


FIG. 3.6: *Illustration du principe de résolution pour le cas particulier du système (3.6.9)-(3.6.12)*

Sur le schéma de la Figure 3.6, les notations  $\eta^{(k,1)}$ ,  $\eta^{(k,2)}$ ,  $\eta^{(k,3)}$  et  $\eta^{(k,4)}$  représentent les résultats (concernant la variable  $\eta$ ) de la résolution à l'itération  $k$  des problèmes 1, 2, 3 et 4 respectivement.

Par rapport à la définition du **PE** (3.3.1)-(3.3.2) les correspondances suivantes sont obtenues :

- Pour le **PE 1** représenté par l'équation (3.6.9)

$$\begin{aligned} \xi &\leftrightarrow x_1 \quad ; \quad x^c \leftrightarrow 0 \quad ; \quad \nu \leftrightarrow \begin{pmatrix} k_0 \bar{x}_1 \exp\left(-\frac{E}{R\bar{x}_2 x_2}\right) \\ P_1 \end{pmatrix} \\ \xi^{ref}(\cdot) &\leftrightarrow x_1^{(k-1)} \quad ; \quad \nu^{ref} \leftrightarrow \begin{pmatrix} k_0 \bar{x}_1 \exp\left(-\frac{E}{R\bar{x}_2 y_1}\right) \\ P_1^{(k-1)} \end{pmatrix} \end{aligned} \quad (3.6.13)$$

Le calcul de l'état  $x_2$  à l'itération  $k$  solution du **PE 1** est donné par

$$x_2^{(k,1)} = -\frac{E}{R\bar{x}_2 \ln\left(\frac{\nu_1}{k_0\bar{x}_1}\right)}$$

- Pour le **PE 2** défini par l'équation (3.6.10)

$$\begin{aligned} \xi \leftrightarrow x_2 \quad ; \quad x^c \leftrightarrow u \quad ; \quad \nu \leftrightarrow \frac{k_0(-\Delta H)}{\rho c_p} \bar{x}_1 x_1 \exp\left(-\frac{E}{R\bar{x}_2 x_2}\right) \\ \xi^{ref}(\cdot) \leftrightarrow y_1 \quad ; \quad \nu^{ref} \leftrightarrow \frac{k_0(-\Delta H)}{\rho c_p} \bar{x}_1 x_1 \exp\left(-\frac{E}{R\bar{x}_2 y_1}\right) \end{aligned} \quad (3.6.14)$$

Le calcul de l'état  $x_1$  à l'itération  $k$  solution du **PE 2** est donné par

$$x_1^{(k,2)} = \frac{\nu}{\frac{k_0(-\Delta H)}{\rho c_p} \bar{x}_1 \exp\left(-\frac{E}{R\bar{x}_2 x_2}\right)}$$

- Pour le **PE 3** défini par l'équation (3.6.11)

$$\begin{aligned} \xi \leftrightarrow x_3 \quad ; \quad x^c \leftrightarrow 0 \quad ; \quad \nu \leftrightarrow \begin{pmatrix} k_0 \bar{x}_3 \exp\left(-\frac{E}{R\bar{x}_4 x_4}\right) \\ x_1 \end{pmatrix} \\ \xi^{ref}(\cdot) \leftrightarrow x_3^{(k-1)} \quad ; \quad \nu^{ref} \leftrightarrow \begin{pmatrix} k_0 \bar{x}_3 \exp\left(-\frac{E}{R\bar{x}_4 y_2}\right) \\ x_1^{(k-1)} \end{pmatrix} \end{aligned} \quad (3.6.15)$$

Le calcul de l'état  $x_4$  à l'itération  $k$  solution du **PE 3** est donné par

$$x_4^{(k,3)} = -\frac{E}{R\bar{x}_4 \ln\left(\frac{\nu_1}{k_0\bar{x}_3}\right)}$$

- Pour le **PE 4** défini par l'équation (3.6.12)

$$\begin{aligned} \xi \leftrightarrow x_4 \quad ; \quad x^c \leftrightarrow u \quad ; \quad \nu \leftrightarrow \begin{pmatrix} \frac{k_0(-\Delta H)}{\rho c_p} \bar{x}_3 x_3 \exp\left(-\frac{E}{R\bar{x}_4 x_4}\right) \\ x_2 \end{pmatrix} \\ \xi^{ref}(\cdot) \leftrightarrow y_2 \quad ; \quad \nu^{ref} \leftrightarrow \begin{pmatrix} \frac{k_0(-\Delta H)}{\rho c_p} \bar{x}_3 x_3 \exp\left(-\frac{E}{R\bar{x}_4 y_2}\right) \\ x_2^{(k-1)} \end{pmatrix} \end{aligned} \quad (3.6.16)$$

Le calcul de l'état  $x_3$  à l'itération  $k$  solution du **PE 4** est donné par

$$x_3^{(k,4)} = \frac{\nu_1}{\frac{k_0(-\Delta H)}{\rho c_p} \bar{x}_3 \exp\left(-\frac{E}{R\bar{x}_4 y_2}\right)}$$

Comme on peut noter, à l'itération  $k$  sont obtenues :

- 3 trajectoires optimales pour l'état  $x_1$ , dénotés par  $x_1^{(k,1)}$ ,  $x_1^{(k,2)}$  et  $x_1^{(k,3)}$  qui proviennent du **PE 1** où  $x_1$  est l'état (dénommé  $\xi$ ), du **PE 2** où  $x_1$  s'obtient à partir de la commande (dénommé  $\nu$ ) et du **PE 3** où  $x_1$  est la commande (dénommé  $\nu_2$ ) sont obtenues.
- 3 trajectoires optimales pour l'état  $x_2$ , dénotés  $x_2^{(k,1)}$ ,  $x_2^{(k,2)}$  et  $x_2^{(k,4)}$  qui proviennent du **PE 1** où  $x_2$  s'obtient à partir de la commande (dénommé  $\nu_1$ ), du **PE 2** où  $x_2$  est l'état (dénommé  $\xi$ ) et du **PE 4** où  $x_2$  est la commande (dénommé  $\nu_2$ ) sont obtenues.
- 2 trajectoires optimales pour l'état  $x_3$ , dénotés  $x_3^{(k,3)}$  et  $x_3^{(k,4)}$  qui proviennent du **PE 3** où  $x_3$  est l'état (dénommé  $\xi$ ) et du **PE 4** où  $x_3$  s'obtient à partir de la commande (dénommé  $\nu_1$ ) sont obtenues.
- 2 trajectoires optimales pour l'état  $x_4$ , dénotés  $x_4^{(k,3)}$  et  $x_4^{(k,4)}$  qui proviennent du **PE 3** où  $x_4$  s'obtient à partir de la commande (dénommé  $\nu_1$ ) et du **PE 4** où  $x_4$  est l'état (dénommé  $\xi$ ) sont obtenues.

Afin d'obtenir une seule estimation des états  $x_1$ ,  $x_2$ ,  $x_3$  et  $x_4$ , la phase de traitement des résultats procurés par chacun des problèmes élémentaires est alors réalisée.

Le bloc de traitement est censé définir les valeurs finalement retenues à l'issue de l'itération  $k$  dans la mesure où certaines grandeurs sont obtenues à la fois pour chacun des **PE**. Une exemple de tel traitement peut être tout simplement la moyenne. Ainsi dans un premier temps, le traitement suivant est adopté pour le cas particulier du système (3.6.9)-(3.6.12)

$$x_1^{(k)} = \frac{x_1^{(k,1)} + x_1^{(k,2)} + x_1^{(k,3)}}{3} \quad (3.6.17)$$

$$x_2^{(k)} = \frac{x_2^{(k,1)} + x_2^{(k,2)} + x_2^{(k,4)}}{3} \quad (3.6.18)$$

$$x_3^{(k)} = \frac{x_3^{(k,3)} + x_3^{(k,4)}}{2} \quad (3.6.19)$$

$$x_4^{(k)} = \frac{x_4^{(k,3)} + x_4^{(k,4)}}{2} \quad (3.6.20)$$

On peut noter que le dénominateur pour les équations (3.6.17)-(3.6.20) correspond au nombre de trajectoires optimales obtenues de chacun des états pour les différents problèmes élémentaires. Finalement, à l'issue de l'itération  $k$ , l'estimation retenue de l'état à l'instant  $t$  est donnée par  $x^{(k)}(t)$ .

Comment il a été décrit auparavant, l'algorithme utilise chaque équation du système comme s'il s'agissait de systèmes indépendants. Le couplage est alors introduit au

moyen des signaux de références afin de forcer la solution finale à être compatible avec toutes les équations du système.

**Remarque 3.6.1.** Une version plus sophistiquée de ce traitement peut être basée sur la qualité des optima issus de la résolution des différents problèmes élémentaires. Un indicateur de cette qualité peut être la dérivée seconde de la solution optimale. En effet

$$\frac{\partial^2 V}{\partial \xi^2}(0, x_i^{(k,j)}) \quad (3.6.21)$$

peut être vu comme un indicateur de la qualité de la solution obtenue pour le problème élémentaire  $j$  à l'itération  $k$ . En effet, plus la dérivée seconde est importante, plus le problème élémentaire correspondant est bien conditionné et donc son poids dans le traitement doit être important. S'agissant d'un travail exploratoire, la règle la plus simple de la moyenne arithmétique a été retenue pour l'instant.  $\square$

Dans la section suivante, les résultats obtenus par l'algorithme de résolution appliqué au système (3.6.9)-(3.6.12) sont présentés.

### 3.6.1.2 Résultats

Dans cette partie, les différentes simulations effectuées pour le système (3.6.9)-(3.6.12) sont présentées. Pour les différentes simulations les conditions suivantes sont utilisées :

- Les conditions initiales normalisées sont  $\bar{x}(0) = [0.85 \ 0.884 \ 0.5 \ 0.9]^T$
- Le temps initial est  $t_0 = 0$  et le temps final est  $t_f = 10$ .
- L'entrée  $u$  est donnée par

$$u = \begin{cases} 90 \text{ l/min,} & 0 \leq t < 5 \\ 100 \text{ l/min,} & t \geq 5 \end{cases} \quad (3.6.22)$$

- La perturbation  $P_1$  est donné par

$$P_1 = \begin{cases} 1, & t \leq 0.1t_f \\ \frac{t-0.1t_f}{0.1t_f}0.1 + 1, & 0.1t_f < t \leq 0.2t_f \\ 1.1, & 0.2t_f < t \end{cases} \quad (3.6.23)$$

- Les valeurs réelles positives  $q$  qui servent comme coefficients de pondération pour la définition du critère (3.3.2) de chaque **PE** sont

$$q = [10 \ 100 \ 10 \ 100]^T$$

- Les valeurs réelles positives  $r$  qui servent comme coefficients de pondération pour la définition du critère (3.3.2) de chaque **PE** sont

$$R = \begin{bmatrix} 100 & 1 \\ 10 & 0 \\ 10 & 100 \\ 100 & 10 \end{bmatrix}$$

En appliquant les règles élémentaires du traitement des pondérations énoncées dans la section 3.4.2, il est très facile de voir que seule la valeur de la pondération  $R(1,2)$  est à modifier automatiquement chaque fois que l'algorithme arrive à un point stationnaire incohérent (**PSI**). La nouvelle valeur de  $R(1,2)$  est donnée par  $R(1,2)^{\text{(nouvel)}} = \alpha R(1,2)^{\text{(initial)}}$  où la valeur de  $\alpha$  est donnée par  $\alpha = \alpha/10$  avec la valeur initiale de  $\alpha$  égale à 1.

Dans le Tableau 3.2 sont montrés quelques caractéristiques utilisées pour les différentes simulations.

<u>Simulation</u>	<u>Conditions initiales</u>	<u>Valeurs des Pondérations</u>	<u>Sorties Bruitées</u>
1	$\bar{x}_1^{(0)} = 0.95$ $\bar{x}_3^{(0)} = 0.4$	Restent fixes pendant toutes les itérations	Non
2	$\bar{x}_1^{(0)} = 0.95$ $\bar{x}_3^{(0)} = 0.4$	Changent dynamiquement	Non
3	$\bar{x}_1^{(0)} = 0.95$ $\bar{x}_3^{(0)} = 0.4$	Changent dynamiquement	Oui
4	Le signal de référence initial pour $\bar{x}_1$ est la fonction constante 0.7 Le signal de référence initial pour $\bar{x}_3$ est la fonction constante 0.3	Changent dynamiquement	Non

TAB. 3.2: *Les différentes simulations réalisées.*

La notation  $\bar{x}_j^{(0)}$  utilisée pour les conditions initiales correspond au cas où les références des signaux inconnus sont obtenues par intégration du système à partir des conditions initiales  $\bar{x}_j^{(0)}$ . Ceci est à distinguer de la valeur initiale réelle de l'état  $x(0)$ .

**Remarque 3.6.2.** Les différentes simulations commencent effectivement en  $t_0 = 0$  avec les conditions initiales montrées dans le Tableau 3.2, mais toutes les figures montrent les résultats obtenus dans la fenêtre de temps de 1 à 9  $\square$

### a. Simulation 1

Pour cette première simulation la valeur de  $R(1, 2)$  reste fixe même si

$$\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_{\infty} \leq \epsilon \quad ; \quad \epsilon = 1 \times 10^{-4}$$

est un (PSI). Cette valeur fixe est  $R(1, 2) = 1$

La simulation réalisée est décrite et les résultats obtenus sont présentés ci-dessous.

Sur la Figure 3.7 et la Figure 3.8 sont montrées l'estimation de la trajectoire de l'état  $x_2$  qui est la sortie mesurée  $y_1$  et la trajectoire de l'état  $x_4$  qui est la sortie mesurée  $y_2$  respectivement. On peut noter que les trajectoires optimales de l'état  $x_2$  et  $x_4$  sont déjà obtenues à la première itération (après la phase de traitement) car les références sont la sortie mesurée  $x_2 = y_1$  et  $x_4 = y_2$ . Ces estimations ne bougent plus d'une façon significative, donc elles ne sont plus présentées aux itérations suivantes.

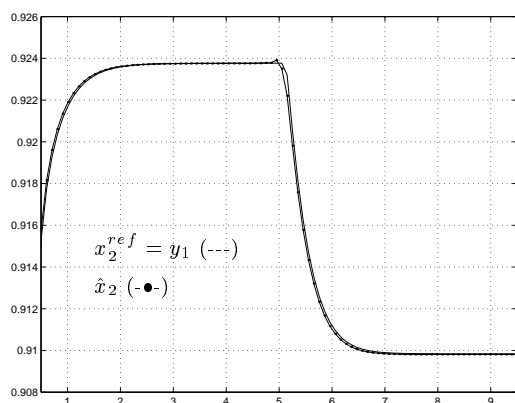


FIG. 3.7: Estimation de l'état  $x_2$

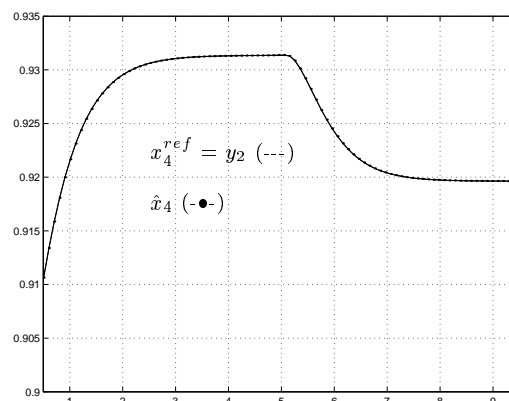


FIG. 3.8: Estimation de l'état  $x_4$

Sur la Figure 3.9 est montrée l'estimation de la trajectoire de l'état  $x_1$  à différentes itérations. Ces estimations sont calculées à partir de la phase de traitement, quand les valeurs des trajectoires de l'état  $x_1$  de chaque **PE** sont obtenues.

À l'itération 1, l'estimation de l'état  $x_1$  ne converge pas à la vraie trajectoire de l'état  $x_1$ , ceci dû à la référence  $x_1^{ref} = x_1^{(0)}$  et à l'influence de la perturbation sur le système. À partir de l'itération 2, le signal de référence est la trajectoire obtenue de la phase de traitement à l'itération  $(k - 1)$ . À l'itération 3, l'estimation commence se rapprocher de la vraie trajectoire de l'état  $x_1$ . À l'itération 8, la norme infinie de l'erreur entre la solution de la phase de traitement à l'itération 8 et la solution de la phase de traitement à l'itération 7 est plus petite que  $\epsilon$

$$\|x^{(8)}(\cdot) - x^{(7)}(\cdot)\|_{\infty} \leq \epsilon \quad ; \quad \epsilon = 1 \times 10^{-4}$$

(voir la Figure 3.15 où l'évolution de  $\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_{\infty}$  est tracée). Ceci implique qu'un point stationnaire a été atteint, donc les trajectoires qui sont obtenues de la phase de traitement ne bougent plus aux itérations suivantes (la trajectoire obtenue à l'itération 10, qui est montrée sur la Figure 3.9, est égale à la trajectoire obtenue à l'itération 8). Donc la valeur de l'estimation de l'état  $x_1$  n'est plus modifiée et cette estimation n'aboutit pas à la vraie trajectoire de l'état  $x_1$  parce que, avec cette valeur de  $R(1, 2)$ , le système "préfère" diminuer l'erreur entre la perturbation  $P_1$  et sa référence (voir la Figure 3.13 à l'itération 10) pour respecter la sortie mesurée  $y_1 = x_2$  et  $y_2 = x_4$ . L'algorithme est resté dans une configuration qui est une proposition incohérente.

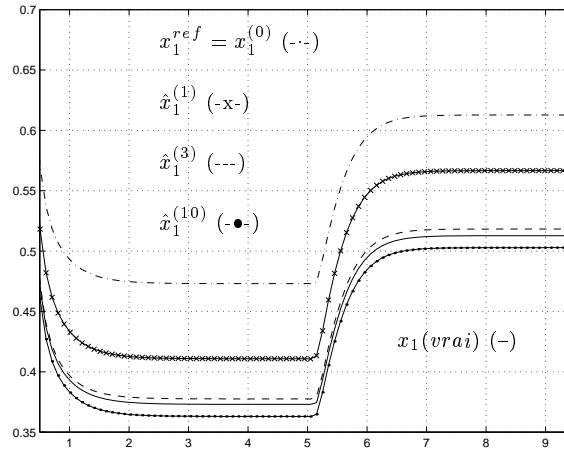


FIG. 3.9: Estimation de l'état  $x_1$  à différentes itérations quand la valeur du coefficient de pondération  $R(1, 2) = 1$  reste fixe. L'algorithme est resté dans une configuration qui est une proposition incohérente.

La Figure 3.10 montre l'erreur entre l'estimation (obtenue à partir de la phase de traitement) et la vraie trajectoire de l'état  $x_1$  à différentes itérations.

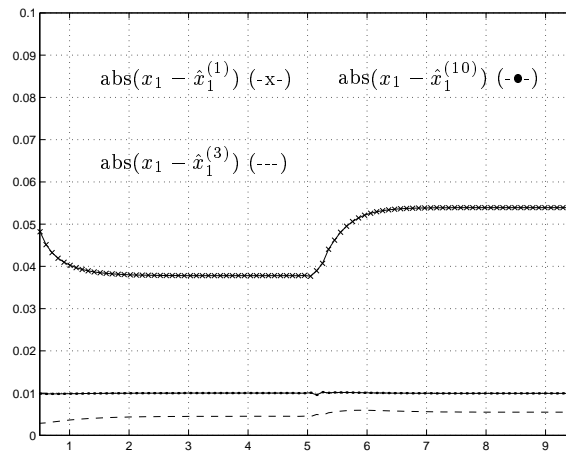


FIG. 3.10: *L'erreur entre l'estimation et la trajectoire vraie de l'état  $x_1$  à différentes itérations quand la valeur du coefficient de pondération  $R(1, 2) = 1$  reste fixe*

Sur la Figure 3.11 est montrée l'estimation de la trajectoire de l'état  $x_3$  à différentes itérations. Ces itérations sont calculées à partir de la phase de traitement, quand les valeurs de l'état  $x_3$  de chaque **PE** sont obtenues. À l'itération 1, l'estimation de l'état  $x_3$  est proche de la vraie trajectoire de l'état  $x_3$  malgré la mauvaise référence  $x_3^{ref} = x_3^{(0)}$  donnée. À partir de l'itération 2, le signal de référence est la trajectoire obtenue de la phase de traitement à l'itération  $(k - 1)$ . À l'itération 10, l'estimation de  $x_3$  est presque obtenue, mais cette estimation n'a pas bougé depuis de l'itération 8.

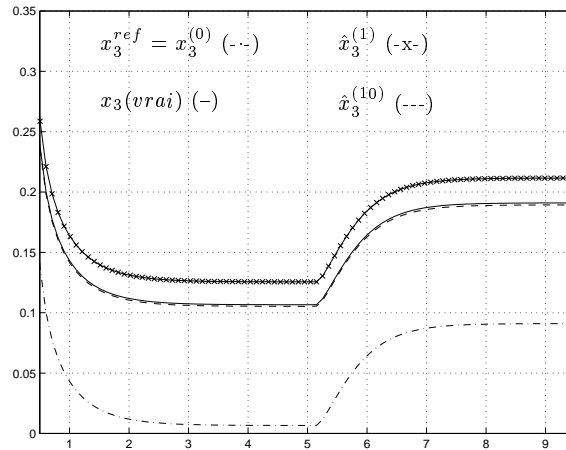


FIG. 3.11: *Estimation de l'état  $x_3$  à différentes itérations quand la valeur du coefficient de pondération  $R(1, 2) = 1$  reste fixe. L'algorithme est resté dans un (PSI)*

La Figure 3.12 montre l'erreur entre l'estimation (obtenue à partir de la phase de traitement) et la vraie trajectoire de l'état  $x_3$  à différentes itérations.



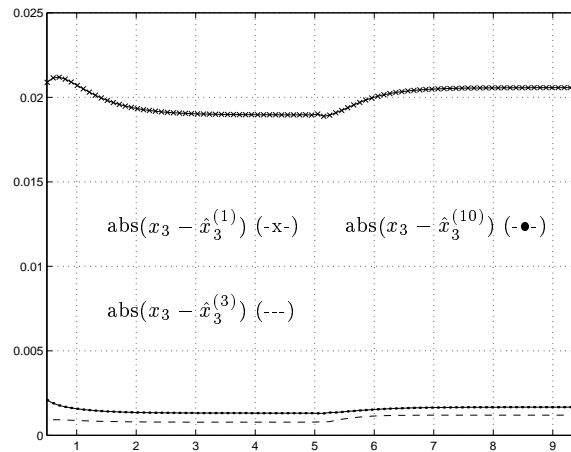


FIG. 3.12: L'erreur entre l'estimation et la trajectoire vraie de l'état  $x_3$  à différentes itérations quand la valeur du coefficient de pondération  $R(1, 2) = 1$  reste fixe

Sur la Figure 3.13 est montrée l'estimation de la trajectoire de la perturbation  $P_1$  à différentes itérations. Ces estimations sont calculées dans le **PE 1** où la perturbation  $P_1$  est une commande, dénoté  $\nu_2$ . Les estimations de  $P_1$  inter-agissent avec les estimations des états  $x_1$  et  $x_3$  à chaque itération afin de respecter toujours les sorties mesurées  $y_1 = x_2$  et  $y_2 = x_4$ , cependant pour cette simulation, l'algorithme se retrouve dans un (**PSI**) et comme la valeur de la pondération  $R(1, 2)$  est restée fixe, l'algorithme a seulement trouvé une proposition incohérente. La référence, dénoté  $\nu^{ref}$ , donnée à la perturbation est  $\nu^{ref} = 1$ . On peut noter que l'estimation de  $P_1$  est plus proche de la référence donnée que de la vraie trajectoire de la perturbation. Sur la Figure 3.14 est montrée l'erreur entre la perturbation réelle  $P_1$  et les trajectoires estimées  $\hat{P}_1$  à différentes itérations.

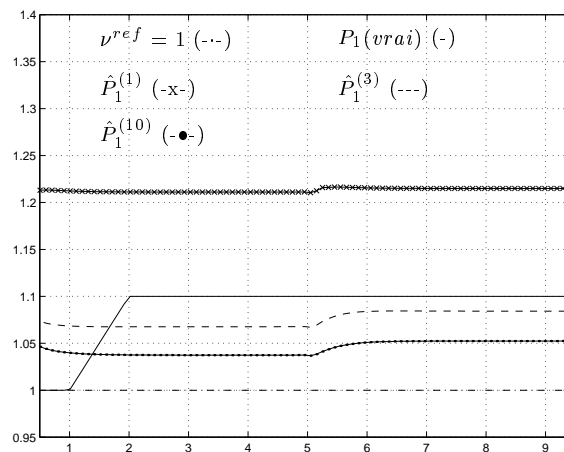


FIG. 3.13: Estimation de la perturbation  $P_1$  à différentes itérations quand la valeur du coefficient de pondération  $R(1, 2) = 1$  reste fixe. L'algorithme a atteint un (**PSI**) et a seulement trouvé une proposition incohérente.

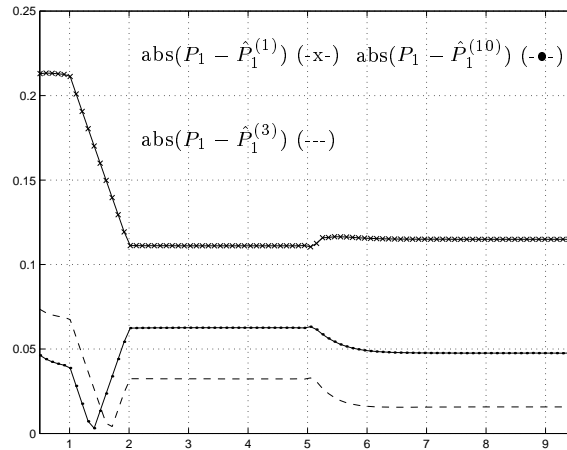


FIG. 3.14: L'erreur entre l'estimation et la trajectoire vraie de la perturbation  $P_1$  à différentes itérations quand la valeur du coefficient de pondération  $R(1, 2) = 1$  reste fixe

Sur la Figure 3.15 est présentée l'évolution de la norme infinie de l'erreur entre la phase de traitement à l'itération  $k$  et la phase de traitement à l'itération  $(k - 1)$ , à savoir  $\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_\infty$

La figure supérieure montre cette évolution de l'erreur dans une échelle normalisée, la figure inférieure montre cette évolution de l'erreur dans une échelle logarithmique. On peut noter dans la figure inférieure que à l'itération 8 l'erreur est plus petite que  $1 \times 10^{-4}$  ceci montre qu'un point stationnaire de l'algorithme a été trouvé, par suite pour les itérations suivantes (9,10), les trajectoires obtenues à partir de la phase de traitement ne bougent plus (comme il peut être noté à l'itération 10 de la Figure 3.9 et de la Figure 3.11) et l'erreur reste plus petite que  $1 \times 10^{-4}$  ce qui indique que l'algorithme se maintient dans le (PSI).

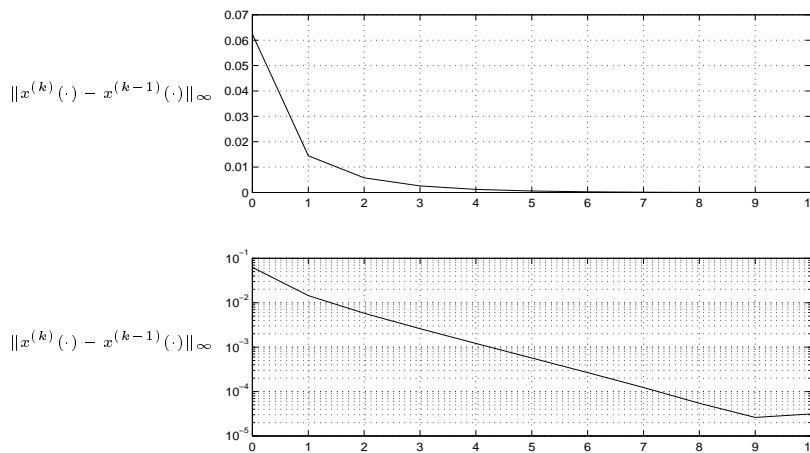


FIG. 3.15: L'évolution de  $\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_\infty$  de la phase de traitement par rapport aux itérations. Échelle normale et échelle logarithmique

### b. Simulation 2

La simulation précédente est répétée mais en changeant la valeur du coefficient  $R(1, 2)$  (celui qui pénalise à la perturbation) de manière automatique quand un **(PSI)** a été atteint.

Sur la Figure 3.16 et la Figure 3.17 sont montrées l'estimation de la trajectoire de l'état  $x_2$  qui est la sortie mesurée  $y_1$  et la trajectoire de l'état  $x_4$  qui est la sortie mesurée  $y_2$  respectivement. Comme dans la Simulation 1, les trajectoires optimales de l'état  $x_2$  et  $x_4$  sont déjà obtenues à la première itération (après de la phase de traitement) donc elles ne sont plus présentées aux itérations suivantes.

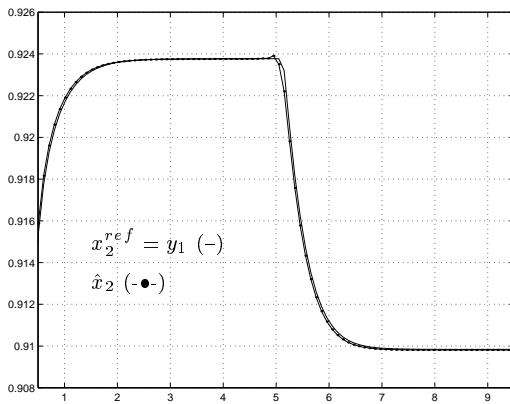


FIG. 3.16: Estimation de l'état  $x_2$

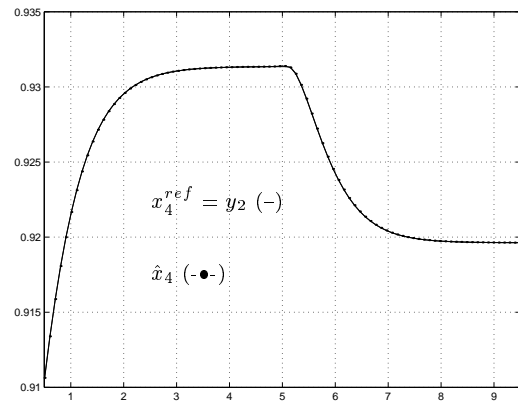


FIG. 3.17: Estimation de l'état  $x_4$

Sur la Figure 3.18 est montrée l'estimation de la trajectoire de l'état  $x_1$  à différentes itérations. Ces estimations sont calculées à partir de la phase de traitement, quand les valeurs des trajectoires de l'état  $x_1$  de chaque **PE** sont obtenues.

Comme dans la Simulation 1, l'estimation de l'état  $x_1$  à l'itération 1 ne converge encore à sa vraie trajectoire. Le signal de référence est à partir de l'itération 2, la trajectoire obtenue de la phase de traitement à l'itération  $(k - 1)$ .

De la même manière que dans la Simulation 1, à l'itération 8, la norme infinie de l'erreur entre la solution de la phase de traitement à l'itération 8 et la solution de la phase de traitement à l'itération 7 est plus petite que  $\epsilon$ , à savoir,  $\|x^{(8)}(\cdot) - x^{(7)}(\cdot)\|_\infty \leq \epsilon$  ;  $\epsilon = 1 \times 10^{-4}$  ceci signifie que l'algorithme a arrivé à un point stationnaire. Les solutions de l'état  $x_1$  des différents problèmes élémentaires ne sont pas égales et sont aussi différentes de la solution obtenue de la phase de traitement dans ce point stationnaire, donc le point stationnaire trouvé est un **(PSI)**.

Cependant, cette fois à différence du cas précédente (Simulation 1) la valeur de  $R(1, 2)$  change, et cette nouvelle valeur est  $R(1, 2) = 0.1$  par conséquent le calcul de l'estimation de l'état  $x_1$  continue.

À l'itération 15, un nouveau (**PSI**) est rencontré, alors la valeur de  $R(1,2)$  change autre fois et cette nouvelle valeur est  $R(1,2) = 0.01$  (voir la Figure 3.24 et la Figure 3.25 pour apprécier l'évolution de  $\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_\infty$  et de la valeur de  $R(1,2)$ ).

Pour ce cas, l'estimation de l'état  $x_1$  est atteinte à l'itération 20, car à cette itération l'algorithme est arrivé à un (**PSC**), sur la Figure 3.18 les tracés de l'estimation et de la vraie trajectoire de l'état  $x_1$  sont complètement confondus. Il peut être constaté que avec un changement dynamique de la valeur de  $R(1,2)$  l'algorithme aboutit à estimer l'état  $x_1$ , ce qu'il n'arrive pas si la valeur de  $R(1,2)$  reste fixe. (Comparer la différence entre les résultats obtenus pour la Simulation 2 et la Simulation 1 sur la Figure 3.18 et la Figure 3.9 respectivement).

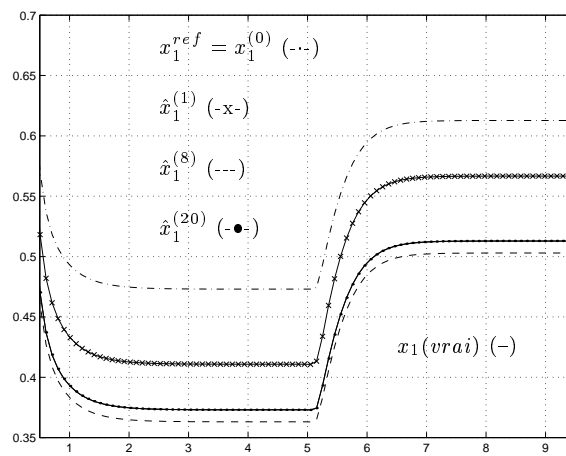


FIG. 3.18: Estimation de l'état  $x_1$  à différentes itérations

La Figure 3.19 montre l'erreur entre l'estimation (obtenue à partir de la phase de traitement) et la vraie trajectoire de l'état  $x_1$  à différentes itérations.

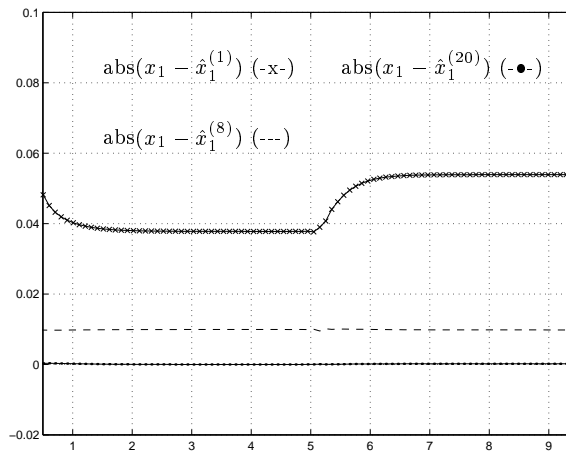


FIG. 3.19: L'erreur entre l'estimation et la trajectoire vraie de l'état  $x_1$  à différentes itérations

Sur la Figure 3.20 est montrée l'estimation de la trajectoire de l'état  $x_3$  à partir de la phase de traitement, quand les valeurs de l'état  $x_3$  de chaque **PE** à différentes itérations sont obtenues. À l'itération 1, l'estimation de l'état  $x_3$  est proche de la vraie trajectoire de l'état  $x_3$ , malgré la mauvaise référence  $x_3^{ref} = x_3^{(0)}$  donnée. Comme dans la Simulation 1, le signal de référence est à partir de l'itération 2, la trajectoire obtenue de la phase de traitement à l'itération  $(k-1)$ . Cette fois, la valeur de  $R(1, 2)$  change de façon automatique quand un **(PSI)** de l'algorithme est atteint. Alors, le calcul de l'estimation de la trajectoire de l'état  $x_3$  est continué. L'estimation de  $x_3$  est atteinte à l'itération 20 comme pour le cas de l'état  $x_1$ , car cette fois un **(PSC)** est trouvé. Les tracés de l'estimation à l'itération 20 et de la vraie trajectoire de l'état  $x_3$  sont complètement confondus comme il peut être remarqué sur la Figure 3.20.

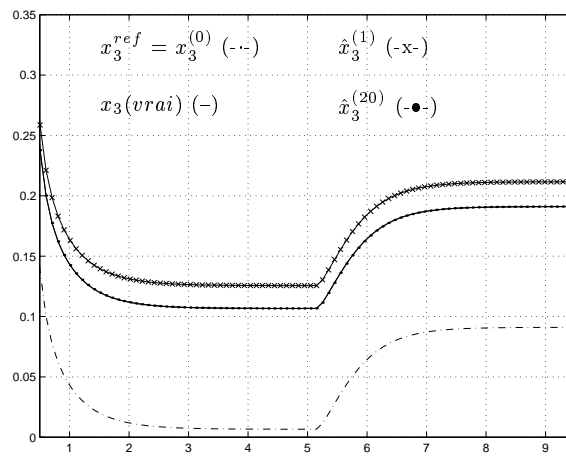


FIG. 3.20: Estimation de l'état  $x_3$  à différentes itérations

La Figure 3.21 montre l'erreur entre l'estimation (obtenue à partir de la phase de traitement) et la vraie trajectoire de l'état  $x_3$  à différentes itérations.

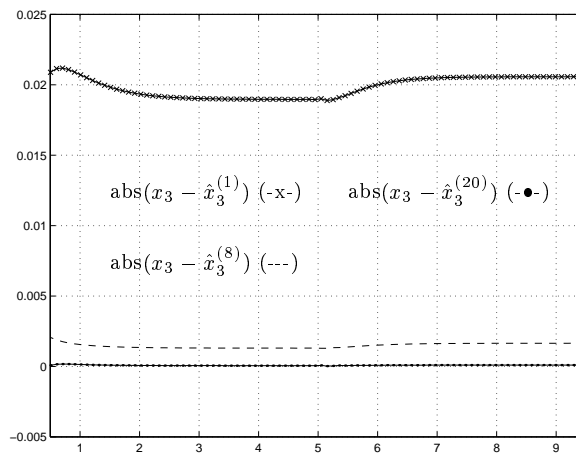


FIG. 3.21: L'erreur entre l'estimation et la trajectoire vraie de l'état  $x_3$  à différentes itérations

Sur la Figure 3.22 est montrée l'estimation de la trajectoire de la perturbation  $P_1$  à différentes itérations. Ces estimations sont calculées dans le **PE 1** où la perturbation  $P_1$  est une commande, dénoté  $\nu_2$ . Les estimations de  $P_1$  inter-agissent avec les estimations des états  $x_1$  et  $x_3$  à chaque itération afin de respecter toujours les sorties mesurées  $y_1 = x_2$  et  $y_2 = x_4$  et cette fois, l'algorithme a trouvé une proposition cohérente. La référence, dénoté  $\nu^{ref}$ , donnée à la perturbation est comme dans la Simulation 1, à savoir,  $\nu^{ref} = 1$ . Pour ce cas, la valeur de  $R(1, 2)$  qui pénalise précisément à la perturbation change automatiquement quand l'algorithme se trouve dans un (**PSI**). Il peut être noté qu'une estimation de  $P_1$  n'est pas complètement atteinte, cependant une trajectoire qui reflète globalement le comportement de  $P_1$  est obtenue, ce qu'il ne se produit pas dans la Simulation 1 (voir la Figure 3.13) où l'estimation se "dirige" vers sa référence au lieu de la vraie trajectoire  $P_1$ . Sur la Figure 3.23 est montrée l'erreur entre la perturbation réelle  $P_1$  et les trajectoires estimées  $\hat{P}_1$  à différentes itérations.

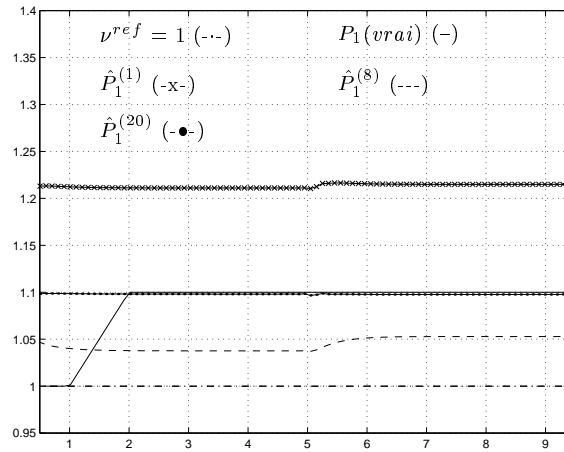


FIG. 3.22: Estimation de la perturbation  $P_1$  à différentes itérations

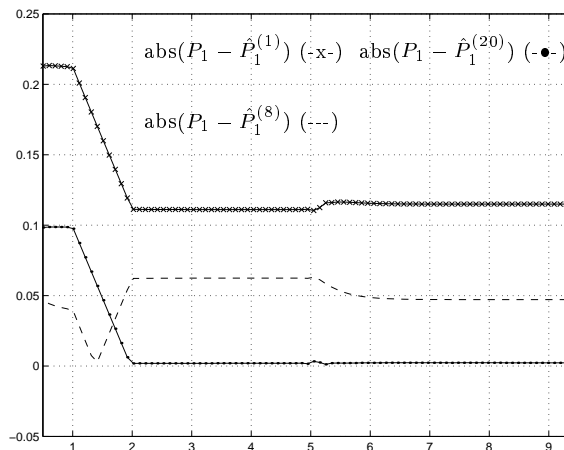


FIG. 3.23: L'erreur entre l'estimation et la trajectoire vraie de la perturbation  $P_1$  à différentes itérations

Sur la Figure 3.24 (échelle normalisée -figure supérieure- échelle logarithmique -figure inférieure-) l'évolution de la norme infinie de l'erreur entre la phase de traitement à l'itération  $k$  et la phase de traitement à l'itération  $(k-1)$  est présentée et il peut être remarqué clairement qui est aux itérations 8, 15 et 20 quand

$$\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_{\infty} \leq \epsilon$$

Il peut être noté que l'erreur a augmenté à l'itération 9. Ceci est dû au fait qu'avec la modification de la valeur de la pondération  $R(1,2)$ , les sorties du module de traitement ont changé conduisant à la correction des signaux des référence des problèmes élémentaires. Ceci est justement l'objectif de cette modification de la pondération  $R(1,2)$ . La même situation se produit à l'itération 16.

Sur la Figure 3.25, l'évolution de la valeur de  $R(1,2)$  par rapport au nombre d'itérations est montrée. La valeur de  $R(1,2)$  a changé aux itérations 8 et 15 (quand l'algorithme se trouve dans un **(PSI)**) mais cette nouvelle valeur (0.1 et 0.01) est utilisée à partir des itérations 9 et 16 respectivement.

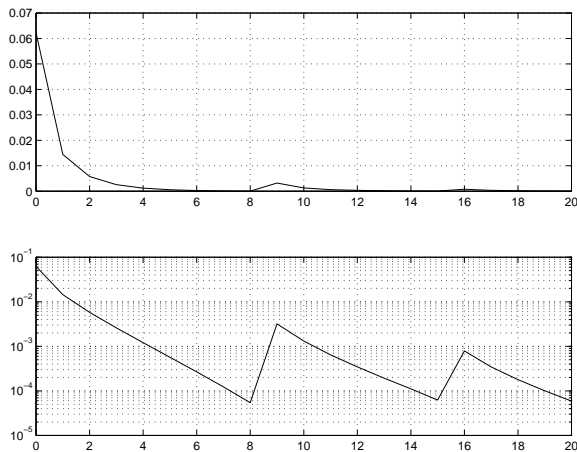


FIG. 3.24: *L'évolution de  $\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_{\infty}$  par rapport aux itérations*

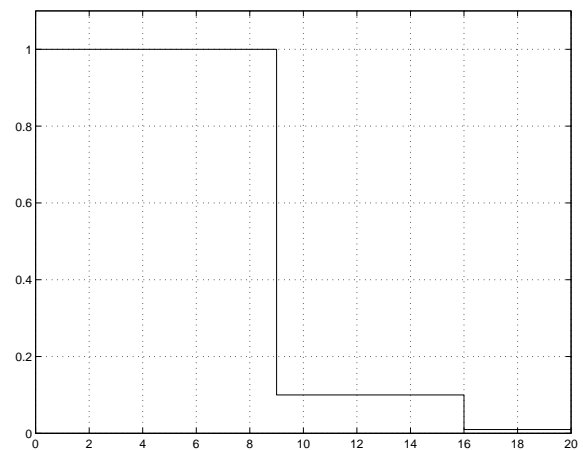


FIG. 3.25: *L'évolution de la valeur de  $R(1,2)$  par rapport aux itérations*

### c. Simulation 3

Cette simulation est une extension de la simulation précédente dans le sens où dans cette simulation les sorties sont contaminées avec un bruit de mesure. La génération du bruit est faite en utilisant la procédure DRNUN<sup>4</sup> de la bibliothèque IMSL du compilateur FORTRAN. Le vecteur du bruit de mesure est donné par

$$Brm(t) = 2b_{max}(br(t) - 0.5)$$

où  $br(t)$  est la sortie de DRNUN et  $b_{max} = 1 \times 10^{-3}$ .

L'écart-type (standard déviation) de  $Brm(t)$  pour la sortie  $y_1 = x_2$  est égal à  $8 \times 10^{-3}$  et l'écart-type (standard déviation) de  $Brm(t)$  pour la sortie  $y_2 = x_4$  est égal à  $7 \times 10^{-3}$ .

Il faut signaler qu'il existera une étape de filtrage des sorties avant de réintroduire les signaux des références dans l'algorithme de résolution. Cette simulation est réalisée pour tester la *robustesse* au bruit de mesure de l'algorithme même sans l'étape de filtrage.

Cependant, déjà à cause du bruit, il existe des problèmes numériques avec le solveur des équations différentielles partielles (le solveur de l'équation de Hamilton-Jacobi) si la simulation commence avec une valeur de  $R(1, 2) = 1$ . Pour cette raison, la valeur de  $R(1, 2)$  est diminuée, cette valeur est  $R(1, 2) = 0.1$ .

L'estimation de la trajectoire de l'état  $x_2$  qui est la sortie mesurée bruitée  $y_1$  et la trajectoire de l'état  $x_4$  qui est la sortie mesurée bruitée  $y_2$  sont montrées sur la Figure 3.26 et la Figure 3.27 respectivement. Il peut être apprécié que les trajectoires optimales de l'état  $x_2$  et  $x_4$  sont obtenues de manière acceptable à la première itération, malgré l'effet du bruit de mesure. Ces estimations ne bougent plus d'une forme significative aux itérations suivantes, donc elles ne sont plus présentées.

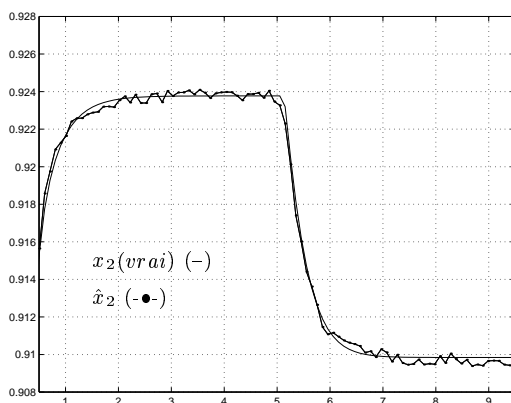


FIG. 3.26: Estimation de l'état  $x_2$  sous l'influence du bruit de mesure

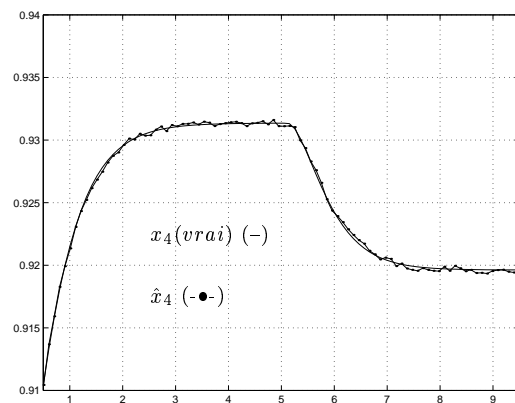


FIG. 3.27: Estimation de l'état  $x_4$  sous l'influence du bruit de mesure

4. Cette procédure produit des nombres aléatoires d'une distribution uniforme (0,1)



Sur la Figure 3.28 est montrée l'estimation de la trajectoire de l'état  $x_1$  à différentes itérations. Dans cette simulation l'estimation de l'état  $x_1$  à l'itération 1 est plus proche de la référence initiale  $x_1^{ref} = x_1^{(0)}$  que dans la Simulation 2 (voir Figure 3.18) dû à la valeur de  $R(1, 2)$  donnée. Comme dans les deux cas précédents, à partir de l'itération 2, le signal de référence est la trajectoire obtenue de la phase de traitement à l'itération  $(k-1)$ .

L'influence de la sortie bruitée  $y_1 = x_2$  sur l'estimation de la trajectoire de l'état  $x_1$  peut être constatée à l'itération 7. Un **(PSI)** de l'algorithme arrive à l'itération 14 et à l'itération 19 (dans les simulations précédentes le **(PSI)** est atteint à l'itération 8 et à l'itération 15) ce qui signifie que la valeur de  $R(1, 2)$  change deux fois. Ces valeurs de  $R(1, 2)$  sont 0.01 et 0.001 respectivement.

Pour apprécier l'évolution de  $\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_\infty$  et de la valeur de  $R(1, 2)$  voir la Figure 3.34 et la Figure 3.35 respectivement.

À l'itération 25 un **(PSC)** est atteint. Il peut être noté que l'estimation de l'état  $x_1$  est atteinte malgré le bruit de mesure et les tracés de l'estimation et de la vraie trajectoire de  $x_1$  sont complètement confondus.

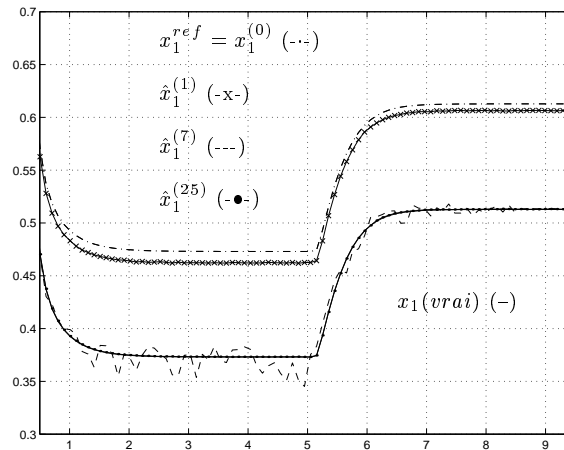


FIG. 3.28: Estimation de l'état  $x_1$  à différentes itérations quand les sorties  $y_1 = x_2$  et  $y_2 = x_4$  sont sous l'influence du bruit de mesure

La Figure 3.29 montre l'erreur entre l'estimation (obtenue à partir de la phase de traitement) et la vraie trajectoire de l'état  $x_1$  à différentes itérations. On peut apercevoir sur cette figure l'erreur causée par la présence du bruit de mesure (itération 7).

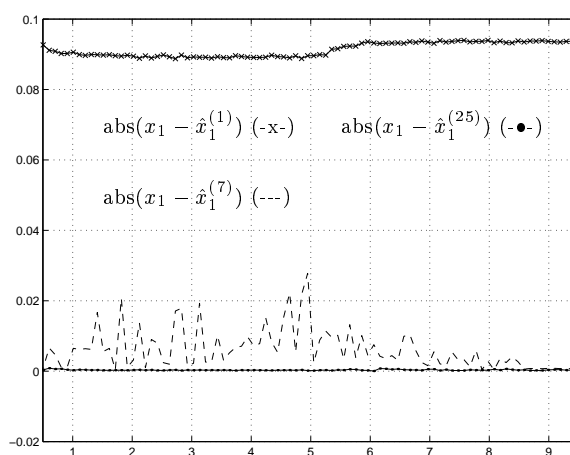


FIG. 3.29: L'erreur entre l'estimation et la trajectoire vraie de l'état  $x_1$  à différentes itérations quand les sorties  $y_1 = x_2$  et  $y_2 = x_4$  sont sous l'influence du bruit de mesure

Sur la Figure 3.30 est montrée l'estimation de la trajectoire de l'état  $x_3$  à partir de la phase de traitement, quand les valeurs de l'état  $x_3$  de chaque **PE** à différentes itérations sont obtenues.

À l'itération 1, il peut être remarqué que l'estimation de l'état  $x_3$  est très influencée par le bruit de mesure qui donne une courbe éloignée de la vraie trajectoire de l'état  $x_3$ . Ceci est dû au fait que l'équation dynamique de l'état  $x_3$  est fonction des états  $x_1$  et  $x_4$  et l'état  $x_1$  est affecté par l'état  $x_2$  et la perturbation  $P_1$ .

Cependant il peut être distingué qu'à l'itération 25, l'estimation de  $x_3$  est complètement obtenue et que l'influence du bruit de mesure sur l'estimation de l'état  $x_3$  a été presque rejetée.

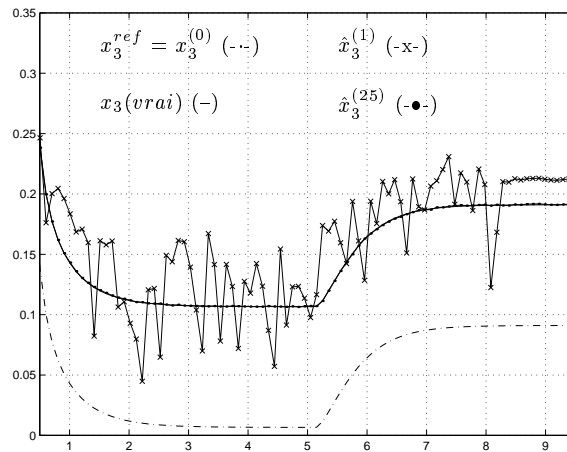


FIG. 3.30: Estimation de l'état  $x_3$  à différentes itérations quand les sorties  $y_1 = x_2$  et  $y_2 = x_4$  sont sous l'influence du bruit de mesure

La Figure 3.31 montre l'erreur entre l'estimation (obtenue à partir de la phase de traitement) et la vraie trajectoire de l'état  $x_3$  à différentes itérations. On peut voir la différence entre le niveau du bruit à l'itération 1, à l'itération 7 et à l'itération 25.

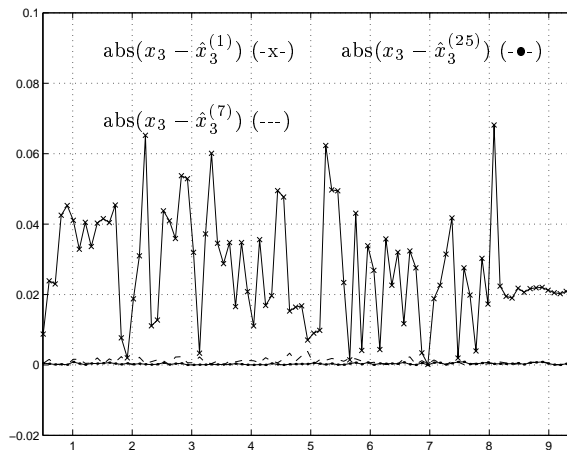


FIG. 3.31: L'erreur entre l'estimation et la trajectoire vraie de l'état  $x_3$  à différentes itérations quand les sorties  $y_1 = x_2$  et  $y_2 = x_4$  sont sous l'influence du bruit de mesure

Sur la Figure 3.32 est montrée l'estimation de la trajectoire de la perturbation  $P_1$  à différentes itérations. Comme dans les deux cas précédents, ces estimations sont calculées dans le **PE 1** où la perturbation  $P_1$  est une commande, dénoté  $\nu_2$  et la référence donnée à la perturbation est  $\nu^{ref} = 1$ .

Comme il a été mentionné auparavant, le but est que les estimations de  $P_1$  interagissent avec les estimations des états  $x_1$  et  $x_3$  à chaque itération afin de respecter toujours les sorties mesurées  $y_1 = x_2$  et  $y_2 = x_4$  même si maintenant ces sorties sont

bruitées.

Il peut être noté la grande influence du bruit de mesure sur l'estimation de  $P_1$  et qu'une estimation de  $P_1$  n'est pas complètement atteinte, cependant une trajectoire qui reflète de manière acceptable le comportement de  $P_1$  est obtenue.

On peut dire que avec les valeurs de  $R(1,2)$  calculées, l'algorithme arrive à éliminer l'influence du bruit des états  $x_1$  et  $x_3$  en punissant l'estimation de la perturbation  $P_1$ .

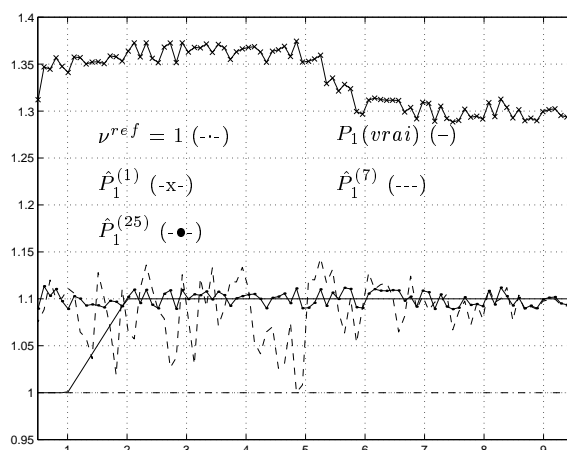


FIG. 3.32: Estimation de la perturbation  $P_1$  à différentes itérations quand les sorties  $y_1 = x_2$  et  $y_2 = x_4$  sont sous l'influence du bruit de mesure

Sur la Figure 3.33 est montrée l'erreur entre la perturbation réelle  $P_1$  et les trajectoires estimées  $\hat{P}_1$  à différentes itérations (voir l'influence du bruit).

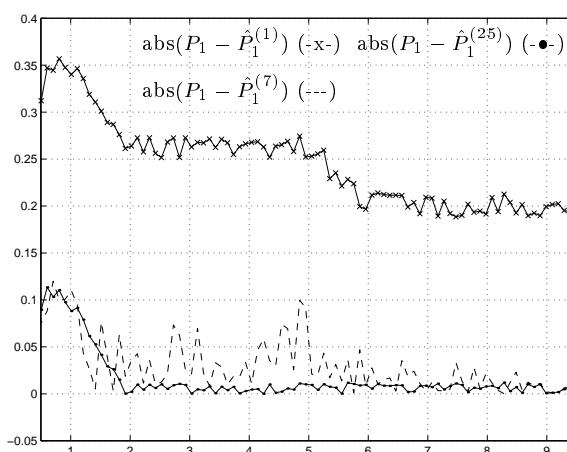


FIG. 3.33: L'erreur entre l'estimation et la trajectoire vraie de la perturbation  $P_1$  à différentes itérations quand les sorties  $y_1 = x_2$  et  $y_2 = x_4$  sont sous l'influence du bruit de mesure

Sur la Figure 3.34 (échelle normalisée -figure supérieure- échelle logarithmique -figure inférieure) l'évolution de la norme infinie de l'erreur entre la phase de traitement à l'itération  $k$  et la phase de traitement à l'itération  $(k - 1)$  est montrée et il peut être noté qu'aux itérations 14, 19 et 25, un point stationnaire est trouvé, ce qui est donné par

$$\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_{\infty} \leq \epsilon$$

Il peut être remarqué que l'erreur a augmenté à l'itération 15. Ceci est dû, comme dans la Simulation 2, au fait qu'avec la nouvelle valeur de la pondération  $R(1, 2)$  (obtenue à l'itération 14), les signaux obtenus de la phase de traitement ont changé par rapport aux signaux des itérations précédentes ce qui oblige à redéfinir les trajectoires de référence des problèmes élémentaires qui est exactement le but attendu de réaliser la modification de la valeur de la pondération de  $R(1, 2)$ . Cette même situation est répétée à l'itération 20.

Sur la Figure 3.35, l'évolution de la valeur de  $R(1, 2)$  par rapport au nombre d'itérations est présentée. La valeur de  $R(1, 2)$  a changé aux itérations 14 et 19, quand l'algorithme a atteint un (**PSI**), mais ces nouvelles valeurs (0.01 et 0.001) sont utilisées à partir des itérations 15 et 20 respectivement.

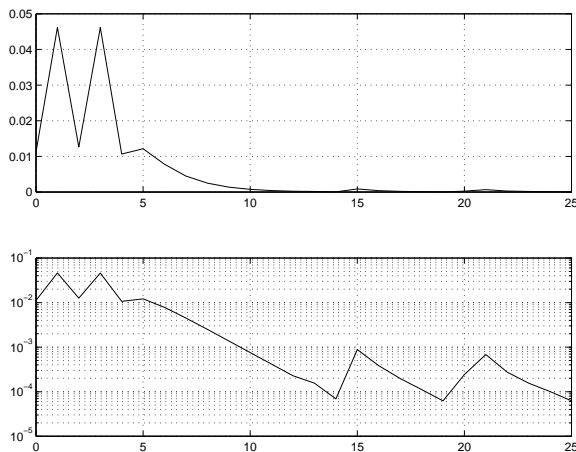


FIG. 3.34: L'évolution de  $\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_{\infty}$  par rapport aux itérations

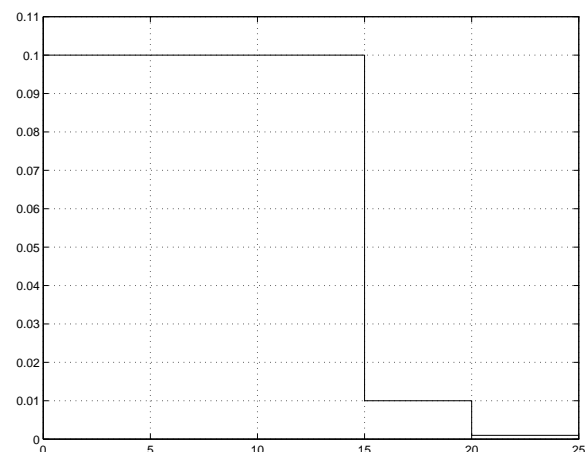


FIG. 3.35: L'évolution de la valeur de  $R(1, 2)$  par rapport aux itérations

#### d. Simulation 4

Pour cette simulation les signaux initiaux de référence ont changé par rapport aux simulations précédentes. Les nouvelles trajectoires de référence sont  $x_1^{ref} = 0.7$  et  $x_3^{ref} = 0.3$  pour l'état  $x_1$  et l'état  $x_3$  respectivement.

Sur la Figure 3.36 et la Figure 3.37 sont montrées l'estimation de la trajectoire de l'état  $x_2$  et de l'état  $x_4$  qui sont les sorties mesurées  $y_1$  et  $y_2$  respectivement. Dans cette simulation, comme dans les autres trois simulations faites auparavant, les trajectoires optimales de l'état  $x_2$  et  $x_4$  sont déjà obtenues à la première itération (après de la phase de traitement) donc elles ne sont plus présentées aux itérations suivantes.

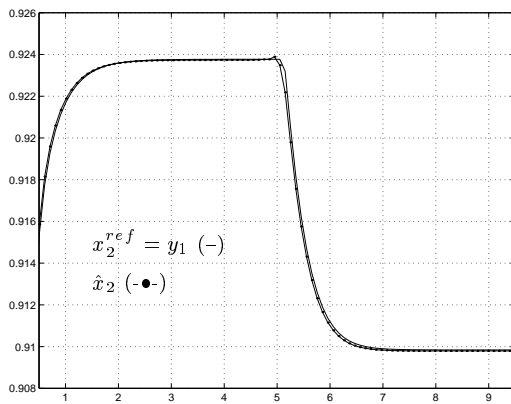


FIG. 3.36: Estimation de l'état  $x_2$

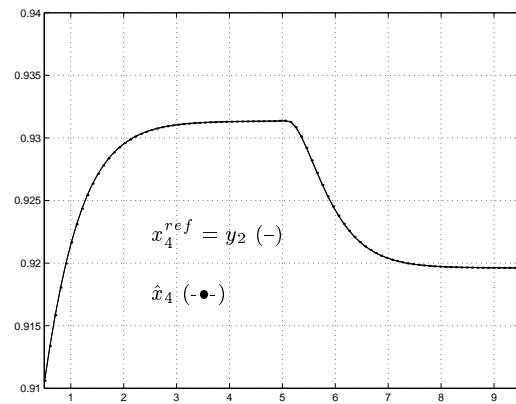


FIG. 3.37: Estimation de l'état  $x_4$

Sur la Figure 3.38 est montrée l'estimation de la trajectoire de l'état  $x_1$  à différentes itérations. Ces estimations sont calculées à partir de la phase de traitement, quand les valeurs des trajectoires de l'état  $x_1$  de chaque **PE** sont obtenues.

À l'itération 1, l'estimation de l'état  $x_1$  est encore loin de la vraie trajectoire de l'état  $x_1$ , mais la trajectoire de l'estimation se ressemble déjà à la vraie trajectoire malgré la référence  $x_1^{ref} = 0.7$  donnée et à l'influence de la perturbation sur le système. À partir de l'itération 2, le signal de référence est la trajectoire obtenue de la phase de traitement à l'itération  $(k - 1)$ .

Un point stationnaire est trouvé à l'itération 9, ceci signifie que la norme infinie de l'erreur entre la solution de la phase de traitement à l'itération 9 et la solution de la phase de traitement à l'itération 8 est plus petite que  $\epsilon$ , c'est à dire  $\|x^{(9)}(\cdot) - x^{(8)}(\cdot)\|_\infty \leq \epsilon$  ;  $\epsilon = 1 \times 10^{-4}$ . Dans ce point stationnaire les solutions de l'état  $x_1$  des différents problèmes élémentaires ne sont pas égales et sont aussi différentes de la solution obtenue de la phase de traitement, donc le point stationnaire trouvé est un **(PSI)**. Ceci implique que la valeur de la pondération  $R(1, 2)$  doit changer de 1 à 0.1 et par conséquent le calcul de l'estimation de  $x_1$  continue.

À l'itération 16, un nouveau **(PSI)** est rencontré, alors une nouvelle valeur de la

pondération  $R(1, 2)$  est calculée et la nouvelle valeur de  $R(1, 2)$  est  $R(1, 2) = 0.01$ . Voir la Figure 3.44 et la Figure 3.45 pour regarder l'évolution de  $\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_\infty$  et de la valeur de la pondération  $R(1, 2)$  respectivement.

L'estimation de l'état  $x_1$  est atteinte à l'itération 20 quand un **(PSC)** est trouvé. Les courbes de l'estimation et de la vraie trajectoire de l'état  $x_1$  sont complètement confondues.

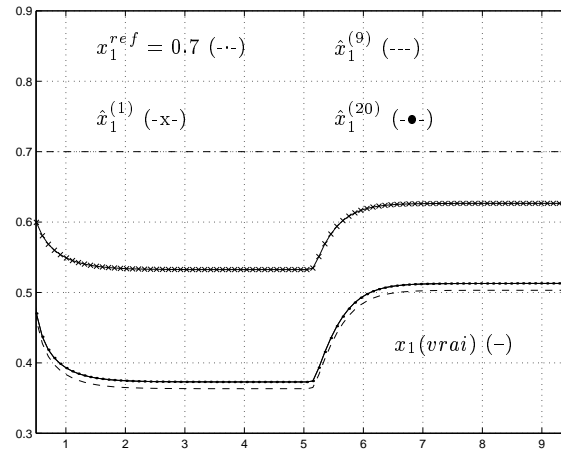


FIG. 3.38: Estimation de l'état  $x_1$  à différentes itérations

La Figure 3.39 montre l'erreur entre l'estimation (obtenue à partir de la phase de traitement) et la vraie trajectoire de l'état  $x_1$  à différentes itérations.

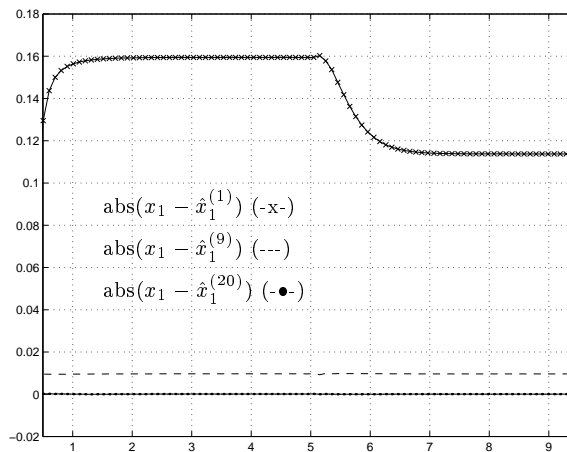


FIG. 3.39: L'erreur entre l'estimation et la trajectoire vraie de l'état  $x_1$  à différentes itérations

Sur la Figure 3.40 est montrée l'estimation de la trajectoire de l'état  $x_3$  à partir de la phase de traitement, quand les valeurs de l'état  $x_3$  de chaque **PE** à différentes itérations sont obtenues.

À l'itération 1, l'estimation de l'état  $x_3$  est encore loin de la vraie trajectoire de l'état  $x_3$ , mais la trajectoire de l'estimation se ressemble déjà à la vraie trajectoire malgré la référence  $x_3^{ref} = 0.3$  donnée. À partir de l'itération 2, le signal de référence est la trajectoire obtenue de la phase de traitement à l'itération  $(k - 1)$ . À l'itération 20 un (PSC) est retrouvé et donc l'estimation de  $x_3$  est atteinte. Les courbes de l'estimation et de la vraie trajectoire de l'état  $x_3$  sont complètement confondues.

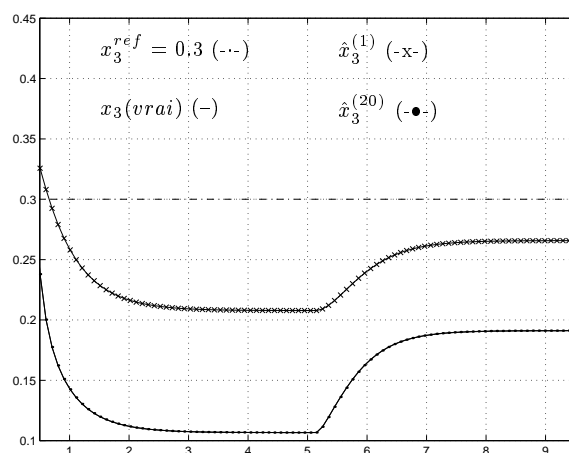


FIG. 3.40: Estimation de l'état  $x_3$  à différentes itérations

La Figure 3.41 montre l'erreur entre l'estimation (obtenue à partir de la phase de traitement) et la vraie trajectoire de l'état  $x_3$  à différentes itérations.

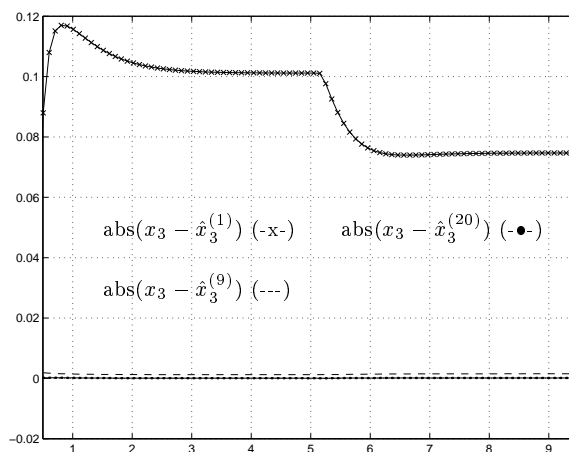


FIG. 3.41: L'erreur entre l'estimation et la trajectoire vraie de l'état  $x_3$  à différentes itérations

Sur la Figure 3.42 est montrée l'estimation de la trajectoire de la perturbation  $P_1$  à différentes itérations.



Comme dans le cas de la Simulation 2, il peut être noté qu'une estimation de  $P_1$  n'est pas complètement atteinte, cependant une trajectoire qui reflète globalement le comportement de  $P_1$  est obtenue. Une fois encore, l'algorithme a trouvé une proposition cohérente avec les signaux des états  $x_1$  et  $x_3$  et de la perturbation  $P_1$  qui explique le comportement des sorties  $y_1 = x_2$  et  $y_2 = x_4$ .

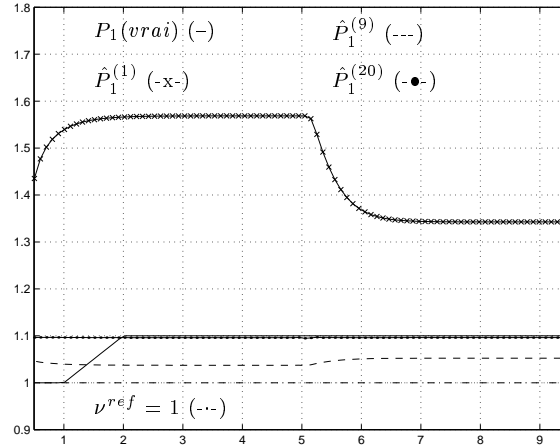


FIG. 3.42: Estimation de la perturbation  $P_1$  à différentes itérations

Sur la Figure 3.43 l'erreur entre la perturbation réelle  $P_1$  et les trajectoires estimées  $\hat{P}_1$  à différentes itérations est montrée.

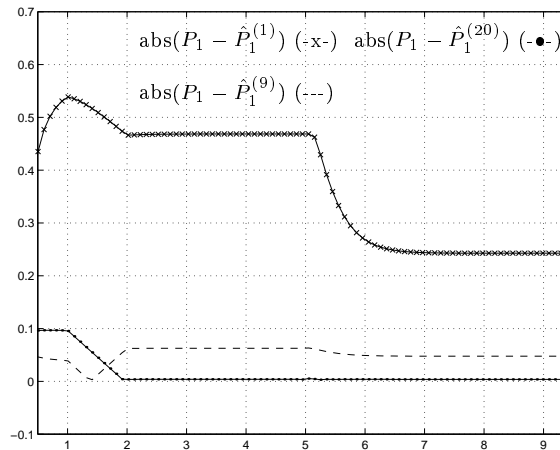


FIG. 3.43: L'erreur entre l'estimation et la trajectoire vraie de la perturbation  $P_1$  à différentes itérations

Sur la Figure 3.44 (échelle normalisée -figure supérieure- échelle logarithmique -figure inférieure-) l'évolution de la norme infinie de l'erreur entre la phase de traitement à l'itération  $k$  et la phase de traitement à l'itération  $(k - 1)$  est présentée et il peut être constaté qu'aux itérations 9, 16 et 20 est quand

$$\|x^{(\cdot)}(t) - x^{(\cdot)}(t)\|_{\infty} \leq \epsilon$$

ce qui représente un point stationnaire.

Il peut être noté que l'erreur a augmenté aux itérations 10 et 17. Ceci est dû au fait qu'avec la modification de la valeur de la pondération  $R(1, 2)$ , les signaux provenant de la phase de traitement ont changé ce qui amène à l'amélioration des signaux de référence afin de forcer la solution finale à être compatible avec tous les problèmes élémentaires.

Sur la Figure 3.45, l'évolution de la valeur de  $R(1, 2)$  par rapport au nombre d'itérations est présentée. La valeur de  $R(1, 2)$  a changé aux itérations 9 et 16, quand un (PSI) a été trouvé, mais ces nouvelles valeurs (0.1 et 0.01) sont utilisées à partir des itérations 10 et 17 respectivement.

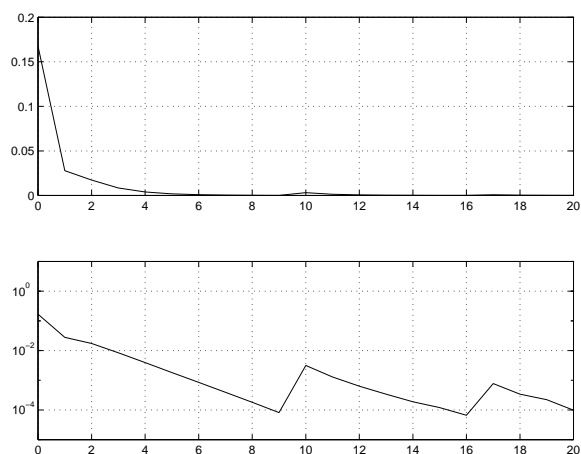


FIG. 3.44: *L'évolution de  $\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_\infty$  par rapport aux itérations*

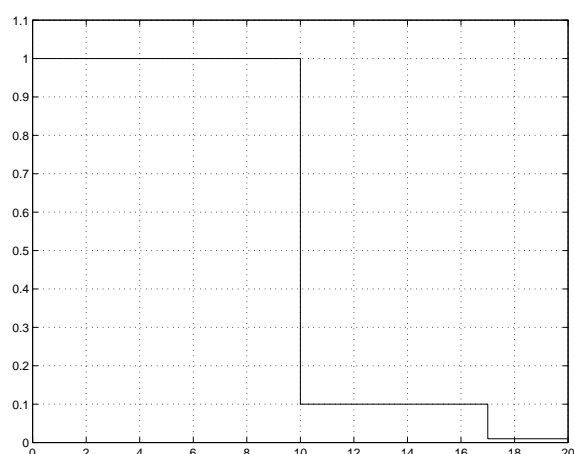


FIG. 3.45: *L'évolution de la valeur de  $R(1, 2)$  par rapport aux itérations*

## 3.7 Conclusion

Dans ce chapitre, un algorithme de synthèse des observateurs optimaux robustes a été présenté. Il a été montré que l'algorithme est indépendant des propriétés structurelles du système avec la capacité de tenir explicitement compte des perturbations et/ou des incertitudes de modèle ce qui donne le caractère robuste du schéma d'observation proposé. Cependant, l'objectif d'améliorer le temps de calcul reste une perspective de ce travail.

Il a été souligné que la méthodologie proposée passe par la définition d'un *Problème Élémentaire* (il se trouve autant de problèmes élémentaires que d'états) qui est un problème d'optimisation scalaire avec les caractéristiques suivantes

- Le système est temps-variant.
- L'état initial est inconnu.

La résolution du problème élémentaire est faite en deux étapes

- D'abord le problème élémentaire est transformé en une équation de Hamilton-Jacobi **scalaire**, qui est résolue en utilisant des codes de calcul existants et efficaces, afin d'obtenir les valeurs des solutions optimales correspondants à tous les états initiaux possibles.
- L'état initial est alors choisi pour être celui qui minimise la valeur de la solution optimale correspondante.

Il a été montré qu'une fois obtenues les solutions des différents problèmes élémentaires, la phase de traitement est réalisée afin de diriger l'algorithme vers la définition de problèmes élémentaires dont la solution est compatible avec le système dynamique dans sa totalité. Ceci est fait à travers des signaux de référence obtenus de la phase de traitement pour obliger la solution finale à être compatible avec toutes les équations du système.

Il a été montré aussi que dans beaucoup de cas et sous réserve de faire un choix particulier des vecteurs de commande dans les problèmes élémentaires, les équations de Hamilton-Jacobi peuvent prendre la forme particulière des équations de Riccati temps-variant. Cependant, la transformation d'un problème élémentaire en un problème linéaire nécessite parfois de modifier la définition des critères quadratiques.

Bien que les choix des valeurs des pondérations restent assez flexible, un ensemble de règles a été proposé visant à modifier dynamiquement les coefficients des pondérations dans chaque problème élémentaire afin de forcer l'algorithme itératif à converger vers un point stationnaire cohérent en quittant les pièges que sont les points stationnaires incohérents. Ces notions sont clairement définies à travers des informations disponibles que sont les signaux internes de l'algorithme.

En résumé, on peut citer pour la méthodologie de résolution proposée les caractéristiques suivantes

- La méthodologie décrite dans ce travail présente l'avantage d'être complètement indépendante de conditions structurelles souvent nécessaires pour la synthèse d'observateur analytique pour les systèmes non linéaires.
- Cette méthodologie permet de séparer le problème d'optimisation original dans des problèmes d'optimisation scalaires, ceci signifie que les systèmes de grande dimension peuvent être résolus avec une relative facilité.
- La mise en oeuvre parallèle est directe et peut accélérer la résolution du problème d'observation global. En effet, la seule étape de coordination est la phase de traitement des résultats des solutions élémentaires. Or cette phase ne comporte que des opérations algébriques très élémentaires (calcul des moyennes pondérées).
- De plus, même pour des systèmes incertains/perturbés pour lesquels il n'est pas facile de qualifier l'observabilité, l'algorithme proposé peut être employé comme un outil d'étude en permettant de montrer si la configuration de plusieurs signaux peut mener à la même sortie dans certaines circonstances.
- Cependant, il existe deux questions principales qui devront toujours être considérées dans des travaux postérieurs
  - ★ L'étude des propriétés de convergence. Il s'agit là d'une question très difficile à laquelle nous n'avons actuellement pas de réponse ni même un début de piste. Ceci est presque normal étant donné le niveau de généralité de la méthodologie.
  - ★ L'accélération du temps de calcul. Cette question reste toutefois relative étant donnée la disparité des temps caractéristiques des systèmes physiques concernés.



Résultats Supplémentaires sur les  
Observateurs Non Linéaires à Horizon  
Glissant



## Chapitre 4

# Résultats Supplémentaires sur les Observateurs Non Linéaires à Horizon Glissant

Dans le chapitre 2, une étude des différentes techniques existantes pour la synthèse d'observateurs pour les systèmes non linéaires a été présentée et dans le chapitre 3 une méthodologie pour construire des observateurs optimaux robustes à horizon glissant basée sur l'idée de la décomposition en problèmes élémentaires a été proposée.

Dans ce chapitre, l'intérêt est concentré sur les observateurs à horizon glissant plus classiques [4; 95; 115; 116; 152]. Ce sont des observateurs qui emploient l'intégrale du carré de l'erreur de prédiction de la sortie dans le processus d'estimation de l'état.

L'avantage principal dans l'utilisation de tels observateurs est l'absence de toute hypothèse sur l'existence de formes canoniques pour que l'observateur correspondant soit conçu. Cependant, les calculs associés sont généralement très lourds et peuvent empêcher l'utilisation de ces observateurs pour des systèmes avec une dynamique "rapide".

Dans [4], des solutions techniques ont été proposées pour partiellement surmonter ces inconvénients à l'aide de la technique de post-stabilisation. Il a été prouvé que, de cette manière, le rapport (précision/coût de calcul) peut être sensiblement amélioré. Il a été aussi proposé l'utilisation d'un schéma de régularisation afin d'éviter les singularités au prix d'un observateur qui ne converge que vers un voisinage de l'état. La dimension de ce voisinage dépend du "*degré d'observabilité*" du système.

Dans [4] cependant, une hypothèse de régularité uniforme globale a été employée en même temps qu'une borne sur les trajectoires donnée a priori afin d'obtenir les résultats de convergence. Une hypothèse de régularité globale est plus ou moins implicitement employée dans tous les travaux concernant les systèmes non linéaires généraux [8; 9; 102; 103].



Le but de ce chapitre est de traiter le cas où de telles hypothèses ne sont pas assumées. Les outils utilisés dans ce chapitre sont ceux typiquement employés dans les travaux concernant la stabilisation semi-global par retour de sortie [47; 130; 131].

Le résultat du présent chapitre peut être récapitulé comme suit :

Étant donné un ensemble compact  $S_0$  d'états initiaux possibles, il y a un observateur à grand gain à horizon glissant qui converge à condition que l'erreur initial d'estimation soit inférieure au rayon d'observabilité associé à cet ensemble compact. La définition de ce rayon fait partie des contributions de ce chapitre.

Il a été aussi prouvé que sous l'hypothèse de régularité uniforme globale utilisée dans [4], le rayon d'observabilité tend vers l'infini quand le rayon de  $S_0$  tend vers l'infini. Ceci permet au résultat de [4] d'être récupéré comme un cas particulier de la formulation proposée dans ce chapitre. Les résultats de ce chapitre sont essentiellement de nature théorique. Ils permettent une meilleure compréhension de ce que coûterait l'absence de l'hypothèse de régularité globale utilisée dans [4]. Sur le plan de la synthèse elle-même, les résultats de ce chapitre ne sont pas vraiment nouveaux.

Il est important de noter que des idées semblables ont été discutées dans [113] pour des systèmes semi-linéaires (systèmes linéaires avec le terme non-linéaire satisfaisant une condition de Lipschitz) basées sur des résultats généraux sur des systèmes du type Lipschitz proposés dans [112]. Ces résultats ont été rectifiés dans [1].

Dans la section suivante, les définitions et les notations utilisées dans l'ensemble de ce chapitre sont présentées.

## 4.1 Définitions et Notations

Dans cette section la classe des systèmes étudiés ainsi que les définitions associées au système sont d'abord présentées. Puis les définitions associées à la notion d'observabilité sont précisées. Enfin, les définitions associées à la synthèse de l'observateur sont données.

### 4.1.1 Les définitions associées au système

Les systèmes non linéaires considérés sont régis par les équations suivantes

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= h(x) \end{aligned} \tag{4.1.1}$$

où  $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^p$  et  $u \in U \subset \mathbb{R}^m$  représentent respectivement l'état, la sortie mesurée et la commande. Les fonctions  $f$  et  $h$  sont supposées deux fois continûment différentiables.

La commande est supposée appartenir à un ensemble compact  $U \subset \mathbb{R}^m$ . Cette saturation de la commande doit être compris dans le contexte de la stabilisation semi-globale où des saturations sont explicitement imposées à la commande afin d'éviter des temps d'évasion finis non contrôlés (plus courts que le temps de convergence de l'estimation) quand une commande par retour d'état est combinée avec un observateur non linéaire [47; 130; 131].

Il est aussi supposé qu'une commande constante par morceaux est utilisée avec une certaine période d'échantillonnage fixe  $s > 0$  afin que les profils admissibles de la commande  $\tilde{u} \in U^{[t_0, t_0 + Ns]}$  appartiennent au sous-ensemble compact de dimension fini  $U^N$  de  $\mathbb{R}^{Nm}$ . Sur la base de cette hypothèse, le sous-ensemble  $U^N$  et l'ensemble  $\mathbb{R}^{[t-Ns, t]}$  sont utilisés, selon le contexte, pour désigner un profil de commande sur l'intervalle  $\mathbb{R}^{[t-Ns, t]}$ .

Les notations suivantes sont utilisées dans la suite :

- $X(t; t_0; x_0; \tilde{u})$  et  $Y(t; t_0; x_0; \tilde{u}) := h(X(t; t_0; x_0; \tilde{u}))$  représentent respectivement les trajectoires de l'état et de la sortie avec les conditions initiales  $(t_0, x_0)$  sous le profil de commande  $\tilde{u} \in U^{[t_0, t]}$ .
- Étant donné une fonction  $\varphi : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ , la notation  $\varphi \in \mathcal{K}$  désigne les propriétés suivantes :
  - ★  $\varphi(\cdot)$  est continue et strictement croissante.
  - ★  $\varphi(0) = 0$ .
- Étant donné une fonction  $\beta : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ , la notation  $\beta \in \mathcal{KL}$  désigne que  $\beta \in \mathcal{K}$  par rapport au premier argument et qu'elle est décroissante vers 0 par rapport au second argument.
- Pour toute fonction vectorielle  $g(\cdot)$  définie sur l'intervalle de temps  $I$ , la notation suivante est utilisée :

$$\|g(\cdot)\|_{L^i}^I := \int_I \|g(\tau)\|^i d\tau \quad (4.1.2)$$

$$\|g(\cdot)\|_{\infty}^I := \sup_{t \in I} \|g(\tau)\| \quad (4.1.3)$$

- Pour toute matrice  $A$ ,  $\underline{\sigma}(A)$  dénote la plus petite valeur singulière de  $A$ .
- Pour un ensemble borné  $E$ ,
  - ★  $\rho(E) := \min_{x \in E} \|x\|$  dénote le rayon de l'ensemble borné  $E$  avec la convention  $\rho(\emptyset) = \infty$ .
  - ★  $d_2(E) := \{(x, x) \mid x \in E\}$  est la diagonale de  $E \times E$ .
  - ★ Si  $E$  est un sous-ensemble d'un espace Euclidien et  $z \notin E$ , alors  $d(z, E)$  dénote la distance de  $z$  à  $E$  avec la convention  $d(z, \emptyset) = \infty$ .

### 4.1.2 Définitions associées à l'observabilité

Dans [4], des définitions simples pour l'observabilité et la détectabilité ont été proposées et comparées à celles existantes [62; 66; 128]. Pour simplicité, seuls les systèmes complètement uniformément observables sont considérés pour présenter la notion théorique développée dans ce chapitre afin de découpler les difficultés. Cependant pour mieux comprendre la synthèse de l'observateur et les notions qui seront développées, les définitions données dans [4] sont présentées

**Définition 4.1.1. [Systèmes Détectables][4]**

Le système (4.1.1) est dit détectable s'il y a une partition

$$x = \begin{pmatrix} x_I \\ x_{II} \end{pmatrix}$$

de l'espace d'état telle que pour tout  $T > 0$ , les conditions suivantes sont satisfaites :

1. La norme  $L_2$  de l'erreur de prédiction de la sortie donne une bonne indication sur l'erreur d'estimation de  $x_I$  dans le sens où il existe une fonction  $\varphi_1 \in \mathcal{K}$  telle que pour tout  $x^{(1)}, x^{(2)} \in \mathbb{R}^n$  et tout  $\tilde{u} \in U^N$  admissible, l'on ait

$$\|x_I^{(1)} - x_I^{(2)}\| \leq \varphi_1 \left( \|Y(\cdot; t_0; x^{(1)}; \tilde{u}) - Y(\cdot; t_0; x^{(2)}; \tilde{u})\|_{L_2}^{[t_0, t_0+T]} \right) \quad (4.1.4)$$

2. Le sous-état  $x_{II}$  n'a aucun effet sur la sortie, à savoir :

$$\forall \tau \quad \forall x \quad \frac{\partial Y}{\partial x_{II}} \left( \tau; t_0; x; \tilde{u} \right) \cdot f_{II}(x, \tilde{u}) = 0 \quad (4.1.5)$$

3. Pour toutes les évolutions d'état  $x^{(1)}(t)$  et  $x^{(2)}(t)$  satisfaisant :

$$\|x_I^{(1)}(t) - x_I^{(2)}(t)\| \rightarrow 0$$

on a

$$\|X_{II}(t; t_0; x^{(1)}; \tilde{u}) - X_{II}(t; t_0; x^{(2)}; \tilde{u})\| \rightarrow 0 \quad (4.1.6)$$



La condition (4.1.4) signifie que deux états sont toujours distinguables en ce qui concerne leur première partie  $x_I$ , et ceci en comparant leur sorties correspondantes pendant un intervalle de temps de longueur  $T$ , tandis que la condition (4.1.6) assure que la deuxième partie  $x_{II}$  peut être asymptotiquement approchée toutes les fois qu'une bonne évaluation de la première partie est obtenue.

**Remarque 4.1.1.** Il est à noter que la définition ci-dessus de la détectabilité et la décomposition relative de l'espace d'état est une généralisation directe du cas de systèmes détectables linéaires. □

Il est important de noter que seule l'existence de cette décomposition est supposée, la connaissance des détails de cette décomposition et les changements de coordonnées qui y conduisent ne sont pas nécessaires à la synthèse elle-même de l'observateur. En d'autres mots, il s'agit d'une condition permettant la preuve de la convergence qui n'est cependant pas indispensable pour la construction du système dynamique qu'est l'observateur à horizon-glissant. Quant à l'observabilité, la définition suivante est utilisée

**Définition 4.1.2. [Systèmes Complètement Uniformément Observables][4]**

Le système (4.1.1) est dit complètement uniformément observable s'il est détectable au sens de la Définition 4.1.1 avec  $x_I = x$ . Plus précisément

$$\|x^{(1)} - x^{(2)}\| \leq \varphi_1 \left( \|Y(\cdot; t_0; x^{(1)}; \tilde{u}) - Y(\cdot; t_0; x^{(2)}; \tilde{u})\|_{L_2}^{[t_0, t_0+T]} \right) \quad (4.1.7)$$



La Définition 4.1.2 liée à la Définition 4.1.1 déclare simplement que la norme  $L_2$  de l'erreur de prédiction de la sortie reflète l'erreur d'estimation de l'ensemble de l'état et finalement quand l'erreur de prédiction de la sortie tend vers 0, l'erreur d'estimation de l'état tend aussi vers 0. Ceci est directement lié à la notion des états distinguables qui était à l'origine des définitions intuitives originales de l'observabilité.

Dans la section suivante, les définitions associées à la synthèse de l'observateur sont présentées.

**4.1.3 Les définitions associées à la conception de l'observateur**

Soit un certain horizon fixe de longueur  $T = Ns > 0$ . Dans ce qui suit  $\tilde{u}_t$  dénote un élément admissible dans  $U^{[t-T, t]}$  et  $\tilde{u}_t(\tau)$  sa valeur à l'instant  $\tau \in [t - T, t]$ . Supposons que pendant l'évolution du système, à chaque instant  $t$ , on dispose des mesures passées de la sortie et de la commande sur  $[t - T, t]$ . En utilisant les notations :

$$x^-(t) := x(t - T) \quad ; \quad u^-(t) := (t - T)$$

l'application suivante peut être définie :

$$J : \mathbb{R}^n \times \mathbb{R}^n \times U^N \rightarrow \mathbb{R}^+ \\ (z, x^-, \tilde{u}) \rightsquigarrow J(z, x^-, \tilde{u}) := \|Y(\cdot; 0; z; \tilde{u}) - Y(\cdot; 0; x^-; \tilde{u})\|_{L_2}^{[0, T]} \quad (4.1.8)$$

Il convient de noter qu'afin d'évaluer  $J$ , la connaissance complète des états  $x^-$  n'est pas nécessaire. Seulement les sorties correspondantes sont nécessaires.

Par la Définition 4.1.2 de l'observabilité, il est facile de constater que  $J$  est une quantité principale dans la conception de l'observateur. En effet, cette quantité représente l'erreur de prédiction de la sortie associée à l'estimation  $z(t)$  de  $x^-(t) = x(t - T)$ . Car, selon (4.1.7), pour toute commande  $\tilde{u}$  dans  $U^N$ , le fait que  $J(z, x^-, \tilde{u}) = 0$

impliquerait l'égalité  $z = x^-$ . Ceci signifierait que  $x = X(T; 0; z; \tilde{u})$ .

Partant, la conception de l'observateur peut être basée sur l'idée suivante : "*Définir  $z(t)$  comme l'état interne d'un observateur. Doter  $z$  d'une dynamique qui force  $J(z, x^-, \tilde{u})$  à décroître et prendre l'estimation actuelle de  $x(t)$ , à savoir  $\hat{x}(t) = X(T; 0; z(t); \tilde{u})$  comme la sortie de l'observateur*".

La manière dont la dynamique de  $z$  est définie afin de conférer un comportement décroissant à  $J$  est basée sur une stratégie simple de descente de gradient. En effet, les fonctions  $f$  et  $h$  étant supposées continuellement différentiables, les fonctions  $X(\tau; 0; z; \tilde{u})$  et par conséquent  $J(z, x^-, \tilde{u})$  sont continuellement différentiables par rapport à  $z$ . Ainsi, le gradient de  $J$  par rapport à  $z$

$$G(z(t), x^-(t), \tilde{u}_t) := \frac{\partial J}{\partial z}(z(t), x^-(t), \tilde{u}_t) \quad (4.1.9)$$

est une fonction bien définie, continue. Cette fonction est d'une importance capitale dans ce qui suit.

Dans [4],  $G$  est supposé être globalement uniformément de rang 1. Cette hypothèse ainsi que d'autres hypothèses de nature plutôt technique sur les bornes de certaines fonctions ont permis d'obtenir un résultat de convergence globale. Ici, un point de vue différent est considéré pour mieux comprendre ce qui arrive en absence de telles hypothèses globales.

Tout d'abord, ces hypothèses seront rappelées dans cette partie, afin de montrer les conditions restrictives liées à de telles hypothèses. Le but de ce rappel est de présenter les résultats obtenus dans [4] qui représentent la théorie de base des nouveaux concepts développés dans ce chapitre.

En plus de l'hypothèse que le système est détectable au le sens de la Définition 4.1.1, une condition de régularité sur la matrice Jacobienne  $G(z(t), x^-(t), \tilde{u}_t)$  définie par (4.1.9) est nécessaire.

**Hypothèse 4.1.1. [Régularité Uniforme de  $G$ ][4]**

Il y a une fonction  $\sigma \in \mathcal{K}$  telle que

$$\underline{\sigma}(G(z(t), x^-(t), \tilde{u}_t)) \geq \sigma\left(\|z_I(t) - x_I^-(t)\|\right) \quad (4.1.10)$$

où  $z_I$  et  $x_I^-$  sont les sous-états définis par la partition associée à l'hypothèse de détectabilité de la Définition 4.1.1. ♠

En d'autres mots, la matrice Jacobienne  $G(z(t), x^-(t), \tilde{u}_t)$  est de rang 1 uniformément loin de la diagonale  $z_I(t) = x_I^-(t)$ . La grandeur  $\underline{\sigma}$  est employée pour mesurer la proximité à la singularité. En d'autres termes, quand  $x(t)$  se déplace, pour tout  $t \geq 0$ , la fonction

coût  $J(\cdot, x^-(t), \tilde{u}_t)$  possède un minimum global. Cela est clairement une condition tout à fait restrictive.

Afin de cerner clairement le problème étudié dans ce chapitre, il convient de rappeler quelques résultats de [4]. Un premier résultat technique a été donné dans [4]. Dans ce résultat, un gain d'innovation variable est utilisé sur la base de la connaissance de bornes supérieures de certaines expressions. Nous contentons de rappeler un corollaire du résultat principal de [4] qui se présente sous forme d'un observateur asymptotique à grand gain.

**Corollaire 4.1.2.** [*Un observateur à grand gain*][4]

*Si*

1. Le système (4.1.1) est détectable au le sens de la Définition 4.1.1.
2. L'Hypothèse 4.1.1 est satisfaite.
3. Il y a un domaine borné  $\mathcal{D}$  contenant toutes les évolutions possibles de  $z$  et de  $x$

*alors il y a un gain  $\gamma$  suffisamment grand pour que le système dynamique défini par*

$$\dot{z} = f(z) - \gamma G^T [GG^T]^{-1} \sqrt{J} \quad (4.1.11)$$

$$\hat{x} = X(T; t_0; z(t), \tilde{u}_t) \quad (4.1.12)$$

*où  $J(t) = J(z(t), x^-(t), \tilde{u}_t)$  et  $G(t) = G(z(t), x^-(t), \tilde{u}_t)$  soit un observateur asymptotique pour (4.1.1). ♠*

**Remarque 4.1.2.** Le Corollaire 4.1.2 donne un observateur à grand gain sans utiliser les conditions structurelles basées sur les dérivées successives de la sortie mesurée:

$$\begin{pmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{pmatrix} \quad (4.1.13)$$

qui sont généralement employées dans le contexte des observateurs à grand gain [54].

□

Il est important de noter que l'exigence technique liées à la synthèse des observateurs à grand gain classiques a été remplacée par la combinaison de deux conditions, à savoir

- L'Hypothèse 4.1.1 sur la régularité du gradient  $G$  qui permet la définition du gain de l'observateur, et

- L'hypothèse de détectabilité qui permet d'établir une équivalence entre la convergence de l'erreur d'estimation et celui de l'erreur de prédiction de la sortie.

Le problème avec les équations de l'observateur comme elles ont été proposées par le Corollaire 4.1.2 est que ces équations peuvent mener à l'instabilité numérique en s'approchant de la solution  $J = 0$ . En effet, il est clair qu'à  $J = 0$  correspond  $G = 0$  et donc, les inévitables erreurs de troncature peuvent mener à des situations où  $J \approx 0$  (mais  $J \neq 0$ ) tandis que  $G = 0$ .

Le corollaire suivant propose une légère modification du Corollaire 4.1.2 qui régularise les équations de l'observateur autour de la solution au prix de l'obtention d'une borne arbitrairement petite qui assure de l'erreur d'estimation.

Les résultats sont donnés dans le cas des systèmes complètement observables au sens de la Définition 4.1.2. Le cas des systèmes détectables peut être obtenu de la même façon en utilisant certaines hypothèses techniques complémentaires qui sont évidentes.

**Corollaire 4.1.3.** [*Un observateur à grand gain approximatif avec dynamique régularisée*][4]

Une borne admissible  $\eta > 0$  est choisie sur l'erreur d'estimation.

Si

1. Le système (4.1.1) est complètement observable au sens de la Définition 4.1.2.
2. L'Hypothèse 4.1.1 est satisfaite.
3. Il y a un domaine borné  $\mathcal{D}$  contenant toutes les évolutions possibles de  $z$  et de  $x$

alors il y a un gain  $\gamma$  suffisamment grand et un  $\epsilon > 0$  suffisamment petit pour que le système dynamique défini par

$$\dot{z} = f(z) - \gamma G^T [GG^T + \epsilon]^{-1} \sqrt{J} \quad (4.1.14)$$

$$\hat{x} = X(T; t_0; z(t), \tilde{u}_t) \quad (4.1.15)$$

où  $J(t) = J(z(t), x^-(t), \tilde{u}_t)$  et  $G(t) = G(z(t), x^-(t), \tilde{u}_t)$  soit un observateur approximatif pour (4.1.1) dans le sens suivant

1. Le système (4.1.14)-(4.1.15) atteint l'ensemble

$$E_\eta := \{(\hat{x}, x) \in \mathbb{R}^{2n} \mid \|\hat{x} - x\| < \eta\} \quad (4.1.16)$$

après un temps fini.

2. Le système (4.1.14)-(4.1.15) rend (4.1.16) invariant par rapport à la dynamique du système/observateur.



L'ensemble des résultats de [4] sont essentiellement fondés sur l'hypothèse restrictive de la régularité du gradient du critère. Le travail proposé dans ce chapitre ne change pas la synthèse de l'observateur. Il ne fait que préciser ce qui se passe si cette hypothèse de régularité globale n'était pas satisfaite.

Ainsi, les hypothèses du résultat de ce chapitre sont modifiées tout en gardant les mêmes équations pour l'observateur. Le résultat obtenu est alors un résultat de convergence semi-global de l'observateur. En d'autres mots, la sortie de l'observateur converge vers l'état du système si l'erreur d'estimation initial est suffisamment petite. Ce "*suffisamment petit*" se traduit par le fait que l'état estimé initialement se trouve dans la boule centrée sur l'état réel et de rayon égal à ce qui va être défini comme le rayon de convergence.

## 4.2 Le rayon d'observabilité

Rigoureusement parlant, le concept du rayon d'observabilité développé dans cette section est partiellement lié à l'utilisation du **G**radient dans le schéma d'observation à **H**orizon **G**lissant (**GHG**). Il conviendrait alors de parler du rayon d'observabilité-**GHG**. On rappelle que seulement les systèmes complètement uniformément observables sont considérés pour présenter la notion théorique développée dans ce chapitre. D'abord, le lemme suivant est utilisé pour établir la régularité locale du gradient  $G$  :

**Lemme 4.2.1.** [*Régularité Uniforme Locale de G*][4]

*Si pour tout  $(t, x) \in \mathbb{R} \times \mathbb{R}^n$  et tout  $\tilde{u}_t \in U^{[t, \infty]}$  admissible, le système obtenu à partir de (4.1.1) par la linéarisation autour de la trajectoire qui passe par  $(t, x)$ , à savoir :*

$$\dot{\xi}(\tau) = \frac{\partial f}{\partial x} \left( X(\tau; t; x; \tilde{u}_t), \tilde{u}_t(\tau) \right) \xi(\tau) \quad (4.2.1)$$

$$\nu(\tau) = \frac{\partial h}{\partial x} \left( X(\tau; t; x; \tilde{u}_t) \right) \xi(\tau) \quad (4.2.2)$$

*est uniformément complètement observable, alors pour tout ensemble compact  $\mathcal{X} \subset \mathbb{R}^n$  il existe une fonction continue strictement positive  $\alpha(\cdot)$  telle que*

$$\underline{\sigma} \left( \frac{\partial^2 J}{\partial z^2} (x^-, x^-, \tilde{u}_t) \right) \geq \alpha(x^-) > 0 \quad (4.2.3)$$

*pour tout  $(t, x) \in \mathbb{R} \times \mathcal{X}$*



En fait, ceci résulte du fait que la matrice Hessienne n'est rien d'autre que le grammien d'observabilité du système linéaire temps-variant correspondant. L'implication pratique du Lemme 4.2.1 est la suivante :



"Étant donnée une paire  $(t, x)$  fixée et une commande  $\tilde{u}_t \in U^N$  admissible, tout point singulier  $z$  de  $J(\cdot, x^-, \tilde{u})$  est nécessairement suffisamment loin de  $x^-$ ".

Plus rigoureusement, le résultat suivant est obtenu :

**Corollaire 4.2.2.** *Conformément aux hypothèse du Lemme 4.2.1, pour tout  $\mathcal{X} \subset \mathbb{R}^n$  compact et tout  $\tilde{u}_t \in U^N$  l'inégalité suivante est satisfaite : Voir la Figure 4.1*

$$d\left(0, \left\{ r > 0 \quad \exists (x^-, z) \in \mathcal{X} \times \bar{B}(x^-, r) \setminus \{x^-\}, \quad t. q. \quad G(z, x^-, \tilde{u}) = 0 \right\}\right) > 0 \quad (4.2.4)$$

cette distance, dénotée par la suite par  $R_r(\mathcal{X})$  est mentionnée comme le rayon de régularité-**GHG** par rapport à l'ensemble compact  $\mathcal{X}$ . ♠

Le signification de l'équation (4.2.4) est le suivant :

Il existe un nombre strictement positif tel que, autour de tout  $x^- \in \mathcal{X}$  la boule centrée sur  $x^-$  et ayant ce rayon ne contient aucun point singulier  $z \neq x^-$  (tel que  $G(z, x^-, \tilde{u}) = 0$ ). Noter que ce rayon est le même pour tout  $x^- \in \mathcal{X}$ .

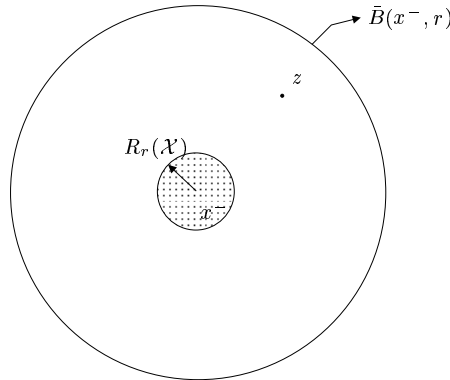


FIG. 4.1: Représentation géométrique de l'équation (4.2.4)

Autrement dit,  $R_r(\mathcal{X})$  est un minorant de la distance entre  $x^-$  et un autre état singulier  $z$  pour  $J(\cdot, x^-, \tilde{u})$ .

Pour la preuve du Corollaire 4.2.2 voir l'Annexe C.

**Remarque 4.2.1.** Il convient de noter que quand la régularité uniforme globale de  $G(\cdot, x^-, \tilde{u})$  est assumée comme c'est le cas dans [4],  $R_r(\mathcal{X})$  défini par (4.2.4) est la distance de 0 à un ensemble vide et par conséquent  $R_r(\mathcal{X}) = \infty$  □

**Remarque 4.2.2.** La remarque 4.2.1 permet de renforcer le fait que le rayon de régularité-**GHG** ne représente pas un rayon autour de  $x^-$  au-delà duquel la linéarisation

devient grossière. En d'autres termes,  $z$  peut demeurer dans le rayon de régularité-**GHG** autour de  $x^-$  bien au-delà du domaine de validité de la linéarisation. Ce rayon reflète vraiment à quelle "distance" se trouve le système par rapport à l'hypothèse de régularité uniforme globale. (Voir l'exemple explicatif de la Section 4.4 pour une illustration quantitative de ce propos).  $\square$

Le rayon de régularité-**GHG** donne la norme admissible de l'erreur d'estimation avant que des singularités locales associées à une approche de descente du type gradient deviennent possibles à rencontrer.

Les deux ensembles suivants, reliés à la définition de  $R_r(\mathcal{X})$  sont nécessaires pour la suite de l'étude (Voir la Figure 4.2) :

$$\mathcal{I}_r^{ad}(\mathcal{X}) = \left\{ (z, x^-) \in \mathbb{R}^n \times \mathcal{X} \mid \|z - x^-\| \leq R_r(\mathcal{X}) \right\} \quad (4.2.5)$$

$$\partial\mathcal{I}_r^{ad}(\mathcal{X}) = \left\{ (z, x^-) \in \mathbb{R}^n \times \mathcal{X} \mid \|z - x^-\| = R_r(\mathcal{X}) \right\} \quad (4.2.6)$$

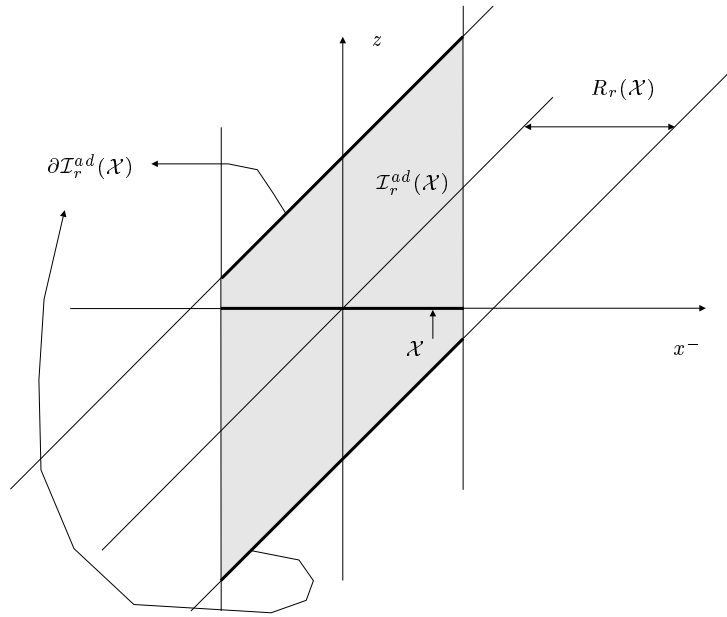


FIG. 4.2: Définition des ensembles  $\mathcal{I}_r^{ad}(\mathcal{X})$  et  $\partial\mathcal{I}_r^{ad}(\mathcal{X})$

Selon la définition du rayon de régularité  $R_r(\mathcal{X})$ , il est clair qu'il n'y a aucune paire  $(z, x^-)$  qui est dans  $\mathcal{I}_r^{ad}(\mathcal{X})$  tel que  $z \neq x^-$  et  $G(z, x^-, \tilde{u}) = 0$ .

En relation avec le sous-ensemble  $\partial\mathcal{I}_r^{ad}(\mathcal{X})$  défini ci-dessus, soit la quantité suivante :

$$\bar{J}_r(\mathcal{X}) = \inf_{(z, x^-, \bar{u}) \in \partial\mathcal{I}_r^{ad}(\mathcal{X}) \times U^N} J(z, x^-, \bar{u}) \quad (4.2.7)$$

Il est facile de noter que  $\bar{J}_r(\mathcal{X}) > 0$ , cela vient aisément de la compacité de l'ensemble  $\partial\mathcal{I}_r^{ad}(\mathcal{X}) \times U^N$  sur lequel la minimisation est effectuée et de la "distinguable" uniforme de toute paire  $(z, x^-)$  appartenant à  $\partial\mathcal{I}_r^{ad}(\mathcal{X})$ .

**Remarque 4.2.3.** Notons encore une fois que si la régularité globale uniforme de  $G$  est supposée, une paire  $(z, x^-)$  dans  $\partial\mathcal{I}_r^{ad}(\mathcal{X})$  satisfait la propriété suivante :

$$\lim_{\rho(\mathcal{X}) \rightarrow \infty} \left[ \inf_{(z, x^-) \in \partial\mathcal{I}_r^{ad}(\mathcal{X})} \|z - x^-\| \right] = \lim_{\rho(\mathcal{X}) \rightarrow \infty} \left[ R_r(\mathcal{X}) \right] = \infty \quad (4.2.8)$$

(Voir le Remarque 4.2.1) ce qui, avec (4.1.7) donne la propriété suivante :

$$\lim_{\rho(\mathcal{X}) \rightarrow \infty} \left[ \bar{J}_r(\mathcal{X}) \right] = \infty \quad (4.2.9)$$

□

La dernière définition qui est nécessaire pour définir le rayon d'observabilité-**GHG** est la suivante :

$$J_U^{-1}(\mathcal{X}) = \left\{ (z, x^-) \in \mathcal{I}_r^{ad}(\mathcal{X}) \text{ tel que } \sup_{\tilde{u} \in U^N} J(z, x^-, \tilde{u}) \leq \bar{J}_r(\mathcal{X}) \right\} \quad (4.2.10)$$

Ce sous-ensemble de  $\mathcal{I}_r^{ad}(\mathcal{X})$  est clairement non vide puisqu'il contient  $d_2(\mathcal{X})$ . En outre, puisque  $J(\cdot, \cdot, \tilde{u}) = 0$  sur  $d_2(\mathcal{X})$  pour tout  $\tilde{u}$ ,  $J_U^{-1}(\mathcal{X})$  contient un ensemble ouvert contenant  $d_2(\mathcal{X})$  (voir la Figure 4.3). Avec les définitions ci-dessus, le rayon d'observabilité-**GHG** peut être défini

**Définition 4.2.1.** *Sous les hypothèses du Lemme 4.2.1, le rayon d'observabilité-**GHG** du système (4.1.1) par rapport à l'ensemble compact  $\mathcal{X}$  est donné par*

$$R_O(\mathcal{X}) = \sup \left\{ r < R_r(\mathcal{X}) \text{ tel que } B(x^-, r) \subset J_U^{-1}(\mathcal{X}) \quad \forall x^- \in \mathcal{X} \right\} \quad (4.2.11)$$

où  $J_U^{-1}(\mathcal{X})$  est défini par (4.2.10)

♠

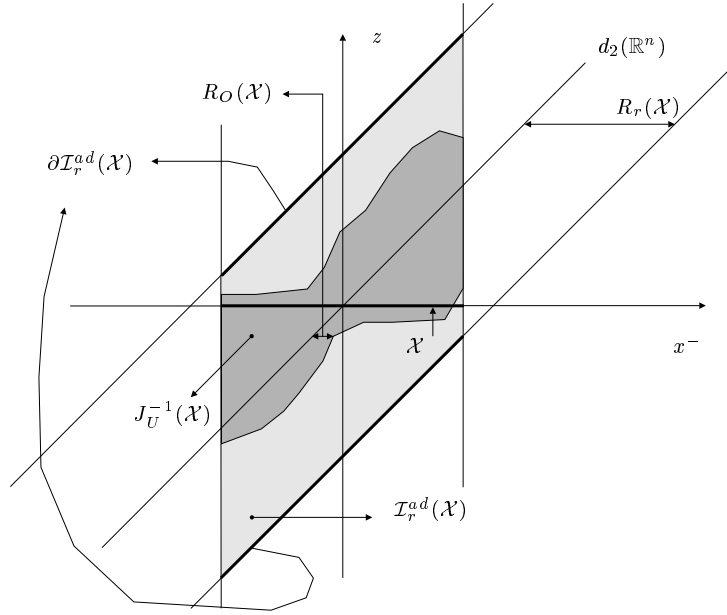


FIG. 4.3: Définition de l'ensemble  $J_U^{-1}(\mathcal{X})$

La proposition suivante énonce les propriétés du rayon d'observabilité-**GHG** qui sont de grand intérêt pour le résultat principal de ce chapitre :

**Proposition 4.2.3.** *Sous les hypothèses du lemme 4.2.1 (observabilité uniforme du linéarisé autour de toute trajectoire résultant d'une commande admissible), le rayon d'observabilité-**GHG** objet de la définition 4.2.1 est strictement positif. En outre, si le gradient  $G$  est globalement uniformément régulier alors*

$$\lim_{\rho(\mathcal{X}) \rightarrow \infty} R_O(\mathcal{X}) = \infty$$

pour tout ensemble compact  $\mathcal{X}$ .



La pertinence de la définition du rayon d'observabilité ci-dessus devient plus claire à la lumière du résultat principal de convergence qui fait l'objet de la section suivante.

### 4.3 Résultats de convergence

Le résultat principal de ce chapitre est le suivant :

**Proposition 4.3.1.** *Soit  $S_0 \subset \mathbb{R}^n$  un ensemble compact contenant l'origine et tous les états initiaux d'intérêt.*

*Si le système linéarisé (le long de la trajectoire du système admissible) est uniformément observable alors il existe un  $\gamma > 0$  suffisamment grand pour que le système dynamique donné par*

$$\dot{z}(t) = f(z(t), u^-(t)) - \gamma G^T [GG^T]^{-1} \sqrt{J} \quad (4.3.1)$$

$$\hat{x}(t) = X(t; t-T; z(t); \tilde{u}_t) \quad (4.3.2)$$

*(dans lequel  $G = G(z(t), x^-(t), \tilde{u}_t)$  et  $J = J(z(t), x^-(t), \tilde{u}_t)$ ) soit un observateur asymptotique pour le système (4.1.1) pourvu que la configuration initiale  $(x^-, z) \in S_0 \times \mathbb{R}^n$  satisfasse pour un certain  $\alpha > 1$  la condition suivante*

$$\|z - x^-\| < R_O \left( B \left( 0, \alpha \rho(S_0) \right) \right) \quad (4.3.3)$$

*En particulier, si le gradient  $G$  est globalement uniformément régulier alors l'équation (4.3.1)-(4.3.2) est un observateur globalement convergent pour (4.1.1) quelque soit la configuration initiale. ♠*

Pour le preuve voir l'Annexe C.

La Proposition 4.3.1 donne clairement un résultat de convergence semi-globale. Ce résultat permet de retrouver le résultat de convergence globale de [4] comme un cas particulier dans le cas où l'hypothèse de régularité globale du gradient du critère est satisfaite.

Il va sans dire que le calcul des quantités clé permettant l'évaluation du rayon d'observabilité est très difficile et devient vite impossible pour des systèmes ayant un nombre d'état important. C'est pour cette raison que le résultat de la Proposition 4.3.1 reste essentiellement théorique. Il permet d'avoir une idée conceptuelle des paramètres qui interviennent dans la convergence de l'observateur à horizon glissant proposé dans [4]. Cependant, il a été largement démontré dans [4] que la mise en œuvre de l'observateur (4.3.1)-(4.3.2) est, elle, tout à fait faisable et loin d'être conceptuelle.

Malgré les difficultés génériques rencontrées lors du calcul du rayon d'observabilité en général, un exemple simple est proposé dans la section suivante afin de bien fixer les

idées de base et les nombreuses définitions associées.

## 4.4 Exemple d'illustration

Dans cette section, un exemple simple est donné afin d'illustrer les résultats et les définitions présentées ci-dessus. Pour ce faire, l'oscillateur de Van der Pol est employé

$$\dot{x}_1 = x_2 \quad (4.4.1)$$

$$\dot{x}_2 = -9x_1 + 2(1 - x_1^2)x_2 \quad (4.4.2)$$

$$y = x_1 \quad (4.4.3)$$

Les deux choix suivants pour l'ensemble  $\mathcal{X}$  de conditions initiales sont considérés

$$\mathcal{X}_1 = [-2, 2] \times [-2, 2] \quad ; \quad \mathcal{X}_2 = [-10, 10] \times [-10, 10]$$

Ces deux choix représente deux situations selon lesquelles, l'ensemble de conditions initiales potentielles est soit  $\mathcal{X}_1$ , soit  $\mathcal{X}_2$ . Il est alors naturelle de s'attendre à ce que le rayon d'observabilité lorsque  $\mathcal{X}_1$  est considéré soit plus grand que lorsque c'est  $\mathcal{X}_2$  qui contient les conditions initiales possibles.

La fonction suivante de trois-variables est employée pour estimer via l'optimisation le rayon d'observabilité-**GHG** de (4.4.1)-(4.4.3) par rapport à  $\mathcal{X}_1$  et  $\mathcal{X}_2$  respectivement,

$$F_\epsilon(x, \theta) = \|G(z_\epsilon(x, \theta), x)\| \quad ; \quad z_\epsilon(x, \theta) = x + \epsilon \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \quad (4.4.4)$$

Selon la définition (4.2.4), calculant le minimum de  $F_\epsilon$  pour  $x \in \mathcal{X}, \theta \in [0, 2\pi]$  et pour  $\epsilon > 0$  croissant, permet de calculer le rayon de Régularité-**GHG**  $R_r(\mathcal{X})$  ou une limite inférieure.<sup>1</sup>

Les Figures 4.4-4.5 montrent l'évolution de la quantité

$$\min_{x \in \mathcal{X}, \theta \in [0, 2\pi]} F_\epsilon(x, \theta) = \min_{x \in \mathcal{X}, \|z-x\|=\epsilon} \|G(z, x)\|$$

dans les deux cas  $\mathcal{X} = \mathcal{X}_1$  et  $\mathcal{X} = \mathcal{X}_2$ . Des Figures 4.4-4.5 il peut être déduit que

$$R_r(\mathcal{X}_1) > 25 \quad ; \quad R_r(\mathcal{X}_2) \approx 0.7$$

et en utilisant ces résultats,  $\bar{J}_r(\mathcal{X}_i)$  peut être calculé selon (4.2.7) en minimisant  $J(z_\epsilon(x, \theta), x)$  (avec  $\epsilon = R_r(\mathcal{X}_2)$ ) sur  $x \in \mathcal{X}, \theta \in [0, 2\pi]$ .

Ceci donne

$$\bar{J}_r(\mathcal{X}_1) \approx 0.0044 \quad ; \quad \bar{J}_r(\mathcal{X}_2) \approx 0.25 \times 10^{-7}$$

---

1. La procédure DCBPOL de la bibliothèque IMSL est utilisée pour accomplir cette tâche d'optimisation.

Le rayon d'observabilité-**GHG** de (4.4.1)-(4.4.3) peut être évalué selon la Définition 4.2.1 pour  $\mathcal{X}_1$  et  $\mathcal{X}_2$  respectivement (voir la Figure 4.6) pour obtenir

$$R_O(\mathcal{X}_1) \approx 0.295 \quad ; \quad R_O(\mathcal{X}_2) = 0.0065$$

Ceci confirme bien ce à quoi nous devons nous attendre, à savoir, plus le domaine des conditions initiales est large, plus il est important d'avoir une erreur initiale faible afin d'être sûr qu'une singularité du gradient est exclue.

Ces résultats montre aussi un autre aspect du résultat, à savoir le fait qu'en utilisant la notion de rayon de convergence, il est possible, pour des systèmes de petite dimension de trouver des conditions qui ne sont pas locales ou du type "*il existe une erreur d'estimation initiale suffisamment petite telle que ...*". En effet, lorsque l'ensemble  $\mathcal{X}_1$  est considéré comme un ensemble de conditions initiales, l'étude précédente montre que l'observateur converge pour toute erreur initiale de moins de 0.295.

## 4.5 Conclusion

Dans ce chapitre, des résultats supplémentaires concernant la version dynamique des observateurs à horizon glissant proposés dans [4] sont présentés. En particulier le concept du rayon de convergence-**GHG**  $R_r(\mathcal{X})$  est proposé. Ce rayon est directement lié à la norme admissible de l'erreur d'estimation initiale garantissant que des singularités locales associées à une approche de gradient de descente ne sont pas rencontrées par les trajectoires de l'état augmenté.

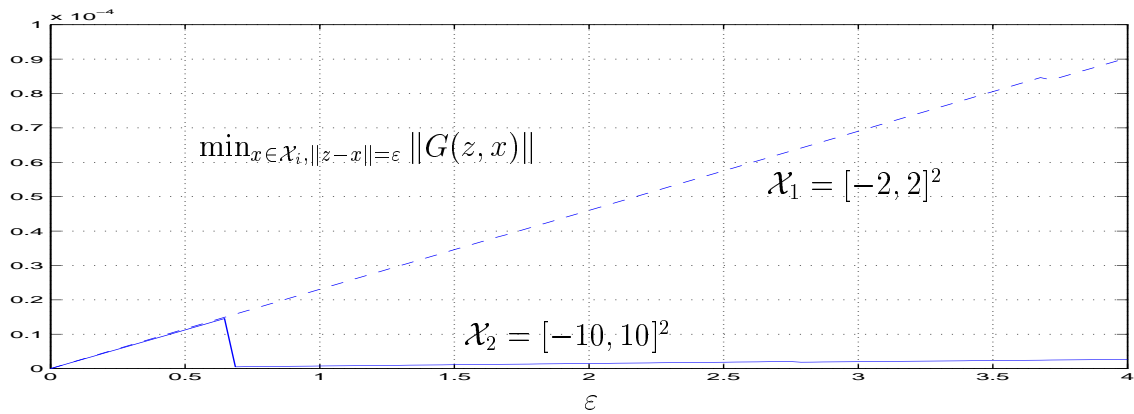


FIG. 4.4: Calcul du rayon de régularité pour le système (4.4.1)-(4.4.3) pour  $\mathcal{X} = \mathcal{X}_1$  et  $\mathcal{X} = \mathcal{X}_2$

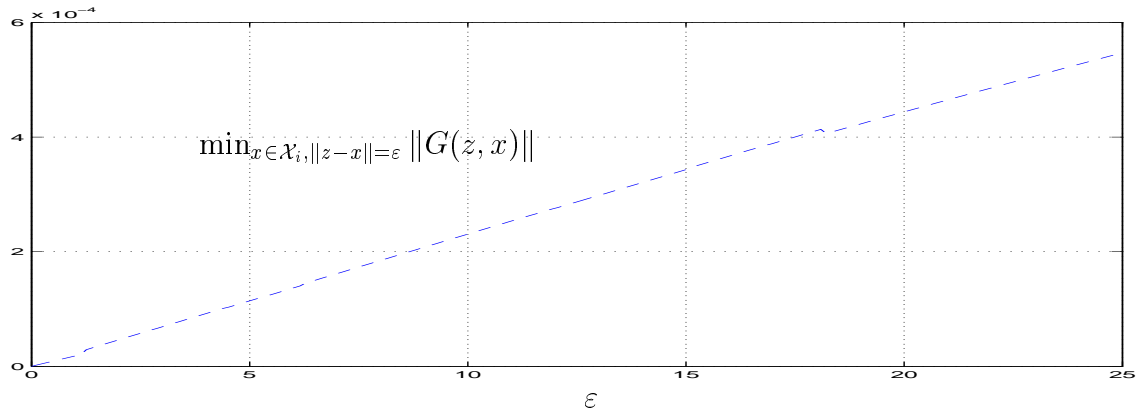


FIG. 4.5: Calcul du rayon de régularité pour le système (4.4.1)-(4.4.3) pour  $\mathcal{X} = \mathcal{X}_1$

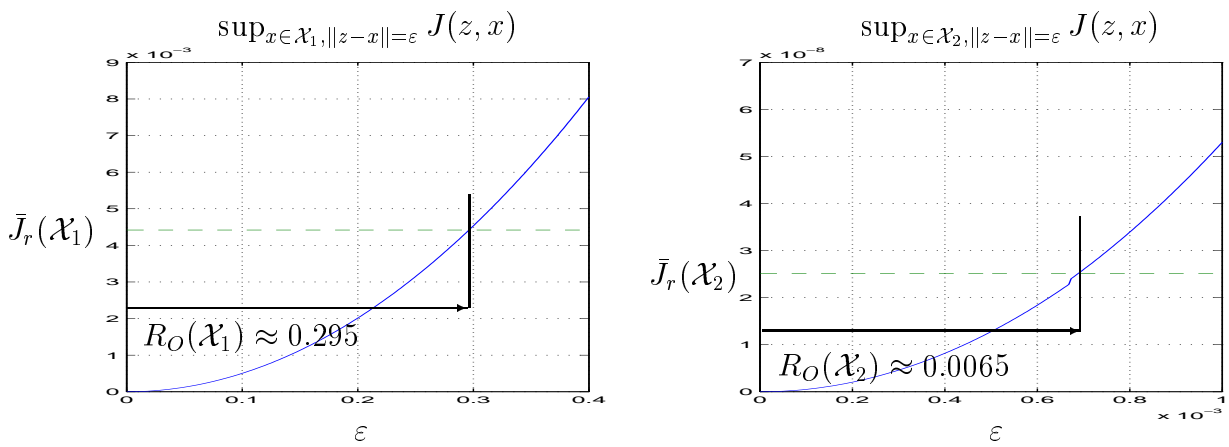


FIG. 4.6: Calcul du rayon d'observabilité pour le système (4.4.1)-(4.4.3) pour  $\mathcal{X} = \mathcal{X}_1$  et  $\mathcal{X} = \mathcal{X}_2$





# Conclusion Générale



# Chapitre 5

## Conclusion Générale

Dans cette partie, un bilan du travail réalisé est présenté.

L'étude comparative des différentes techniques de synthèse d'observateurs pour des systèmes non linéaires a permis de donner un panorama général qui peut aider à choisir entre une technique et une autre en fonction

- Des conditions particulières de structure du système.
- Des performances propres (vitesse de convergence, robustesse par rapport au bruit, aux incertitudes, réglages possibles, etc) de chaque technique de synthèse.

Cette étude bibliographique a aussi servi pour montrer qu'aucune méthodologie ne peut prétendre aboutir systématiquement à un observateur asymptotiquement et globalement convergent pour tout système même uniformément observable et l'étude a montré aussi que ces observateurs souffrent au moins de l'un des inconvénients suivants :

- Conditions structurelles restrictives à satisfaire.
- Les erreurs du modèle et les perturbations qui ne sont pas explicitement prises en considération.
- Calculs lourds à effectuer.

Pour ces raisons, l'un des buts de ce travail a été de proposer un schéma de synthèse qui traite les inconvénients décrits ci-dessus. Plus spécifiquement, l'objectif a été d'aller vers l'obtention d'une nouvelle méthodologie d'optimisation qui essaie de tenir les objectifs suivants :

- *Indépendance vis à vis des conditions structurelles du système de manière à obtenir une approche plus systématique et générique.*

La synthèse de l'observateur proposé n'a pas besoin d'une structure particulière du système à observer.

- *Prise en compte explicite des incertitudes et/ou des perturbations.*

La prise en compte explicite des incertitudes présente l'avantage de tenir compte de la façon précise dont ces incertitudes apparaissent dans la dynamique du système.

- *Temps de calcul réduit.*

En comparaison aux schémas d'observation du type "*horizon glissant*" qui cherchent à chaque période d'échantillonnage le minimum global d'un problème d'optimisation de dimension  $n$ .

Ainsi, le schéma proposé divise d'abord le problème d'optimisation principal en  $n$  problèmes d'optimisation scalaires, où chaque problème scalaire est appelé "*Problème Élémentaire*" où ce problème est transformé en une équation de Hamilton-Jacobi scalaire et cette équation est résolue en employant des codes existantes et efficaces.

Les solutions des différents problèmes scalaires sont alors "analysées" pour définir un nouvel ensemble de problèmes scalaires pour l'itération suivante. Cette analyse est exécutée dans la phase de traitement afin de mener l'algorithme vers la définition de problèmes élémentaires dont la solution est compatible avec le système dynamique dans son intégralité. Ceci est accompli au moyen des signaux de référence obtenus à la sortie de la phase de traitement pour imposer à la solution d'être compatible avec toutes les équations du système.

Une autre façon de procéder est que dans beaucoup de cas et sous réserve de faire un choix judicieux des vecteurs de commande dans les problèmes élémentaires, les équations de Hamilton-Jacobi peuvent prendre la forme particulière des équations de Riccati temps-variant. Ceci permet de trouver la solution optimale de chaque problème élémentaire en résolvant une équation de Riccati temps-variant par les solveurs classiques encore plus performants. Cependant, la transformation d'un problème élémentaire en un problème linéaire suscite parfois de modifier la définition des critères quadratiques.

Bien que les choix des valeurs des pondérations restent assez flexibles, un ensemble de règles a été proposé visant à modifier dynamiquement les coefficients des pondérations dans chaque problème élémentaire afin de forcer l'algorithme itératif à converger vers un point stationnaire cohérent en quittant les pièges que sont les points stationnaires incohérents. Ces notions sont clairement définies à travers des informations disponibles que sont les signaux internes de l'algorithme.

L'autre partie de ce travail a consisté en donner des résultats supplémentaires concernant la version dynamique des observateurs à horizon glissant proposés dans [4]. En particulier le concept du rayon de régularité-**GHG**  $R_r(\mathcal{X})$  a été proposé qui est directement lié à la norme admissible de l'erreur d'estimation initiale garantissant que des singularités locales associées à une approche de gradient de descente ne sont pas rencontrées par les trajectoires de l'état augmenté.

# Annexes



# Annexe A

## Objets et opérateurs différentiels

Dans le paragraphe suivant sont rappelées les définitions de quelques outils de base

### Objets et opérateurs différentiels

Soit  $h$  une fonction réelle  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  et  $f$  et  $g$  deux champs de vecteurs  $f = (f_i)_{1 \leq i \leq n}$  et  $g = (g_i)_{1 \leq i \leq n}$ , définis sur un sous-ensemble ouvert connexe  $U$  de  $\mathbb{R}^n$  dans  $\mathbb{R}^n$ , alors, les définitions suivantes sont présentées [29] :

- La différentielle ou gradient de  $h$  en  $x$

$$dh(x) := \frac{\partial h}{\partial x} = \left( \frac{\partial h}{\partial x_1}, \dots, \frac{\partial h}{\partial x_n} \right) \quad (\text{A.0.1})$$

- La matrice Jacobienne de  $f$  en  $x$

$$\frac{\partial f}{\partial x} := \left( \frac{\partial f_i}{\partial x_j} \right)_{1 \leq i, j \leq n} \quad (\text{A.0.2})$$

- Le crochet de Lie de  $f$  et  $g$  en  $x$

$$[f, g](x) := \frac{\partial g}{\partial x} f(x) - \frac{\partial f}{\partial x} g(x) \quad (\text{A.0.3})$$

Pour répéter plusieurs fois le crochet de Lie, la notation suivante est utilisée

$$\begin{aligned} ad_f^k g &= [f, ad_f^{k-1} g] \quad ; \quad k \geq 1 \\ ad_f^0 g &= g \end{aligned} \quad (\text{A.0.4})$$

- La dérivée de Lie de  $h$  par rapport à  $f$

$$L_f h = \langle dh, f \rangle = \frac{\partial h}{\partial x} f = \sum_{i=1}^n \frac{\partial h}{\partial x_i} f_i \quad (\text{A.0.5})$$



Pour des applications répétées de la dérivée de Lie, la notation suivante est utilisée

$$\begin{aligned}L_f^i h &= L_f(L_f^{i-1} h) \quad ; \quad i \geq 1 \\L_f^0 h &= h\end{aligned}\tag{A.0.6}$$

- La dérivée de Lie de  $dh$  par rapport à  $f$

$$L_f dh = \left( \frac{\partial dh^T}{\partial x} f \right)^T + dh \frac{\partial f}{\partial x}\tag{A.0.7}$$

avec cette définition la relation suivante est vraie

$$dL_f h = L_f dh$$

- Enfin, les dérivées de Lie obéissent la formule de Leibnitz suivante

$$L_{[f,g]} h = \langle dh, [f, g] \rangle = L_g L_f h - L_f L_g h\tag{A.0.8}$$

□

## Annexe B

# Modules Scalaires Parallèles

Dans cette partie du travail les différentes modules qui sont utilisés pour construire l'algorithme de résolution présenté dans le Chapitre 3 sont montrés. Ces modules sont :

- Module Paramètre
- Module FLH
- Module Initial
- Module Résolution
- Module Traitement
- Programme Principal

La programmation de l'algorithme de résolution (des différents modules) est fait en utilisant le compilateur FORTRAN 90. La description des différents composants (variables et procédures internes) de chaque module est exposée ci-dessous.

### B.1 Module Paramètre

Dans ce module les paramètres et les variables globales qui sont utilisés par l'ensemble des modules et par le programme principal par rapport à la définition du problème élémentaire sont définis.

Dans le Chapitre 3 a été présenté le problème élémentaire qui est donné par

$$\dot{\xi} = F(\xi, x^c, \nu, \tau) = f(\xi, x^c, \tau) + g(\xi, x^c, \tau)\nu(\tau) \quad ; \quad \tau \in [t - T, t] \quad (\text{B.1.1})$$

où

- $\xi \in \mathfrak{R}$  est l'état.

- $x^c(\cdot)$  est un signal dont l'évolution est connue sur  $[t - T, t]$ .
- $\nu$  est un vecteur de commande satisfaisant une contrainte du type  $\nu(\tau) \in V$

Les paramètres suivants peuvent être définis :

- $npe$  est un paramètre de type entier qui indique le nombre de problèmes élémentaires.
- $np\_xi$  est un paramètre de type entier qui indique le nombre de points pour  $\xi$ .
- $np\_t$  est un paramètre de type entier qui indique le nombre de points pour  $t$ .
- $n\_var$  est un paramètre de type entier qui indique le nombre total de variables du système.
- $n\_iter$  est un paramètre de type entier qui indique le nombre des itérations à réaliser.
- $n\_entre$  est un paramètre de type entier qui indique le nombre des entrées.
- $n\_perturb$  est un paramètre de type entier qui indique le nombre des perturbations.
- $n\_sortie$  est un paramètre de type entier qui indique le nombre des sorties.
- $epsilonp$  est un paramètre de type réel qui indique si la valeur de l'erreur de la norme à l'infinie entre la solution de la phase de traitement à l'itération  $k$  et la solution de la phase de traitement à l'itération  $(k - 1)$  est dans un point stationnaire.
- $epsilon$  est un paramètre de type réel qui indique si la valeur de l'erreur de la norme à l'infinie entre la solution de la phase de traitement à l'itération  $k$  et la solution de la phase de traitement à l'itération  $(k - 1)$  est dans un point stationnaire cohérent.
- $bmax$  est un paramètre de type réel qui indique la valeur maximale du coefficient du bruit de mesure.
- $vec\_n\_xc$  est un paramètre sous forme de vecteur d'entiers de dimension  $npe$  qui indique le nombre de variables  $x^c$  pour chaque problème élémentaire.
- $vec\_n\_nu$  est un paramètre sous forme de vecteur d'entiers de dimension  $npe$  qui indique le nombre de variables  $\nu$  pour chaque problème élémentaire.

Les variables globales suivantes sont définies :

- $vec\_temps$  est un vecteur de réels de dimension  $np\_t$  qui indique le temps.
- $t_0$  est une variable de type réel qui indique la valeur du temps initial du vecteur  $vec\_temps$ .
- $t_f$  est une variable de type réel qui indique la valeur du temps final du vecteur  $vec\_temps$ .

- $dt$  est une variable de type réel qui indique le pas entre chaque valeur du vecteur  $vec\_temps$ .
- $vec\_perturb$  est un vecteur de réels de dimension  $(np\_t \times n\_perturb)$  qui définit les fonctions qui représentent les perturbations pour chaque problème élémentaire.
- $T\_xi$  est un tableau d'entiers de dimension  $(npe \times 1)$  qui contient dans chaque ligne les numéros des variables à prendre dans le tableau  $T\_var$  (i.e. le numéro de ligne de  $T\_var$ ) correspondant aux états  $\xi$  de chaque problème élémentaire.
- $T\_xi\_ref$  est un tableau d'entiers de dimension  $(npe \times 1)$  qui contient dans chaque ligne les numéros des variables à prendre dans le tableau  $T\_var$  (i.e. le numéro de ligne de  $T\_var$ ) correspondant à la trajectoire de référence  $\xi^{ref}$  de chaque problème élémentaire.
- $T\_xc$  est un tableau d'entiers de dimension  $(npe \times max(vec\_n\_xc))$  qui contient dans chaque ligne les numéros des variables à prendre dans le tableau  $T\_var$  (i.e. le numéro de ligne de  $T\_var$ ) correspondant aux signaux connus  $x^c$  de chaque problème élémentaire.
- $T\_nu$  est un tableau d'entiers de dimension  $(npe \times max(vec\_n\_nu))$  qui contient dans chaque ligne les numéros des variables à prendre dans le tableau  $T\_var$  (i.e. le numéro de ligne de  $T\_var$ ) correspondant à la commande  $\nu$  de chaque problème élémentaire.
- $T\_nu\_ref$  est un tableau d'entiers de dimension  $(npe \times max(vec\_n\_nu))$  qui contient dans chaque ligne les numéros des variables à prendre dans le tableau  $T\_var$  (i.e. le numéro de ligne de  $T\_var$ ) correspondant à la trajectoire de référence  $\nu^{ref}$  de chaque problème élémentaire.
- $T\_val\_min\_max$  est un tableau de réels de dimension  $(n\_var \times 2)$  qui contient les valeurs minimales et maximales de toutes les variables du système.
- $T_q$  est un tableau de dimension  $(npe \times 1)$  qui contient les valeurs réelles positives  $q$  qui servent comme éléments de pondération pour la définition du critère de chaque problème élémentaire.
- $T_r$  est un tableau de dimension  $(npe \times max(vec\_n\_nu))$  qui contient les valeurs réelles positives  $r$  qui servent comme éléments de pondération pour la définition du critère de chaque problème élémentaire.
- $ident$  est une variable de type entier qui s'utilise pour indiquer le numéro de problème élémentaire qui est en train d'être résolu.
- $iter$  est une variable de type entier qui s'utilise pour indiquer le numéro de l'itération qui est en train de s'exécuter.
- $xibar$  est un vecteur de type réel de dimension  $np\_xi$  qui contient les  $\xi$  admissibles.
- $x0$  est un vecteur de type réel de dimension  $npe$  qui contient les conditions initiales des états.

- $u\_connu$  est un tableau de réels de dimension  $(np\_t \times n\_entre)$  qui contient le profil (signal connu) pour chaque entrée.
- $u\_init$  est un vecteur de réels de dimension  $n\_entre$  qui garde le résultat de l'interpolation entre  $vec\_temps$  et  $u\_connu$ .
- $T\_var$  est un tableau de réels de dimension  $(n\_var \times np\_t)$  qui contient l'information de toutes les variables du système. L'ordre qui ont les variables dans  $T\_var$  est le suivant :
  - ★ états.
  - ★ entrées.
  - ★ sorties.
  - ★ perturbations.

Cette façon d'accommoder les variables facilite le remplissage de  $T\_var$  puisque le nombre des états, le nombre des entrées, le nombre des sorties et le nombre des perturbations est connu et un numéro d'identification peut être donné à chaque variable du système, c'est à dire, la ligne où se trouve l'information pour chacune des variables du système. Au début de l'itération  $k$ ,  $T\_var$  contient les valeurs actuelles de toutes les variables du système, cette information est représentée pour les valeurs des variables aux différents instants d'échantillonnage, à la fin de l'itération, une nouvelle estimation de ces valeurs est obtenue. Cette information est utilisée par chaque problème élémentaire qui utilise exclusivement les variables qui le concernent. Suit l'étape de résolution de chaque problème élémentaire et l'étape de traitement (on décrira après ces étapes), dans  $T\_var$  est gardé la nouvelle information de toutes les variables à l'itération  $k$  et cette information représente les nouvelles conditions pour le système à l'itération  $k + 1$ .

- $T\_trait$  est un cube de réels de dimension  $(n\_var \times (np\_t + 1) \times (npe + 1))$ . Dans ce cube sont gardées les solutions pour chaque problème élémentaire et la somme pondérée de ces variables, dans le cas où il existe une variable ou plus qui concernent plusieurs problèmes élémentaires. Dans les premiers  $npe$  tableaux de dimension  $(n\_var \times (np\_t + 1))$ , les solutions de chaque problème élémentaire sont gardées, ainsi, en la colonne  $np\_t + 1$  (la dernière colonne de chaque tableau) on mets un 1 à la ligne correspondant si la variable est concernée par le problème élémentaire et on mets 0 dans cette colonne dans le cas contraire. Donc, dans le tableau  $npe + 1$  (le dernier tableau du cube) la somme des valeurs des premiers  $npe$  tableaux et en la colonne  $np\_t + 1$  de ce tableau est gardée, la somme des éléments de la colonne  $np\_t + 1$  des  $npe$  tableaux est gardée aussi et cette colonne indique le chiffre par lequel chaque ligne de ce dernier tableau doit être divisée.
- $T\_iter$  est un cube de réels de dimension  $(n\_iter \times n\_var \times np\_t)$  où sont gardées les solutions du cube  $T\_trait$  à chaque itération.
- $T\_difiter$  est un cube de réels de dimension  $(n\_iter \times n\_var \times np\_t)$  où sont gardées les solutions de la différence du cube  $T\_iter$  à l'itération  $k$  et à l'itération  $(k - 1)$ .

- $T\_iniaux$  est un tableau de réels de dimension  $(npe \times np\_t)$  où est gardée la valeur d'une référence obtenue à partir du tableau  $T\_var$  pour chaque problème élémentaire. La valeur du tableau  $T\_iniaux$  est gardée dans le cube  $T\_iter$  à la première itération.
- $bruit$  qui est un vecteur de dimension  $np\_t$  contenant le dévie uniforme aléatoire  $(0, 1)$ .
- $bruitm$  qui est un vecteur de dimension  $np\_t$  représentant le bruit de mesure obtenue à partir du vecteur  $bruit$  et la variable  $bmax$

Dans ce module se trouve aussi la définition des paramètres et des variables pour les procédures **DMOLCH**, **DBS2IN**, **DBS2GD**, **DBSNAK**, **RNSET** de la bibliothèque **IMLS**, aussi que pour la procédure **LSODA**.

Les procédures contenues dans le module PARAMÈTRE sont

- **initialise**
- **c\_vec\_temps**
- **c\_xibar**
- **c\_vec\_tgrid**
- **c\_xigrid**

Dans la procédure **initialise** on définit la variable du système (en utilisant son numéro d'identification) qui est utilisée pour chacun des différents tableaux  $T\_xi$ ,  $T\_xi\_ref$ ,  $T\_xc$ ,  $T\_nu$  et  $T\_nu\_ref$ , les valeurs minimales et maximales de chaque variable, les valeurs des éléments du tableau  $T_q$  et du tableau  $T_r$  et on produit un vecteur de nombres aléatoires en utilisant la procédure **DRNUN** de la bibliothèque **IMSL**, qui fournit des nombres pseudo-aléatoires d'une distribution uniforme  $(0, 1)$ . L'utilisation de cette procédure de manière générale est

CALL DRNUN ( $nr, r$ )

où l'argument d'entrée  $nr$  est le nombre de nombres aléatoires à se produire et l'argument de sortie  $r$  est un vecteur de dimension  $nr$  contenant le dévie uniforme aléatoire  $(0, 1)$ .

La procédure **c\_vec\_temps** qui est utilisée pour construire le vecteur de temps, a les arguments d'entrée suivants :

- $t\_deb$  qui indique le temps initial.
- $t\_fin$  qui indique le temps final.

La procédure **c\_vec\_temps** n'a pas des arguments de sortie explicites, puisque le vecteur de temps est représenté par la variable globale  $vec\_temps$  où sont gardées les résultats de cette procédure.

La procédure **c\_xibar** donne comme résultat un vecteur contenant les points pour les différents types de splines utilisées par les procédures **DMOLCH**, **DBS2IN**, **DBSNAK**. Cette procédure a les arguments d'entrée suivants :

- *xibar\_min* qui indique la valeur inférieure du vecteur *xi\_bar*.
- *xibar\_max* qui indique la valeur supérieure du vecteur *xi\_bar*.

La procédure **c\_xibar** n'a pas des arguments de sortie, puisque les résultats sont gardées dans la variable globale *xi\_bar*

Les procédures **c\_vec\_tgrid** et **c\_xigrid** sont construites de la même manière que les procédures **c\_vec\_temps** et **c\_xibar** et les résultats sont gardés dans les variables globales *tgrid* et *xigrid* respectivement (variables qui appartient aux procédures **DBS2IN**, **DBS2GD**, **DBSNAK** ). Ces vecteurs contient les coordonnées en *x* et *y* auxquelles les splines (utilisés par la procédure **DBS2GD**) doivent être évalués.

## B.2 Module FLH

Dans ce module sont définies

- Les équations qui décrivent le système dynamique original.
- La fonction qui est utilisée pour réaliser l'interpolation de  $x^c$ ,  $\nu^{ref}$  et  $\xi^{ref}$ .
- Les fonctions qui décrivent chaque problème élémentaire.
- Les fonctions qui décrivent le critère à minimiser.
- Les fonctions qui décrivent l'Hamiltonien.
- Les fonctions qui décrivent la commande optimale  $\hat{\nu}$ .
- Les équations qui décrivent le système dynamique en boucle fermée.

Le module FLH contient les procédures suivantes :

- **Fun\_System**
- **Fun\_Interp**
- **f**
- **g**
- **Fun\_F**
- **Fun\_L**

- **Fun\_H**
- **nu\_chapeau**
- **Fun\_Syst\_Dif2**

Dans la procédure **Fun\_System** on décrit les équations différentielles du système dynamique qui sont intégrées dans le module INITIAL pour obtenir les signaux de référence. La procédure **Fun\_System** possède les arguments suivants :

- *neq* : de type entier qui indique le nombre d'équations différentielles.
- *t* : de type réel qui indique la variable indépendant.
- *y* : de dimension *neq* de type réel qui indique la variable dépendante.
- *ydot* : de dimension *neq* de type réel qui indique l'équation différentielle qui est intégrée.

La procédure **Fun\_Interp** réalise l'interpolation de  $x^c$ ,  $\nu^{ref}$  et  $\xi^{ref}$ . Cette procédure utilise la variable globale *T\_var* qui n'est pas donnée comme argument explicite. La procédure d'interpolation **Fun\_Interp** dispose les arguments d'entrée suivants :

- *id* : de type entier est utilisée pour indiquer le problème élémentaire qui est en train d'être résolu, c'est à dire, en fonction du problème élémentaire, on sait quelles variables de *T\_var* sont utilisées dans l'interpolation, puisque chaque problème élémentaire utilise différentes variables du tableau *T\_var*.
- *t\_interp* : de type réelle représente l'instant auquel l'interpolation est faite.
- *vec\_temps* : représente le vecteur de temps où est fait le parcours pour retrouver les valeurs entre lesquelles se trouve *t\_interp* pour réaliser l'interpolation.

La procédure d'interpolation **Fun\_Interp** possède les arguments de sortie suivants qui sont les résultats de l'interpolation pour les différentes variables de *T\_var*

- *xc* : représente un signal d'évolution connue.
- *nu\_ref* : représente une trajectoire connue.
- *xi\_ref* : représente une trajectoire connue.

Dans la procédure **Fun\_Interp** la procédure **DCSIEZ** est utilisée qui est une procédure de la bibliothèque **IMSL**, qui fait l'interpolation du type spline cubique et retourne les résultats de l'interpolation à l'instant indiqué par *t\_interp*. L'utilisation de cette procédure de manière générale est

```
CALL DCSIEZ (ndata, xdata, fdata, n, xvec, value)
```

Les arguments d'entrée sont les suivants :

- *ndata* : de type entier qui indique le nombre de points.



- *xdata*: vecteur de type réel de dimension *ndata* contenant les points pour les abscisses.
- *fdata*: vecteur de type réel de dimension *ndata* contenant les points pour les ordonnées.
- *n*: de type entier indique la longueur du vecteur *xvec*.
- *xvec*: vecteur de type réel de dimension *n* contenant les points auxquels l'interpolation doit être évaluée.

L'argument de sortie est le suivant :

- *value*: de type réel contenant les valeurs de l'interpolation aux points dans *xvec*.

La fonction **f** de type réel décrit une partie de chaque problème élémentaire. Les arguments d'entrée pour cette fonction sont les suivants :

- *id*: indique le problème élémentaire qui est en train d'être résolu.
- *xi*: représente l'état du système.
- *xc*: représente un signal d'évolution connue qui a été obtenu dans la procédure **Fun\_Interp**.
- *nu\_ref*: représente le signal de référence (obtenu dans la procédure **Fun\_Interp**) pour la commande  $\nu$ .
- *tau*: représente le temps.

Comme **f** est une fonction, alors elle même est l'argument de sortie.

La fonction **g** de type réel décrit l'autre partie de chaque problème élémentaire. Les arguments d'entrée pour cette fonction sont les suivants :

- *id*: indique le problème élémentaire qui est en train d'être résolu.
- *xi*: représente l'état du système.
- *xc*: représente un signal d'évolution connue qui a été obtenu dans la procédure **Fun\_Interp**.
- *tau*: représente le temps.

Comme **g** est une fonction, alors elle même est l'argument de sortie.

La fonction **Fun\_F** de type réel décrit complètement à chaque problème élémentaire. La fonction **Fun\_F** dispose les arguments d'entrée suivants :

- *id*: indique le problème élémentaire qui est en train d'être résolu.

- $xi$ : représente l'état du système.
- $xc$ : représente un signal d'évolution connue qui a été obtenu dans la procédure **Fun\_Interp**.
- $nu\_ref$ : représente le signal de référence (obtenu dans la procédure **Fun\_Interp**) pour la commande  $\nu$ .
- $nu$ : représente le vecteur de commande.
- $tau$ : représente le temps.

Comme **Fun\_F** est une fonction, alors elle même est l'argument de sortie.

La fonction **Fun\_L** de type réel décrit l'intégrant du critère qui est minimisé pour résoudre le problème optimal. La fonction **Fun\_L** possède les arguments d'entrée suivants :

- $id$ : indique le problème élémentaire qui est en train d'être résolu et par conséquent, quel coefficient de  $T_q$  et de  $T_r$  doit être utilisé.
- $xi$ : représente l'état du système.
- $xi\_ref$ : représente le signal de référence (obtenu dans la procédure **Fun\_Interp**) pour l'état  $\xi$ .
- $nu$ : représente le vecteur de commande.
- $nu\_ref$ : représente le signal de référence (obtenu dans la procédure **Fun\_Interp**) pour la commande  $\nu$ .
- $tau$ : Cette variable représente le temps.

Comme **Fun\_L** est une fonction, alors elle même est l'argument de sortie.

La fonction **Fun\_H** décrit l'Hamiltonien. La fonction **Fun\_H** dispose les arguments d'entrée suivants :

- $id$ : indique le problème élémentaire qui est en train d'être résolu.
- $xi$ : représente l'état du système.
- $xi\_ref$ : représente le signal de référence (obtenu dans la procédure **Fun\_Interp**) pour l'état  $\xi$ .
- $xc$ : représente un signal d'évolution connue qui a été obtenu dans la procédure **Fun\_Interp**.
- $nu$ : représente le vecteur de commande.

- $\nu\_ref$  : représente le signal de référence (obtenu dans la procédure **Fun\_Interp**) pour la commande  $\nu$ .
- $\tau$  : représente le temps.
- $\lambda$  : est un élément de type réel.

Comme **Fun\_H** est une fonction, alors elle même est l'argument de sortie.

La procédure **nu\_chapeau** cherche la commande optimale  $\hat{\nu}$  de chaque problème élémentaire. La procédure **nu\_chapeau** dispose les arguments d'entrée suivants :

- $id$  : indique le problème élémentaire qui est en train d'être résolu.
- $xi$  : représente l'état du système.
- $xc$  : représente un signal d'évolution connue qui a été obtenu dans **Fun\_Interp**.
- $\tau$  : représente le temps.
- $\lambda$  : est un élément de type réel.

La procédure **nu\_chapeau** a l'argument de sortie suivant :

- $optim$  qui est le résultat du problème de minimisation.

$$\hat{\nu}(\tau, \xi, \lambda) = \min_{\nu} H(\xi, \xi^{ref}, x^c, \nu, \nu^{ref}, \tau, \lambda) \quad (\text{B.2.1})$$

La procédure **Fun\_Sys\_Dif2** décrit le problème élémentaire en boucle fermée qui est intégré dans le module **Resolution** pour obtenir la solution optimale. La procédure **Fun\_System\_Dif2** possède les arguments suivants :

- $neq$  : de type entier indique le nombre d'équations différentielles.
- $t$  : de type réel indique la variable indépendante.
- $x$  : de dimension 1 de type réel indique la variable dépendante.
- $xprime$  : de dimension 1 de type réel indique l'équation différentielle (problème élémentaire en boucle fermée) qui est intégrée.

## B.3 Module Initial

Dans ce module sont définies

- Les procédures qui font l'initialisation (intégration) du système dynamique.
- Les procédures qui font le remplissage du tableau  $T\_var$ .

Le module INITIAL contient les procédures suivantes :

- **Integ\_Init**
- **Remplir\_Tvar**

La procédure **Integ\_Init** fait l'intégration du système dynamique original. Les arguments d'entrée de la procédure **Integ\_Init** le sont donnés de forme implicite par les modules PARAMETRE et FLH. Cette procédure dispose les arguments de sortie suivants :

- *vect*: de dimension  $np\_t$  est le vecteur qui définit l'instant  $t$  où la solution de l'intégration a été trouvée.
- *vecx*: de dimension  $(np\_t \times npe)$  est le vecteur qui garde la solution de l'intégration du système dynamique.
- *vecy*: de dimension  $np\_t$  est le vecteur qui représente la solution pour la sortie.
- *vecu*: de dimension  $(np\_t \times n\_entre)$  est le vecteur qui représente les signaux connus après l'interpolation.

Dans la procédure **Integ\_Init**, la procédure **LSODA** qui fait l'intégration du système original est utilisée. L'utilisation de cette procédure de manière générale est

CALL lsoda(F,neq,y,t,tout,itol,rtol,atol,itask,istate,iopt,rwork,lrw,iwork,liw,jac,jt)

Les arguments d'entrée sont les suivants :

- La procédure **F** est donnée par l'utilisateur et représente le système des équations différentielles à résoudre par **LSODA**. La procédure **F** tient la forme  $F(neq, t, y, ydot)$ . Pour notre cas la procédure **F** est donnée par la procédure **Fun\_System** qui se trouve dans le module FLH.
- *neq* indique le nombre des équations différentielles.
- *y* est un vecteur de dimension *neq* qui contient les valeurs initiales.
- *t* représente la valeur initiale de la variable indépendante.
- *tout* premier point où la sortie est désirée.
- *itol* 1 ou 2 si *atol* est un scalaire ou un vecteur.

- *rtol* paramètre de tolérance relative (scalaire).
- *atol* paramètre de tolérance absolue (scalaire ou un vecteur).
- *itask* 1 pour un calcul normal des valeurs de sortie de  $y$  en  $t = tout$ .
- *istate* un drapeau entier. Mettre  $istate = 1$ .
- *iopt* 0 pour indiquer qu'aucune entrée optionnelle est utilisée.
- *rwork* vecteur réel de dimension (au moins)  $22 + neq * \max(16, neq + 9)$
- *lrw* dimension déclarée de *rwork*.
- *iwork* vecteur entier de dimension (au moins)  $20 + neq$ .
- *liw* dimension déclarée de *iwork*.
- **jac** le nom de la procédure pour la matrice Jacobienne.
- *jt* indique le type de Jacobienne. Mettre  $jt = 2$ .

Les arguments de sortie sont les suivants :

- $y$  vecteur qui contient les valeurs calculées du vecteur  $y(t)$ .
- $t$  la valeur de la variable indépendante (normalement *tout*).
- $istate = 2$  si **LSODA** a réussi, négative en cas contraire.

La procédure **Remplir\_Tvar** utilise les résultats donnés par la procédure **Integ\_Init** pour remplir le tableau  $T\_var$  pour chaque problème élémentaire. Donc les arguments d'entrée qui utilise cette procédure sont *id*, *vect*, *vecx*, *vecy* et *vecu*. Comme les résultats sont gardés dans la variable globale  $T\_var$ , alors la procédure **Remplir\_Tvar** n'a pas des argumentes de sortie explicites.

## B.4 Module Résolution

Dans ce module

- L'équation aux dérivées partielles

$$V_\tau + H(\tau, \xi, x^c(\tau), \hat{v}(\tau, \xi, V_\xi), V_\xi) = 0 \quad ; \quad V(T, \xi) = 0 \quad (\text{B.4.1})$$

est résolue.

- Les conditions initiales de  $\xi$ , c'est à dire

$$\hat{\xi}_0 = \min_{\xi} V(t_0, \xi) \quad (\text{B.4.2})$$

où  $V(t_0, \xi)$  représente la solution de (B.4.1) à l'instant  $t_0$ , sont recherchées.

- A partir de la solution de (B.4.1) la dérivée

$$\frac{\partial V(t, \xi)}{\partial \xi} \quad (\text{B.4.3})$$

pour trouver la  $\nu$  optimale en boucle fermé, à savoir  $\hat{\nu}$ , est obtenue.

- La solution de l'équation

$$\dot{\xi} = F(\xi, x^c, \hat{\nu}, \tau) \quad (\text{B.4.4})$$

qui décrit le système dynamique en boucle fermée est calculée.

Le module **RÉSOLUTION** contient les procédures suivantes :

- **Resoudre**
- **xi\_chapeau**
- **gensol**

La procédure **Resoudre** trouve la solution de l'équation (B.4.1). La procédure **Resoudre** dispose les arguments d'entrée suivants :

- *id* : indique le problème élémentaire qui est en train d'être résolu.
- Les procédures externes **Fun\_Vtau** et **fcnbc**.

La procédure **Resoudre** a les arguments de sortie suivants :

- *Vsol* : de dimension  $(1 \times np\_xi)$  indique la solution de l'équation (B.4.1) à l'instant de temps *vec\_temps(j)*.
- *Vbar* : de dimension  $(np\_t \times np\_xi)$  représente la solution de l'équation (B.4.1) pour tous les instants du vecteur de temps *vec\_temps*. Ce tableau se construit de façon rétrograde par rapport au temps, c'est à dire, on part du temps final  $t_f$  jusqu'à le temps initial  $t_0$ .

Dans la procédure **Resoudre**, la procédure **DMOLCH** de la bibliothèque **IMSL** qui résout l'équation (B.4.1) est utilisée.

La procédure **DMOLCH** utilise la méthode des lignes pour résoudre le système d'équations aux dérivées partielles

$$\frac{\partial u_k}{\partial t} = f_k \left( x, t, u_1, \dots, u_M, \frac{\partial u_1}{\partial x}, \dots, \frac{\partial u_M}{\partial x}, \frac{\partial^2 u_1}{\partial x^2}, \dots, \frac{\partial^2 u_M}{\partial x^2} \right)$$

avec les conditions initiales

$$u_k(x, t_0) = u_k^0(x)$$

et les conditions limites

$$\alpha_k u_k + \beta_k \frac{\partial u_k}{\partial x} = \gamma_k \quad \text{en } x = x_1 \quad \text{et en } x = x_N \quad (\text{B.4.5})$$

pour  $k = 1, \dots, M$  où  $M$  est égale au nombre des équations aux dérivées partielles. L'utilisation de cette procédure de manière générale est

CALL DMOLCH(ido, Fcnut, Fcnbc, npdes, t, tend, nx, xbreak, tol, hinit, y, ldy)

Les arguments d'entrée sont les suivants :

- *ido* : de type entier indique l'état du calcul.
- *npdes* : de type entier indique le nombre des équations aux dérivées partielles, dans notre cas  $npdes = 1$ .
- *t* : de type réel indique le temps initial.
- *tend* : de type réel indique la valeur de  $t$  à laquelle la solution est souhaitée.
- *nx* : de type entier indique le nombre de points.
- *xbreak* : est un vecteur de réels de dimension  $nx$  qui contient les points pour l'interpolation avec des polynômes d'ordre 3 (cubic Hermite splines).
- *tol* : de type réel indique la tolérance de l'erreur de l'équation aux dérivées partielles.
- *hinit* : de type réel indique la taille du pas d'intégration initial en  $t$ .
- *ldy* : de type entier indique la dimension  $i$  (le nombre de lignes) du vecteur de solution  $y$ , presque toujours  $ldy = npdes$ .

L'argument de sortie est :

- *y* : est un vecteur de réels de dimension  $(npdes \times nx)$  qui contient la solution à l'instant  $t$ . Pour notre cas,  $y$  représente le profil de la solution de l'équation (B.4.1) à l'instant  $t(j)$ .

La procédure **DMOLCH** nécessite de 2 procédures d'entrée qui doivent être données par l'utilisateur; ces procédures sont **Fcnut** (procédure qui sert pour calculer  $V_\tau$ ) et **Fcnbc** (procédure qui se utilise pour évaluer les conditions limites) qui sont déclarées de forme externe au programme d'appel. La procédure **Fcnut** s'écrit d'une manière générale

FCNUT(npdes,x,t,u,ux,uxx,ut)

Les arguments d'entrée sont les suivants :

- *npdes* : indique le nombre des équations aux dérivées partielles.

- $x$  : représente la variable d'état.
- $t$  : représente le temps.
- $u$  : est un vecteur de réels de dimension  $npdes$  qui contient les valeurs de la variable dépendante  $u$ .
- $ux$  : est un vecteur de réels de dimension  $npdes$  qui contient les valeurs des dérivées de premier ordre de  $u$  par rapport a  $x$ .
- $uux$  : est un vecteur de réels de dimension  $npdes$  qui contient les valeurs des dérivées de second ordre de  $u$  par rapport a  $x$ .

L'argument de sortie est :

- $ut$  : Cette variable est un vecteur de réels de dimension  $npdes$  qui contient les valeurs des dérivées partielles de  $u$  par rapport a  $t$ .

Les fonctions limites qui sont acceptées par **DMOLCH** sont du type (B.4.5), la procédure **Fcnbc** s'écrit de manière générale

$$\text{FCNBC}(npdes,x,t,alpha,beta,gammap)$$

Les arguments d'entrée sont les suivants :

- $npdes$  : indique le nombre des équations aux dérivées partielles.
- $x$  : représente l'état.
- $t$  : représente le temps.

Les arguments de sortie sont les suivants :

- $alpha$  : est un vecteur de réels de dimension  $npdes$  qui contient les valeurs  $\alpha_k$ .
- $beta$  : est un vecteur de réels de dimension  $npdes$  qui contient les valeurs  $\beta_k$ .
- $gammap$  : est un vecteur de réels de dimension  $npdes$  qui contient les valeurs des dérivées

$$\frac{d\gamma_k}{dt} = \gamma'_k$$

**Note** : L'utilisateur doit fournir les valeurs de ces variables.

Pour notre cas, il n'existe pas des conditions limites. Ce qui donne

$$\alpha_1 = \beta_1 = \gamma'_1 = 0$$

Les procédures **Fcnut** (que pour notre cas est la procédure **Fun\_Vtau**) et **Fcnbc** se trouvent dehors du module **Resolution**.

La procédure **xi\_chapeau** est utilisé pour trouver les conditions initiales de l'état



$\xi$  (eq. (B.4.2)). Le calcul de  $\hat{\xi}_0$  est réalisé à partir de la solution de l'équation (B.4.1) et avec une grille dense sur  $\xi$ , c'est à dire,  $\hat{\xi}_0$  est obtenu de une fonction  $V(0, \xi)$  lissée. La procédure **xi\_chapeau** dispose les arguments d'entrée suivants :

- *id* : indique le problème élémentaire qui est en train d'être résolu.
- *Vbar* : de dimension  $(np\_t \times np\_xi)$  représente la solution de l'équation (B.4.1) pour tous les instants du vecteur de temps *vec\_temps*. Mais les valeurs de *Vbar* qui sont utilisées dans cette procédure sont  $Vbar(1, :)$ , en d'autres termes, la solution de l'équation (B.4.1) à l'instant de temps  $t_0$ .

La procédure **xi\_chapeau** a les arguments de sortie suivants :

- *xi\_chap* : représente l'état initial (solution optimale à l'instant  $t_0$ ).

La procédure **genersol** trouve les solutions optimales du système en boucle fermée. La procédure **genersol** possède les arguments d'entrée suivants :

- *id* : indique le problème élémentaire qui est en train d'être résolu.
- *Vbar* : de dimension  $(np\_t \times np\_xi)$  représente la solution de l'équation (B.4.1) pour tous les instants du vecteur de temps *vec\_temps*. À partir de ce tableau, la dérivée  $\partial V(t, \xi)/\partial \xi$  est calculée. Cette dérivée est utilisée pour trouver la commande optimale  $\hat{v}$  en boucle fermée qui est utilisée pour définir l'équation différentielle

$$\dot{\xi} = F(\xi, x^c, \hat{v}, \tau) \quad (\text{B.4.6})$$

La procédure **genersol** a les arguments de sortie suivants :

- *vecxopt* : indique la trajectoire optimale pour l'état qui est la solution de l'équation (B.4.6).
- *vecnuopt* : indique la trajectoire optimale pour la commande.

Le calcul de la dérivée (B.4.3) qui est réalisé dans la procédure **genersol** est obtenu en utilisant les procédures d'interpolation (tensor-product spline) définies dans la bibliothèque **IMSL**. Ces procédures sont **DBSNAK**, **DBS2IN** et **DBS2GD**.

La procédure **DBSNAK** calcule le pas d'une séquence de noeud (not-a-knot spline knot sequence). L'utilisation de cette procédure de manière générale est

CALL DBSNAK(ndata, xdata, korder, xknot)

Les arguments d'entrée sont les suivants :

- *ndata* : de type entier qui indique le nombre de points.
- *xdata* : vecteur de type réel de dimension *ndata* en contenant l'emplacement des points.

- *korder*: ordre du spline

L'argument de sortie est le suivant :

- *xknot*: vecteur de type réel de dimension  $n\text{data} + k\text{order}$  en contenant la séquence de noeud.

La procédure **DBS2IN** calcule un produit bi-dimensionnelle de interpolation, en renvoyant les coefficients du produit de type B-spline. L'utilisation de cette procédure de manière générale est

CALL DBS2IN(*nxd*, *xdata*, *nyd*, *ydata*, *fdata*, *ldf*, *kxord*, *kyord*, *xknot*, *yknot*, *bscoef*)

Les arguments d'entrée sont les suivants :

- *nxd*: de type entier qui indique le nombre de points en la direction *x*.
- *xdata*: vecteur de type réel de dimension *nxd* en contenant les points en la direction *x*.
- *nyd*: de type entier qui indique le nombre de points en la direction *y*.
- *ydata*: vecteur de type réel de dimension *nyd* en contenant les points en la direction *y*.
- *fdata*: vecteur de dimension ( $n\text{xd} \times n\text{yd}$ ) en contenant les valeurs pour être interpolées. *fdata* est la valeur en ( $x\text{data}(I), y\text{data}(J)$ ).
- *ldf*: de type entier qui dénote la principale dimension de *fdata* exactement comme est indiquée dans la déclaration de la dimension du programme d'appel.
- *kxord*: ordre du spline en la direction *x*.
- *kyord*: ordre du spline en la direction *y*.
- *xknot*: vecteur de type réel de dimension  $n\text{xd} + k\text{xord}$  en contenant la séquence de noeud en la direction *x*.
- *yknot*: vecteur de type réel de dimension  $n\text{yd} + k\text{yord}$  en contenant la séquence de noeud en la direction *y*.

L'argument de sortie est le suivant :

- *bscoef*: vecteur de type réel de dimension ( $n\text{xd} \times n\text{yd}$ ) en contenant les coefficients du produit de type B-spline.

La procédure **DBS2GD** évalue la dérivée d'un produit bi-dimensionnelle de interpolation, en donnant son représentation du produit de type B-spline sur un grille. L'utilisation de cette procédure de manière générale est

CALL DBS2GD(*ixder*, *iyder*, *nx*, *xvec*, *ny*, *yvec*, *kxord*, *kyord*,

xknot, yknot, nxcoef, nycoef, bscoef, value, ldvalu)

Les arguments d'entrée sont les suivants :

- *ixder*: ordre de la dérivée en la direction  $x$ .
- *iyder*: ordre de la dérivée en la direction  $y$ .
- *nx*: nombre de points de la grille en la direction  $x$ .
- *xvec*: vecteur de dimension  $nx$  en contenant les coordonnées  $x$  auxquelles le spline doit être évalué.
- *ny*: nombre de points de la grille en la direction  $y$ .
- *yvec*: vecteur de dimension  $ny$  en contenant les coordonnées  $y$  auxquelles le spline doit être évalué.
- *kxord*: ordre du spline en la direction  $x$ .
- *kyord*: ordre du spline en la direction  $y$ .
- *xknot*: vecteur de type réel de dimension  $nxcoef + kxord$  en contenant la séquence de noeud en la direction  $x$ .
- *yknot*: vecteur de type réel de dimension  $nycoef + kyord$  en contenant la séquence de noeud en la direction  $y$ .
- *nxcoef*: nombre de coefficients du B-spline en la direction  $x$ .
- *nycoef*: nombre de coefficients du B-spline en la direction  $y$ .
- *bscoef*: vecteur de type réel de dimension  $(nxcoef \times nycoef)$  en contenant les coefficients du produit de type B-spline.
- *ldvalu*: de type entier qui dénote la principale dimension de *value* exactement comme est indiqué dans la déclaration de la dimension du programme d'appel.

L'argument de sortie est le suivant :

- *value*: la valeur de la dérivée (*ixder*, *iyder*) du spline sur la grille de  $(nx \times ny)$ . La variable *value*( $i, j$ ) contient la dérivée du spline au point  $(xvec(i), yvec(j))$ .

La valeur de la dérivée (B.4.3) est représentée par la variable *value* de dimension  $(np\_t \times np\_xi)$ . La variable *value* est utilisée dans la procédure **Fun\_Syst\_Dif2** qui se trouve dans le module FLH pour obtenir le système des équations différentielles en boucle fermée (B.4.4).

Pour obtenir le profil des trajectoires optimales des états on emploie la procédure **LSODA** (intégration des équations différentielles définies dans la procédure **Fun\_Syst\_Dif2**) et en utilisant ces résultats et la valeur de la variable *value* la commande optimale par retour d'état est retrouvée.

## B.5 Module Traitement

Dans ce module sont définies

- La procédure qui fait le remplissage du cube  $T\_trait$ .
- La procédure qui fait le traitement des variables du système qui sont concernées pour plusieurs problèmes élémentaires.
- La procédure qui compare les solutions obtenues de la phase de traitement à l'itération  $k$  et à l'itération  $(k - 1)$ .

Le module `TRAITEMENT` contient les procédures suivantes :

- **Remplir\_Trait**
- **Traite**
- **Comparaison**

La procédure **Remplir\_Trait** remplit les premiers  $npe$  tableaux de la variable  $T\_trait$  avec les solutions de l'équation différentielle (B.4.6) en fonction du problème élémentaire qui est en train d'être résolu. De plus,

$$\sum_{l=1}^{npe} T\_trait(j, np\_t + 1, l) = \begin{cases} 0 \\ 1 \end{cases} \quad (\text{B.5.1})$$

signifie que la dernière colonne de chaque tableau de la variable  $T\_trait$  possède la valeur de 0 ou 1, selon si la variable  $j$ , ( $j = 1 \dots n\_var$ ) du système est concernée par le problème élémentaire  $l$  ou pas. La procédure **Remplir\_Trait** dispose les arguments d'entrée suivants :

- *id* : indique le problème élémentaire qui est en train d'être résolu.
- *vecxopt* : indique la trajectoire optimale pour l'état qui est la solution de l'équation (B.4.6).
- *vecnuopt* : indique la trajectoire optimale pour la commande.

La procédure **Remplir\_Trait** n'a pas des arguments de sortie explicites, puisque les résultats sont gardés dans la variable globale  $T\_trait$ .

La procédure **Traite** fait le traitement de toutes les variables  $j$ , ( $j = 1 \dots n\_var$ ) concernées pour plusieurs problèmes élémentaires, c'est à dire, une fois que les  $npe$  premiers tableaux de la variable  $T\_trait$  ont été remplis, alors pour le dernier tableau se tient

$$T\_trait(j, i, npe + 1) = \frac{1}{\sum_{h=1}^{npe} T\_trait(j, np\_t + 1, h)} \times \sum_{l=1}^{npe} T\_trait(j, np\_t + 1, l) T\_trait(j, i, l) \quad (\text{B.5.2})$$

où

$$\sum_{h=1}^{npe} T\_trait(j, np\_t + 1, h)$$

est le nombre de problèmes élémentaires concernés par la variable  $j$ ,

$$T\_trait(j, np\_t + 1, l)$$

est 0 ou 1 selon si la variable  $j$  est concernée par le problème élémentaire  $l$  ou pas et

$$T\_trait(j, i, l)$$

est la valeur de la variable  $j$  à l'instant  $vec\_temps(i)$  issue de la résolution du problème élémentaire  $l$ . La procédure **Traite** possède les arguments d'entrée suivants :

- *vecxopt* : indique la trajectoire optimale pour l'état qui est la solution de l'équation (B.4.6).
- *vecnuopt* : indique la trajectoire optimale pour la commande.

La procédure **Traite** n'a pas des arguments de sortie explicites, puisque les résultats du traitement sont gardés dans la variable globale  $T\_var$ .

La procédure **Comparaison** est utilisé pour modifier de façon automatique les valeurs de la pondération  $R$  quand la norme infinie de l'erreur entre la solution de la phase de traitement à l'itération  $k$  et la solution de la phase de traitement à l'itération  $(k - 1)$  soit moins ou égale à la valeur de la variable *epsilon* donnée. Ceci est

$$\|x^{(k)}(\cdot) - x^{(k-1)}(\cdot)\|_{\infty} \leq \epsilon_p \quad ; \quad \epsilon_p = 1 \times 10^{-4}$$

et cette procédure est utilisée aussi pour arrêter le programme quand la norme infinie de l'erreur entre la solution de la phase de traitement à l'itération  $k$  et la solution de la phase de traitement à l'itération  $(k - 1)$  soit moins ou égale à la valeur de la variable *epsilon* donnée. Ceci est

$$\|x^{(k)}(\cdot) - \hat{x}^{(k)}(\cdot)\|_{\infty} \leq \epsilon \quad ; \quad \epsilon = 1 \times 10^{-5}$$

La procédure **Comparaison** dispose les arguments d'entrée suivants :

- *iteration* : indique l'itération qui est en train de s'exécuter.
- *id* : indique le problème élémentaire qui est en train d'être résolu.
- *vecxopt* : indique la trajectoire optimale pour l'état qui est la solution de l'équation (B.4.6).

La procédure **Comparaison** n'a pas des arguments de sortie explicites, puisque les résultats sont modifier la variable globale  $T_r$  ou arrêter le programme.

Dans le **Programme Principal** l'appel aux différents modules décrits auparavant est fait. L'ensemble de tous ces modules représentent l'algorithme de résolution montré dans le Chapitre 3. L'avantage de réaliser l'algorithme en différents modules est la facilité de modifier certains arguments ou procédures sans la nécessité de changer tout le programme.

## B.6 Liste des modules

Dans cette partie, en tant que récapitulatif, les différentes procédures qui sont utilisées dans chaque module sont montrées et aussi les appels qui se réalisent entre les différentes procédures. La forme de lettre *italique* représente les entrées, et la forme de lettre **noire** représente la sortie pour les différentes procédures.

### B.6.1 Module Paramètre

- Paramètres Globales : *npe*, *np\_xi*, *np\_t*, *n\_var*, *n\_iter*, *np\_absis*, *n\_entre*, *n\_perturb*, *n\_sortie*, *epsilonp*, *epsilon*, *bmax*, *vec\_n\_xc(npe)*, *vec\_n\_nu(npe)*
- Variables Globales : *T\_xi(npe)*, *T\_xi\_ref(npe)*, *T\_xc(npe, :)*, *T\_nu(npe, :)*, *T\_nu\_ref(npe, :)*, *ident*, *iter*, *vec\_temps(np\_t)*, *t0*, *tf*, *dt*, *vec\_perturb(np\_t, n\_perturb)*, *xibar(np\_xi)*, *x0(npe)*, *u\_connu(np\_t, n\_entre)*, *u\_init(n\_entre)*, *T\_val\_min\_max(n\_var, 2)*, *T\_q(npe, 1)*, *T\_r(npe, :)*, *T\_var(n\_var, np\_t)*, *T\_traite(n\_var, np\_t + 1, npe + 1)*, *T\_iter(n\_iter, n\_var, np\_t)*, *T\_difiter(n\_iter, n\_var, np\_t)*, *T\_iniaux(npe, np\_t)*, *bruit(np\_t)*, *bruitm(np\_t)*
- Paramètres pour **DMOLCH** : *tol*
- Paramètres et variables pour **LSODA** : *lrw*, *liw*, *jt*, *rtol*, *atol*, *rwork(lrw)*, *itask*, *istate*, *iopt*, *iwork(liw)*, *itol*
- Paramètres et variables pour **DBS2IN**, **DBS2GD**, **DBSNAK** : *ktord*, *kxiord*, *np\_gridt*, *np\_gridxi*, *ldf*, *nptknot*, *np\_xiknot*, *itder*, *ixider*, *ntcoef*, *nxicoef*, *bcoef(np\_t, np\_xi)*, *valueder(np\_gridt, np\_gridxi)*, *tgrid(np\_gridt)*, *xigrid(np\_gridxi)*

- Procédure `initialize()`  
CALL DRNUN(*np\_t*, **bruit**)
- Procédure `c_vec_temps(t_deb, t_fin)`
- Procédure `c_xibar(xibar_min, xibar_max)`
- Procédure `c_vec_tgrid(tgrid_deb, tgrid_fin)`
- Procédure `c_xigrid(xigrid_min, xigrid_max)`
- Procédure `c_vec_perturb(t_fin)`

### B.6.2 Module FLH

- `Fun_System(neq, t, y, ydot)`
- `Fun_Interp(id, t_interp, vec_temps, xc, nu_ref, xi_ref)`  
CALL DCSIEZ(*np\_t*, *vec\_t*, *T\_var*(*T\_xc*(*id*,*i*,:), 1, *t\_interp*, **xc**(**i**))  
CALL DCSIEZ(*np\_t*, *vec\_t*, *T\_var*(*T\_nu\_ref*(*id*,*i*,:), 1, *t\_interp*, **nu\_ref**(**i**))  
CALL DCSIEZ(*np\_t*, *vec\_t*, *T\_var*(*T\_xi\_ref*(*id*,:), 1, *t\_interp*, **xi\_ref**)
- `f(id, xi, xc, nu_ref, tau)`
- `g(id, xi, xc, tau)`
- `Fun_F(id, xi, xc, nu, nu_ref, tau)`
- `Fun_L(id, xi, xi_ref, nu, nu_ref, tau)`
- `Fun_H(id, xi, xi_ref, xc, nu, nu_ref, tau, lambda)`
- `nu_chapeau(id, xi, xc, nu_ref, tau, lambda, optim)`
- `Fun_Syst_Dif2(neq, t, x, xprime)`  
CALL `Fun_Interp(id, t_interp, vec_temps, xc, nu_ref, xi_ref)`  
CALL DCSIEZ(*np\_t*, *vec\_temps*, *valueder*(:,*i*), 1, *t*, **Vxi**(**i**))  
CALL DCSIEZ(*np\_xi*, *xibar*, *Vxi*, 1, *x*, **gradV**)  
CALL `nu_chapeau(id, xi, xc, nu_ref, tau, gradV, optim)`  
`xprime=Fun_F(ident, x, xc, optim, nu_ref, t)`

### B.6.3 Module Initial

- `Integ_Init(vect, vecx, vecy, vecu)`  
CALL DCSIEZ(*np\_t*, *vec\_temps*, *u\_connu*(:,*j*), 1, *t1*, **u\_init**(**j**))  
CALL LSODA (`Fun_System`, *npe*, **x\_integ**, *t1*, **t2**, *itol*, *rtol*, *atol*, *itask*, **istate**,  
*iopt*, *rwork*, *lrw*, *iwork*, *liw*, *jac*, *jt*)
- `Remplir_Tvar(id, vect, vecx, vecy, vecu)`  
CALL `Integ_Init(vect, vecx, vecy, vecu)`

### B.6.4 Module de résolution

- Variables Locales :  $Vsol(1, np\_xi)$ ,  $Vbar(np\_t, np\_xi)$ ,  $xi\_chap(1)$ ,  $vecxopt(np\_t)$
- Resoudre( $id$ , **Vsol**, **Vbar**)  
CALL DMOLCH( $ido$ ,  $Fun\_Vtau$ ,  $fcnbc$ ,  $1$ ,  $t$ ,  $tend$ ,  $np\_xi$ ,  $xibar$ ,  $tol$ ,  $hinit$ , **Vsol**,  $1$ )
- $xi\_chapeau(id, Vbar, xi\_chap)$   
CALL Resoudre( $id$ , **Vsol**, **Vbar**)  
CALL DCSIEZ( $np\_xi$ ,  $xibar$ ,  $Vbar(1,:)$ ,  $np\_absis-1$ ,  $vec\_absis$ , **Vbar\\_interp**)
- $gensol(id, Vbar, vecxopt, vecnuopt)$   
CALL  $xi\_chapeau(id, Vbar, xi\_chap)$   
CALL DBSNAK( $np\_t$ ,  $vec\_temps$ ,  $ktord$ , **tknot**)  
CALL DBSNAK( $np\_xi$ ,  $xibar$ ,  $kxiord$ , **xiknot**)  
CALL DBS2IN( $np\_t$ ,  $vec\_temps$ ,  $np\_xi$ ,  $xibar$ ,  $Vbar$ ,  $ldf$ ,  $ktord$ ,  $kxiord$ ,  $tknot$ ,  $xiknot$ , **bscoef**)  
CALL  $dbs2gd(itder, ixider, np\_gridt, tgrid, np\_gridxi, xigrid, ktord, kxiord, tknot, xiknot, ntcoef, nxicoef, bscoef, valueder, np\_gridt)$   
CALL LSODA ( $Fun\_Sys\_Dif2$ ,  $npe$ , **x\\_integ**,  $t1$ ,  $t2$ ,  $itol$ ,  $rtol$ ,  $atol$ ,  $itask$ , **istate**,  $iopt$ ,  $rwork$ ,  $lrw$ ,  $iwork$ ,  $liw$ ,  $jac$ ,  $jt$ )  
CALL  $Fun\_Interp(id, t\_interp, vec\_temps, xc, nu\_ref, xi\_ref)$   
CALL DCSIEZ( $np\_t$ ,  $vec\_temps$ ,  $valueder(:,k)$ ,  $1$ ,  $t\_interp$ , **Vxi(k)**)  
CALL DCSIEZ( $np\_xi$ ,  $xibar$ ,  $Vxi$ ,  $1$ ,  $vecxopt(i)$ , **gradV**)  
CALL  $nu\_chapeau(id, vecxopt(i), xc, nu\_ref, t\_interp, gradV, optim)$

### B.6.5 Procédures Externes

- $Fun\_Vtau(n, x, t, u, ux, uxx, ut)$   
CALL  $Fun\_Interp(ident, tprime, vec\_temps, xc, nu\_ref, xi\_ref)$   
CALL  $nu\_chapeau(ident, x, xc, nu\_ref, tprime, ux, optim)$   
 $ut = -Fun\_H(ident, x, xi\_ref, xc, optim, nu\_ref, tprime, ux)$   
 $ut = -ut$
- $fcnbc(n, x, t, a, b, gammap)$

### B.6.6 Module Traitement

- Remplir\_Ttrait( $id, vecxopt, vecnuopt$ )  
CALL  $Fun\_Interp(id, t\_interp, vec\_temps, xc, nu\_ref, xi\_ref)$
- Traite( $vecu, vecy$ )
- Comparaison( $iteration, id, vecxopt$ )



## B.6.7 Programme Principal

Le pseudo-codage du Programme Principal est :

```
DO iter = 1, n_iter
  DO ident = 1, npe
    IF iter == 1
      CALL initialize
      CALL c_vec_temps(t0, tf)
      CALL c_vec_tgrid(t0, tf)
      CALL c_vec_perturb(tf)
      CALL c_xibar(T_val_min_max(ident,1), T_val_min_max(ident,2))
      CALL c_xigrid(T_val_min_max(ident,1), T_val_min_max(ident,2))
      CALL Remplir_Tvar(ident, vecdet, vecdex, vecdey, vecdeu)
    ENDIF
    CALL gensol(ident, Vbar, vecxopt, vecnuoptim)
    CALL Remplir_Ttrait(ident, vecxopt, vecnuoptim)
    CALL Traite(vecdeu, vecdey)
    CALL Comparaison(iter, ident, vecxopt)
  ENDDO
ENDDO
```

La figure B.1 montre l'interaction entre les différents modules qui composent l'algorithme.

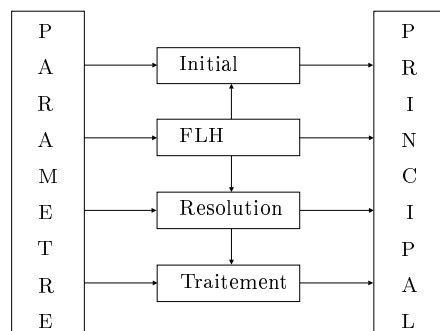


FIG. B.1: Modules de l'algorithme

La figure B.2 montre l'interaction entre les différentes procédures de l'algorithme.

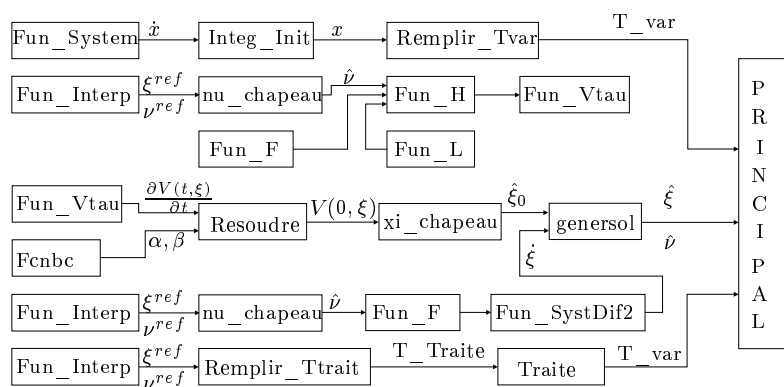


FIG. B.2: Procédures de l'algorithme

# Annexe C

## Preuves

Dans cette partie du travail certaines preuves sont présentées.

### C.1 Idée de la preuve du théorème de vérification 3.3.2

Calculons la dérivée totale de  $V(x(t), t)$  par rapport au temps

$$\frac{dV(x(t), t)}{dt} = [V_x(x(t), t)]^T \dot{x} + V_t(x(t), t) \quad (\text{C.1.1})$$

compte tenu de l'équation de l'état (3.3.4), (C.1.1) s'écrit

$$\frac{dV(x(t), t)}{dt} = [V_x(x(t), t)]^T f(x(t), u(t), t) + V_t(x(t), t) \quad (\text{C.1.2})$$

et par définition de l'Hamiltonien (voir (3.3.8)), on peut remarquer que le premier terme du second membre de l'équation (C.1.2) est égal à

$$[V_x(x(t), t)]^T f(x(t), u(t), t) = H(x(t), u(t), t, V_x(x(t), t)) - L(x(t), u(t), t) \quad (\text{C.1.3})$$

où  $L(\cdot)$  est l'intégrand du critère. D'autre part, d'après (3.3.12), il vient que

$$V_t(x(t), t) = -\hat{H}(x, V_x, t) \quad (\text{C.1.4})$$

En utilisant les équations (C.1.3) et (C.1.4), l'équation (C.1.2) devient

$$\begin{aligned} \frac{dV(x(t), t)}{dt} &= H(x(t), u(t), t, V_x(x(t), t)) - L(x(t), u(t), t) \\ &\quad - \hat{H}(x(t), V_x(x(t), t), t) \end{aligned} \quad (\text{C.1.5})$$

L'intégrale de l'équation (C.1.5) sur l'intervalle  $[t_0, t_f]$  donne

$$\int_{t_0}^{t_f} \frac{dV(x(\tau), \tau)}{d\tau} = \int_{t_0}^{t_f} \left[ H(x(\tau), u(\tau), \tau, V_x(x(\tau), \tau)) - \hat{H}(x(\tau), V_x(x(\tau), \tau), \tau) \right] d\tau$$

$$- \int_{t_0}^{t_f} L(x(\tau), u(\tau), \tau) d\tau \quad (\text{C.1.6})$$

or

$$\int_{t_0}^{t_f} \frac{dV(x(\tau), \tau)}{d\tau} = V(x(t_f), t_f) - V(x(t_0), t_0) \quad (\text{C.1.7})$$

L'équation (C.1.7) utilisée dans l'équation (C.1.6) cause

$$\begin{aligned} V(x(t_f), t_f) - V(x(t_0), t_0) &= \int_{t_0}^{t_f} \left[ H(x(\tau), u(\tau), \tau, V_x(x(\tau), \tau)) - \hat{H}(x(\tau), V_x(x(\tau), \tau), \tau) \right] d\tau \\ &- \int_{t_0}^{t_f} L(x(\tau), u(\tau), \tau) d\tau \end{aligned} \quad (\text{C.1.8})$$

On déplace

$$\int_{t_0}^{t_f} L(x(\tau), u(\tau), \tau) d\tau \quad ; \quad V(x(t_0), t_0)$$

au premier et second membre de (C.1.8) respectivement pour obtenir

$$\begin{aligned} V(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(\tau), u(\tau), \tau) d\tau &= \\ \int_{t_0}^{t_f} \left[ H(x(\tau), u(\tau), \tau, V_x(x(\tau), \tau)) - \hat{H}(x(\tau), V_x(x(\tau), \tau), \tau) \right] d\tau \\ + V(x(t_0), t_0) \end{aligned} \quad (\text{C.1.9})$$

On remarque que le premier membre de (C.1.9) avec  $V(x(t_f), t_f) = \Psi(x(t_f))$  est égal à l'équation (3.3.7), c'est à dire

$$J_T(x(t), u(\cdot), t) = \Psi(x(t_f)) + \int_{t_0}^{t_f} L(x(\tau), u(\tau), \tau) d\tau$$

Comme  $\hat{H}(\cdot)$  est le minimum de  $H(\cdot)$ , alors pour tout  $u(x(\cdot), \cdot) \in \mathbb{U}(x(\cdot))$  la différence  $H(\cdot) - \hat{H}(\cdot)$  est positive, ceci signifie que

$$\int_{t_0}^{t_f} \left[ H(x(\tau), u(\tau), \tau, V_x(x(\tau), \tau)) - \hat{H}(x(\tau), V_x(x(\tau), \tau), \tau) \right] d\tau \geq 0$$

et ainsi on obtient d'après (C.1.9)

$$J_T(x(t), u(\cdot), t) \geq V(x(t_0), t_0)$$

Pour le cas où la stratégie de commande optimale de (3.3.7) est considérée, à savoir

$$u(x(\tau), \tau) = \hat{u}(x(\tau), V_x(x(\tau), \tau), \tau) \quad ; \quad \tau \in [t_0, t_f]$$

on obtient

$$J_T(x(t), u(\cdot), t) = V(x(t_0), t_0)$$

Donc

$$V(x(t_0), t_0) = \min_{u(\cdot)} \int_{t_0}^{t_f} L(x(\tau), u(\tau), \tau) d\tau \quad (\text{C.1.10})$$

est la solution optimale correspondant à condition initial  $(t_0, x_0)$ .

## C.2 Preuve du Corollaire 4.2.2

C'est une conséquence immédiate du lemme 4.2.1. En effet, dans les hypothèses du lemme 4.2.1, le développement de Taylor de  $G(\cdot, x^-, \tilde{u})$  autour de  $x^-$  est donnée par

$$G(z, x^-, \tilde{u}) = \frac{\partial^2 J}{\partial z^2}(x^-, x^-, \tilde{u})(z - x^-) + o(\|z - x^-\|)$$

Maintenant selon (4.2.3), pour tout ensemble compact  $\mathcal{X}$ , il existe un certain  $\mu > 0$  tel que pour  $x^- \in \mathcal{X}$  et tout  $z$  tel que  $\|z - x^-\| \leq \mu$ , on obtienne pour tout  $\tilde{u} \in U^N$

$$\|G(z, x^-, \tilde{u})\| \geq \frac{1}{2}\alpha(x^-)\|z - x^-\|$$

où  $\alpha(\cdot)$  est la fonction invoquée en (4.2.3). Cette dernière inégalité peut aussi être écrite comme suit

$$\left\{ x^- \in \mathcal{X} \quad \text{et} \quad z \neq x^- \quad \text{et} \quad G(z, x^-, \tilde{u}) = 0 \right\} \Rightarrow \left\{ \|z - x^-\| > \mu \right\}$$

ce qui donne clairement (4.2.4) avec  $R_r(\mathcal{X}) \geq \mu > 0$

## C.3 Preuve de la Proposition 4.2.3

La stricte positivité de  $R_O(\mathcal{X})$  vient du fait que

- $J(z, x^-, \tilde{u}) = 0$ ,
- dès que  $(z, x^-) \in d_2(\mathcal{X}) \subset J_U^{-1}(\mathcal{X})$ ,
- que  $\bar{J}_r$  est strictement positif et
- des arguments classiques de continuité sur l'ensemble compact  $J_U^{-1}(\mathcal{X})$ .

La propriété indiquée dans le cas où la régularité uniforme globale est supposée est une conséquence directe de (4.2.9).

## C.4 Preuve de la Proposition 4.3.1

Dans la preuve qui suit la notation suivante est utilisée:

$$\mathcal{X} = \bar{B}\left(0, \alpha\rho(S_0)\right) \tag{C.4.1}$$

la preuve repose sur les trois faits suivants

1. D'abord, si la condition initiale sur  $(z, x^-) \in \mathbb{R}^n \times S_0$  satisfait (4.3.3) alors  $(z, x^-)$  est dans  $J_U^{-1}(\mathcal{X})$ . En outre, pour un  $\gamma > 0$  suffisamment grand, si  $(z, x^-)$  commence dans  $J_U^{-1}(\mathcal{X})$  alors la trajectoire reste dans  $\mathcal{I}_r^{ad}(\mathcal{X})$  pendant quelques périodes de temps  $\Delta t(\mathcal{X}) > 0$ .

2. Tant que les trajectoires restent dans  $\mathcal{I}_r^{ad}(\mathcal{X})$ , l'équation dynamique des observateurs (4.3.1) est bien définie et il existe un  $\beta(\gamma) > 0$  tel que

$$\frac{dJ}{dt} \leq -\beta(\gamma)\sqrt{J} \quad (\text{C.4.2})$$

où  $\beta(\gamma)$  est une fonction propre de  $\gamma$ .

3. Pour un  $\gamma$  suffisamment grand, si la valeur initiale de  $(z, x^-)$  est en  $J_U^{-1}(\mathcal{X})$  alors la solution de (C.4.2) tend à  $J = 0$  en un temps fini qui est plus petit que  $\Delta t(\mathcal{X})$ .

Les trois faits cités ci-dessus permettent clairement d'accomplir la preuve. Commençons par le fait le plus simple à prouver, à savoir le **point (3)**.

### Preuve du point (3)

En effet, il vient immédiatement en intégrant (C.4.2) que

$$J^{1/2}(t) - J^{1/2}(t_0) \leq -\frac{\beta(\gamma)}{2}(t - t_0)$$

dont il devient clair que  $J(t)$  s'annule en un  $t$  satisfaisant :

$$t - t_0 \leq \frac{2}{\beta(\gamma)} J^{1/2}(z(t_0), x^-(t_0), \tilde{u}) \leq \frac{2}{\beta(\gamma)} \bar{J}_r(\mathcal{X}) \quad (\text{C.4.3})$$

par définition de  $J_U^{-1}(\mathcal{X})$  auquel appartient  $(z, x^-)$  par hypothèse. Il est alors suffisant de choisir  $\gamma$  suffisamment grand afin d'avoir

$$\beta(\gamma) > \frac{2\bar{J}_r(\mathcal{X})}{\Delta t(\mathcal{X})}$$

Ceci finit la preuve du **point (3)**.

### Preuve du point (2)

D'après la définition de  $\mathcal{I}_r^{ad}(\mathcal{X})$ , le gradient  $G$  est de rang 1 tant que  $J \neq 0$ . Donc, l'équation dynamique de  $J$  est bien définie tandis que la trajectoire de  $(z, x^-)$  reste dans  $\mathcal{I}_r^{ad}(\mathcal{X})$  qui est l'hypothèse du **point (2)**. La preuve du **point (2)** est facilement obtenue en calculant la dérivée  $dJ/dt$  le long de la trajectoire supposée qui reste dans l'ensemble compact  $\mathcal{I}_r^{ad}(\mathcal{X})$  :

$$\begin{aligned} \frac{dJ}{dt}(t) &= \int_0^T \left[ Y(\tau; 0; z(t), \tilde{u}) - Y(\tau; 0; x^-(t), \tilde{u}) \right]^T \\ &\quad \times \left[ \frac{\partial Y}{\partial z_I}(\tau; 0; z(t), \tilde{u}) f(z(t), u^-(t)) - \frac{\partial Y}{\partial x_I}(\tau; 0; x^-(t), \tilde{u}) f(x^-(t), u^-(t)) \right] d\tau \\ &\quad - \gamma \sqrt{J(z(t), x^-(t), \tilde{u})} \end{aligned}$$

et puisque  $(z(t), x^-(t))$  reste dans l'ensemble compact  $\mathcal{I}_r^{ad}(\mathcal{X})$  aussi et que le profil de commande admissible est dans  $U^N$ , il est possible de trouver un  $K > 0$  tel que la dernière équation implique :

$$\begin{aligned} \frac{dJ}{dt}(t) &\leq K \int_0^T \|Y(\tau; 0; z(t), \tilde{u}) - Y(\tau; 0; x^-(t), \tilde{u})\| d\tau - \gamma \sqrt{J(z(t), x^-(t), \tilde{u})} \\ &\leq \left(\frac{K}{T} - \gamma\right) \sqrt{J} = -\beta(\gamma) \sqrt{J} \end{aligned}$$

avec  $\beta(\gamma) = \gamma - \frac{K}{T}$  qui est clairement propre en  $\gamma$ . Ceci finit la preuve du **point (2)**.

### Preuve du point (1)

- D'abord, il convient de noter que l'état initial  $x^-$  appartient à  $S_0$  qui contient l'origine. Par conséquent, l'état initial  $x^-$  appartient clairement à  $\mathcal{X}$  défini par (C.4.1). En outre, selon (4.3.3),  $z \in B(x^-, R_O(\mathcal{X}))$  et de là, d'après (4.2.11), l'état initial  $(z(t_0), x(t_0)) \in \text{int}\left(J_U^{-1}(\mathcal{X})\right)$   
Partant, par définition :

$$J(z(t_0), x^-(t_0), \tilde{u}) < \bar{J}_r \quad (\text{C.4.4})$$

- Puisque l'ensemble de commandes admissibles est bornée et l'état initial  $x(t_0) \in S_0$ , il existe par définition de  $\mathcal{X}$  (voir (C.4.1)) une période de temps minimale  $\Delta t(\mathcal{X})$  pendant lequel  $x(t)$  reste dans  $\mathcal{X}$ .
- Selon la définition de  $\Delta t(\mathcal{X})$  et  $\mathcal{I}_r^{ad}(\mathcal{X})$ , il vient que pendant l'intervalle de temps  $[t_0, t_0 + \Delta t(\mathcal{X})]$ , la seule manière pour que la trajectoire laisse  $\mathcal{I}_r^{ad}(\mathcal{X})$  est qu'à un certain instant  $t_1 \in ]t_0, t_0 + \Delta t(\mathcal{X})[$  on ait  $(z(t_1), x(t_1)) \in \partial \mathcal{I}_r^{ad}(\mathcal{X})$ . Soit  $t_1^*$  le premier de ces instants, nous prouverons que cela mène à une contradiction.
- En effet, la minimalité de  $t_1^*$  signifie que pendant l'intervalle de temps  $[t_0, t_1^*[$ , l'on ait  $(z(t), x^-(t)) \in \mathcal{I}_r^{ad}(\mathcal{X})$ . Ceci impliquerait, selon le **point (2)** que pendant cet intervalle de temps,  $J$  satisfait l'équation (C.4.2), ceci avec (C.4.4) implique qu'il existe un  $c > 0$  tel que

$$J(z(t_1^*), x^-(t_1^*), \tilde{u}) \leq \bar{J}_r(\mathcal{X}) - c$$

et cela contredit le fait que

$$(z(t_1^*), x^-(t_1^*)) \in \partial \mathcal{I}_r^{ad}(\mathcal{X})$$

puisque cette dernière relation impliquerait selon (4.2.7) que

$$J(z(t_1^*), x^-(t_1^*), \tilde{u}) \geq \bar{J}_r(\mathcal{X})$$

Ceci finit la preuve de la proposition 4.3.1. En effet, les aspects globaux sont des conséquences immédiates de la proposition 4.2.3

# Bibliographie





# Bibliographie

- [1] C. Aboky, G. Sallet, and J.-C. Vivalda. Observers for lipschitz non-linear systems. *International Journal of Control*, 75:204–212, 2002.
- [2] K. Adjallah, D. Maquin, and J. Ragot. Non-linear observer-based fault detection. In *Proceedings of the 3th IEEE Conference on Control Applications*, pages 1115–1120, Glasgow, UK, 1994.
- [3] T. Ahmed-Ali and F. Lamnabhi-Lagarrigue. Sliding observer-controller design for uncertain triangular nonlinear systems. *IEEE Transactions on Automatic Control*, 44:1244–1249, 1999.
- [4] M. Alamir. Optimization based non-linear observers revisited. *International Journal of Control*, 72:1204–1217, 1999.
- [5] M. Alamir and L. A. Calvillo-Corona. Further results on nonlinear receding horizon observers. In *Proceedings of the 40th Conference on Decision and Control*, pages 2948–2954, Orlando, Fl., 2001.
- [6] M. Alamir and L. A. Calvillo-Corona. Further results on nonlinear receding horizon observers. *I.E.E.E. Transactions on Automatic Control*, 47:1184–1188, 2002.
- [7] E. Alcorta-Garcia and P. M. Frank. Deterministic nonlinear observer-based approaches to fault diagnosis: A survey. *Control Engineering Practice*, 5, 5:663–670, 1997.
- [8] A. Alessandri, M. Baglietto, T. Parisini, and R. Zoppoli. Neural state estimators with bounded errors for nonlinear systems. *IEEE Transactions on Automatic Control*, 44, No 11:2028–2042, 1999.
- [9] A. Alessandri, T. Parisini, and R. Zoppoli. Neural approximation for nonlinear finite-memory state estimation. *International Journal of Control*, 67, No 2:275–302, 1997.
- [10] J. Alvarez-Ramirez:95. Observers for a class of continuous tank reactors via temperature measurement. *Chemical Engineering Science*, 50:1393–1399, 1995.
- [11] M. Athans and P. L. Falb. *Optimal Control*. McGraw-Hill, 1966.
- [12] J. A. Ball, J. W. Helton, and M. L. Walker.  $h^\infty$  control for nonlinear systems with output feedback. *IEEE Trans. on Autom. Control*, 38, No 4:546–559, 1993.

- 
- [13] T. Basar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. Academic Press, 1982.
- [14] G. Bastin and D. Dochain. *On-line estimation and adaptive control of bioreactors*. Elsevier, 1990.
- [15] G. Bastin and M. R. Gevers. Stable adaptive observers for nonlinear time-varying systems. *IEEE Transactions on Automatic Control*, 33:650–658, 1988.
- [16] G. Bastin and J. F. Van Impe. Nonlinear and adaptive control in biotechnology: A tutorial. *European Journal of Control*, 1:37–53, 1995.
- [17] W. T. Baumann. Feedback control of multiinput nonlinear systems by extended linearization. *I.E.E.E. Transactions on Automatic Control*, 33:193–197, 1988.
- [18] W. T. Baumann and W. J. Rugh. Feedback control of nonlinear systems by extended linearization. *I.E.E.E. Transactions on Automatic Control*, 31:40–45, 1986.
- [19] W. T. Baumann and W. J. Rugh. Feedback control of nonlinear systems by extended linearization: the multiinput case. *Theory and Applications of Nonlinear Control Systems*, pages 107–113, 1986.
- [20] W. T. Baumann and W. J. Rugh. Feedback control of analytic nonlinear systems by extended linearization. *SIAM Journal on Control and Optimization*, 25:1341–1352, 1987.
- [21] R. W. Beard and T. W. McLain. Successive galerkin approximation algorithms for nonlinear optimal and robust control. *International Journal of Control*, 71, No 5:717–743, 1998.
- [22] C. Benyouseff, B. Dahhou, F. Y. Zeng, and J. L. Rols. Estimation and filtering of nonlinear systems: Application to a wastewater treatment process. *International Journal of Systems Science*, 27:497–505, 1996.
- [23] O. Bernard, Z. Hadj-Sadok, and J. L. Gouzé. Observers for the biotechnological processes with unknown kinetics. application to wastewater treatment. In *Proceedings of the 39th Conference on Decision and Control*, pages 4526–4531, Sidney, Australia, 2000.
- [24] G. Besançon. *Contributions à l'étude et à l'observation des systèmes non linéaires avec recours au calcul formel*. PhD thesis, Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, 1996.
- [25] G. Besançon. Remarks on nonlinear adaptive observer design. *Systems & Control Letters*, 41:271–280, 2000.
- [26] G. Besançon and Q. Zhang. Further developments on adaptive observers for nonlinear systems with application in fault detection. In *Preprints of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain, 2002.

- [27] D. Bestle and M. Zeitz. Canonical form observer design for nonlinear time variable systems. *International Journal of Control*, 38:419–425, 1983.
- [28] D. Birk and M. Zeitz. Extended luenberger observer for nonlinear multivariable systems. *International Journal of Control*, 47:1823–1836, 1988.
- [29] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1975.
- [30] G. Bornard, F. Celle-Couenne, and G. Gilles. Observabilité et observateurs. In *Systèmes non linéaires T. 1, Modélisation-Estimation*, pages 177–221. Masson, 1993.
- [31] G. Bornard and H. Hammouri. A high gain observer for a class for uniformly observable systems. In *Proceedings of the 30th Conference on Decision and Control*, Brighton, GB, 1991.
- [32] M. Boutayeb and M. Darouach. Observers design for linear time varying systems. In *Proceedings of the 39th Conference on Decision and Control*, pages 3183–3187, Sidney, Australia, 2000.
- [33] K. K. Busawon and J. de Leon-Morales. An observer design for uniformly observable nonlinear systems. *International Journal of Control*, 73:1375–1381, 2000.
- [34] K. K. Busawon, M. Farza, and H. Hammouri. Observer design for a special class of nonlinear systems. *International Journal of Control*, 71:405–418, 1998.
- [35] K. K. Busawon and M. Saif. A state observer for nonlinear systems. *I.E.E.E. Transactions on Automatic Control*, 44:2098–2103, 1999.
- [36] L. A. Calvillo-Corona and M. Alamir. Towards the construction of a robust optimal observer. In *Preprints of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain, 2002.
- [37] Y. M. Cho and R. Rajamani. A systematic approach to adaptive observer synthesis for nonlinear systems. *I.E.E.E. Transactions on Automatic Control*, 42:534–537, 1997.
- [38] G. Ciccarella, M. Dalla-Mora, and A. Germani. A luenberger-like observer for nonlinear systems. *International Journal of Control*, 57:537–556, 1993.
- [39] M. Coreless and G. Leitmann. Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. *IEEE Transactions on Automatic Control*, 26:1139–1143, 1981.
- [40] N. Couenne. *Synthèse d’observateurs de systèmes affines en l’état*. PhD thesis, Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, 1988.
- [41] M. Dalla-Mora, A. Germani, and C. Manes. Design of observers from a drift-observability property. *IEEE Transactions on Automatic Control*, 45:1536–1540, 2000.

- [42] D. M. Dawson, Z. Qu, and J. C. Carroll. On the state estimation and output feedback problems for nonlinear uncertain dynamic systems. *Systems & Control Letters*, 18:217–222, 1992.
- [43] E. Delaleau and W. Respondek. Removing input derivatives and lowering their orders in generalized state-space systems. In *Proceedings of the 31th Conference on Decision and Control*, pages 3663–3668, Tucson, Az, 1992.
- [44] F. Deza, E. Busvelle, J. P. Gauthier, and D. Rakotopara. High gain estimation for nonlinear systems. *Systems & control letters*, 18:295–299, 1992.
- [45] X. Ding and P. M. Franck. Nonlinear observer design via an extended luenberger observer canonical form. *Systems & Control Letters*, 15:313–322, 1990.
- [46] C. Edwards, S. K. Spurgeon, and R. J. Patton. Sliding mode observers for fault detection and isolation. *Automatica*, 36:541–553, 2000.
- [47] F. Esfandiari and H. Khalil. Output feedback stabilization of fully linearizable systems. *International Journal of Control*, 56, No 5:1007–1037, 1992.
- [48] A. Fossard and D. Normand-Cyrot. *Systèmes Non Linéaires (3 Volumes), ouvrage collectif-série automatique*. Masson, 1993.
- [49] P. M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - a survey and some new results. *Automatica*, 26, 3:459–474, 1990.
- [50] P. M. Frank and X. Ding. Survey of robust residual generation and evaluation in observer-based fault detection systems. *Journal of Processing Control*, 7, 6:403–424, 1997.
- [51] P. M. Frank, G. Schreier, and E. Alcorta-Garcia. Transformation to state affine system and observer design. In H. Nijmeijer and T. I. Fossen, editors, *New Directions in Nonlinear Observer Design*, pages 399–422. Springer-Verlag, 1999.
- [52] V. Garg and J. K. Hedrick. Fault detection filters for a class of nonlinear systems. In *Proceedings of the American Control Conference*, pages 1647–1651, Seattle, Wa., 1995.
- [53] J. P. Gauthier and G. Bornard. Observability for any  $u(t)$  of a class of bilinear systems. *IEEE Transactions. on Automatic Control*, 26:922–926, 1981.
- [54] J. P. Gauthier, H. Hammouri, and S. Othman. A simple observer for nonlinear systems - application to bioreactors. *IEEE trans. on Automatic Control*, 37:875–880, 1992.
- [55] J. P. Gauthier and I. A. K. Kupka. Observability and observers for nonlinear systems. *SIAM Journal on Control and Optimization*, 32:975–994, 1994.
- [56] D. Georges. Solutions of nonlinear optimal regulator and  $H_\infty$  control problems via galerkin methods. *European Journal of Control*, 2:211–226, 1996.

- [57] A. Glumineau and V. Lopez-Morales. Transformation to state affine system and observer design. In H. Nijmeijer and T. I. Fossen, editors, *New Directions in Nonlinear Observer Design*, pages 59–71. Springer-Verlag, 1999.
- [58] A. Glumineau, H. Moog, and F. Plestan. New algebro-geometric conditions for the linearization by input-output injection. *I.E.E.E. Transactions on Automatic Control*, 41:598–603, 1996.
- [59] H. Hammouri and M. Kinnaert. A new procedure for time-varying linearization up to output injection. *System & Control Letters*, 28:151–157, 1996.
- [60] H. Hammouri, M. Kinnaert, and E. H. El Yaagoubi. Observer-based approach to fault detection and isolation for nonlinear systems. *I.E.E.E. Transactions on Automatic Control*, 44:1879–1884, 1999.
- [61] M. A. Henson and D. E. Seborg. Input-output linearization of general nonlinear processes. *AIChE Journal*, 36, No 11:1753–1757, 1990.
- [62] R. Hermann and A. J. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22:728–740, 1977.
- [63] M. Hou, K. K. Busawon, and M. Saif. Observer design based on triangular form generated by injectiv map. *I.E.E.E. Transactions on Automatic Control*, 45:1350–1355, 2000.
- [64] M. Hou and A. C. Pugh. Observer with linear errors dynamics for nonlinear multi-output systems. *Systems & Control Letters*, 37:1–9, 1999.
- [65] R. Isaacs. Differential games, i, ii, iii, iv. Technical Report RM-1391, 1399, 1411, 1468, RAND Corporation Research Memorandum, 1954–1956.
- [66] A. Isidori. *Nonlinear Control Systems*. Springer Verlag, 1989.
- [67] A. Isidori. *Nonlinear Control Systems*. Springer Verlag, 3rd edition, 1995.
- [68] V. Itkis. *Control Systems of Variable Structure*. John Wiley, 1976.
- [69] N. M. Iyer and A. E. Farrell. Design of a stable adaptive nonlinear observer for an exothermic stirred-tank reactor. *Computers Chemical Engineering*, 20:1141–1147, 1996.
- [70] A. H. Jazwinski. *Stochastic Processes and Filtering Theory, vol. 64 of Mathematics in Science and Engineering*. Academic Press, Inc., 1970.
- [71] N. H. Jo and J. H. Seo. A state observer for nonlinear systems and its application to ball and beam system. *IEEE Transactions on Automatic Control*, 45:968–973, 2000.
- [72] T. Kailath. *Linear Systems*. Prentice Hall, Inc., 1980.
- [73] R. E. Kalman and R. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering D*, 82:35–40, 1960.

- [74] J. C. Kantor. A finite dimensional nonlinear observer for an exothermic stirred-tank reactor. *Chemical Engineering Science*, 44:1503–1510, 1989.
- [75] N. Kazantzis and C. Kravaris. Nonlinear observer design using lyapunov’s auxiliary theorem. *Systems & Control Letters*, 34:241–247, 1998.
- [76] H. Keller. Nonlinear observer design by transformation into a generalized observer canonical form. *International Journal of Control*, 46:1915–1930, 1987.
- [77] S. R. Kou, D. L. Elliott, and T. J. Tarn. Exponential observers for nonlinear dynamic systems. *Information and control*, 29:204–216, 1975.
- [78] G. Kreisselmeier. Adaptive observers with exponential rate of convergence. *IEEE Transactions on Automatic Control*, 22:2–8, 1977.
- [79] G. Kreisselmeier. The generation of adaptive law structures for globally convergent adaptive observers. *IEEE Transactions on Automatic Control*, 24:510–512, 1979.
- [80] A. J. Krener and A. Isidori. Linearization by output injection and nonlinear observers. *Systems & control letters*, 3:47–52, 1983.
- [81] A. J. Krener and W. Respondek. Nonlinear observers with linearizable error dynamics. *S.I.A.M. Journal of Control Optimisation*, 23:197–205, 1985.
- [82] A. J. Krener and M.-Q. Xiao. A necessary and sufficient condition for the existence of a nonlinear observer with linearizable error dynamics. In *Proceedings of the 40th Conference on Decision and Control*, pages 2936–2941, Orlando, Fl., 2001.
- [83] G. Lüders and K. S. Narendra. An adaptive observer and identifier for a linear system. *IEEE Transactions on Automatic Control*, 18:496–499, 1973.
- [84] G. Lüders and K. S. Narendra. A new canonical form for an adaptive observer. *IEEE Transactions on Automatic Control*, 19:117–119, 1974.
- [85] G. Lüders and K. S. Narendra. Stable adaptive schemes for state estimation and identification of linear systems. *IEEE Transactions on Automatic Control*, 19:841–847, 1974.
- [86] F. L. Lewis and V. L. Syrmos. *Optimal Control*. John Wiley & Sons, Inc, 1995.
- [87] C. W. Li and L. W. Tao. Observing nonlinear systems time-variable systems through a canonical observer. *International Journal of Control*, 44:1703–1713, 1986.
- [88] L. Limqueco, J. C. Kantor, and S. Harvey. Nonlinear adaptive observation of an exothermic stirred-tank reactor. *Chemical Engineering Science*, 46:797–805, 1991.
- [89] D. G. Luenberger. An introduction to observers. *I.E.E.E. Transactions on Automatic Control*, 16:592–602, 1971.
- [90] A. F. Lynch and S. A. Bortoff. Non-linear observer design by approximate error linearization. *System & Control Letters*, 32:161–172, 1997.

- 
- [91] A. F. Lynch and S. A. Bortoff. Nonlinear observers with approximately linear error dynamics: The multivariable case. *IEEE Transactions on Automatic Control*, 46:927–932, 2001.
- [92] R. Marino, L. Santosuosso, and P. Tomei. Robust adaptive for nonlinear systems with bounded disturbances. *IEEE Transactions on Automatic Control*, 46:967–972, 2001.
- [93] R. Marino and P. Tomei. Global adaptive observers for nonlinear systems via filtered transformations. *IEEE Transactions on Automatic Control*, 37:1239–1245, 1992.
- [94] R. Marino and P. Tomei. Adaptive observers with arbitrary exponential rate of convergence for nonlinear systems. *IEEE Transactions on Automatic Control*, 40:1300–1304, 1995.
- [95] H. Michalska and D. Q. Mayne. Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40:995–1006, 1995.
- [96] E. A. Misawa and J. K. Hedrick. Nonlinear observers a-state-of-the-art survey. *Journal of Dynamical Systems, Measurements and Control*, 72:1204–1217, 1989.
- [97] K. Nam. An approximate nonlinear observer with polynomial coordinate transformation maps. *IEEE Transactions on Automatic Control*, 42:522–527, 1997.
- [98] S. Nicosia, P. Tomei, and A. Tornambè. Feedback control of elastic robots by pseudolinearization techniques. In *Proceedings of the 25th Conference on Decision and Control*, Athens, Greece, 1986.
- [99] S. Nicosia, P. Tomei, and A. Tornambè. An approximate observer for a class of nonlinear systems. *Systems & Control Letters*, 12:43–51, 1989.
- [100] H. Nijmeijer and A. Van der Schaft. *Nonlinear Dynamical Control Systems*. Springer Verlag, 1990.
- [101] H. Nijmeijer and T. I. Fossen. New directions in nonlinear observer design. In *Vol 244 of Lecture Notes in Control and Information Sciences*. Springer-Verlag, 1999.
- [102] T. Parisini and R. Zoppoli. Neural networks for nonlinear state estimation. *International Journal of Robust and Nonlinear Control*, 4, No 2:231–248, 1994.
- [103] T. Parisini and R. Zoppoli. A receding horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31, No 10:1443–1451, 1995.
- [104] C. De Persis and A. Isidori. A geometric approach to nonlinear fault detection and isolation. *I.E.E.E. Transactions on Automatic Control*, 46:853–865, 2001.
- [105] A. R. Phelps. On constructing nonlinear observers. *SIAM Journal on Control and Optimization*, 29:516–534, 1991.
- [106] F. Plestan. *Linéarisation par injection d'Entrée-Sortie Généralisée et synthèse d'observateurs*. PhD thesis, Thèse de Doctorat, Ecole Centrale de Nantes, France, 1995.



- [107] F. Plestan. A new approach for the linearization of mimo systems by input/output injection. In *Proceedings of the 34th Conference on Decision and Control*, New Orleans, L.A, 1995.
- [108] F. Plestan and A. Glumineau. Linearization by generalized input-output injection. *Systems & Control Letters*, 31:115–128, 1997.
- [109] N. M. Pous, A. Rajab, A. Flaus, and J. M. Engasser. Comparison of estimation methods for biotechnological processes. *Chemical Engineering Science*, 43:1909–1914, 1988.
- [110] T. PH. Proychev and R. L. Mishkov. Transformation of nonlinear systems in observer canonical form with reduced dependency on derivatives of the input. *Automatica*, 29:495–498, 1993.
- [111] S. Raghavan and K. Hedrick. Observer design for a class of nonlinear systems. *International Journal of Control*, 59:515–528, 1994.
- [112] R. Rajamani. Observers for lipschitz nonlinear systems. *I.E.E.E. Transactions on Automatic Control*, 43:397–401, 1998.
- [113] R. Rajamani and Y. M. Cho. Existence and design of observers for nonlinear systems: relation to distance to unobservability. *International Journal of Control*, 69:717–731, 1998.
- [114] R. Rajamani and K. Hedrick. Adaptive observer for active automotive suspensions -theory and experiment. *I.E.E.E. Transactions on Control Systems Technology*, 3:86–93, 1995.
- [115] C. V. Rao and J. B. Rawlings. Nonlinear moving horizon estimation. in *Nonlinear model predictive control, volume 26 of Progress in Systems and Control Theory*, Frank Allgöwer and Alex Zheng, Eds., basel. Birkhäuser, pages 45–69, 2000.
- [116] C. V. Rao, J. B. Rawlings, and J. H. Lee. Constrained linear state estimation - a moving horizon approach. *Automatica*, 37, 10:1619–1628, 2001.
- [117] C. Reboulet and C. Champetier. A new method for linearizing nonlinear systems: the pseudolinearization. *International Journal of Control*, 40:631–638, 1984.
- [118] M. G. Safonov and M. Athans. Robustness and computational aspects of nonlinear stochastic estimators and regulators. *I.E.E.E. Transactions on Automatic Control*, 23:717–725, 1978.
- [119] A. P. Sage and C. C. White III. *Optimum Systems Control*. Prentice-Hall, 1977.
- [120] R. Sanchis and H. Nijmeijer. Sliding controller-sliding observer design for nonlinear systems. *European Journal of Control*, 4:197–208, 1998.
- [121] N. R. Sandell, P. Varaiya, M. Athans, and M. G. Safonov. Survey of decentralized control methods for large scale systems. *I.E.E.E. Transactions on Automatic Control*, 23:108–128, 1978.

- [122] D. D. Siljak. Decentralized control and computations: status and prospects. *Annual Review of Control*, 20:131–141, 1996.
- [123] J.-J. E. Slotine. Sliding controller design for nonlinear systems. *International Journal of Control*, 40, 1984.
- [124] J.-J. E. Slotine, J. K. Hedrick, and E. A. Misawa. Fast nonlinear filtering via garlekin's method. In *Proceedings of the American Control Conference*, Seattle, Wa., 1986.
- [125] J.-J. E. Slotine, J. K. Hedrick, and E. A. Misawa. Nonlinear state estimation using sliding observers. In *Proceedings of the 25th Conference on Decision and Control*, pages 332–339, Athens, Greece, 1986.
- [126] J.-J. E. Slotine, J. K. Hedrick, and E. A. Misawa. On sliding observers for nonlinear systems. *Journal of Dynamics Systems, Measurements and Control*, 109:245–252, 1987.
- [127] J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, Inc., 1991.
- [128] E. D. Sontag and Y. Wang. Output-to-state stability and detectability of nonlinear systems. *Systems & Control Letters*, 29:279–290, 1997.
- [129] H. J. Sussmann. Single input observability of continuous time systems. *Mathematical Systems Theory*, 12:371–393, 1979.
- [130] A. Teel and L. Praly. Global stabilizability and observability imply semi-global stabilizability by output feedback. *Systems & Control Letters*, 22:313–325, 1994.
- [131] A. Teel and L. Praly. Tools for semiglobal stabilization by partial state and output feedback. *SIAM Journal on Control and Optimization*, 33:1433–1488, 1995.
- [132] F. E. Thau. Observing the state of nonlinear systems. *International Journal of Control*, 17:471–479, 1973.
- [133] M-W. L. Thein and E. A. Misawa. Comparison of the sliding observer to several state estimators using a rotational inverted pendulum. In *Proceedings of the 34th Conference on Decision and Control*, pages 3385–3391, New Orleans, L.A, 1995.
- [134] A. Tornambè. High gains observers for nonlinear systems. *International Journal of Systems Science*, 23:1475–1489, 1992.
- [135] M. Törngren and J. Wikander. A decentralized methodology for real-time control applications. *Journal of Control Engineering Practice*, 2:219–228, 1996.
- [136] J. Tsinias. Observer design for nonlinear systems. *Systems & control letters*, 13:135–142, 1989.
- [137] J. Tsinias. Further results on the observer design problem. *Systems & control letters*, 14:411–418, 1990.

- [138] V. I. Utkin. Variable structure systems with sliding mode: a survey. *IEEE Transactions on Automatic Control*, 22, 1977.
- [139] V. I. Utkin. *Sliding Modes and their Application in Variable Structure System*. MIR Publishers, 1978.
- [140] B. Walcott, M. Coreless, and S. Zak. Comparative study of nonlinear observation techniques. *International Journal of Control*, 45:2109–2132, 1987.
- [141] B. Walcott and S. Zak. An observer design technique utilizing the theory of vss. In *Proceedings of the 25th Conference on Decision and Control*, pages 340–349, Athens, Greece, 1986.
- [142] B. L. Walcott and H. Zak. State observation of nonlinear uncertain dynamical systems. *IEEE Transactions on Automatic Control*, 32:166–170, 1987.
- [143] G. B. Wang, S. S. Peng, and H. P. Huang. A sliding observer for nonlinear process control. *Chemical Engineering Science*, 52, No 5:787–805, 1997.
- [144] C. Wen and M. Saif. Robust fault detection in uncertain nonlinear systems via a second order sliding mode observer. In *Proceedings of the 40th Conference on Decision and Control*, pages 573–578, Orlando, Fl., 2001.
- [145] W.M. Wonham. *Linear Multivariable Control: a geometric approach (Third Edition)*. Springer Verlag, 1985.
- [146] X-H. Xia and W-B. Gao. Nonlinear observer design by observer canonical form. *International Journal of Control*, 47:1081–1100, 1988.
- [147] X-H. Xia and W-B. Gao. Nonlinear observer design by observer error linearization. *SIAM Journal on Control and Optimization*, 27:199–216, 1989.
- [148] X. G. Yan, J. Lam, H. S. Li, and I. M. Chen. Decentralized control of nonlinear large-scale systems using dynamic output feedback. *Journal of Optimization Theory and Applications*, 104:459–475, 2000.
- [149] M. Zeitz. The extended luenberger observer for nonlinear systems. *Systems & control letters*, 9:149–153, 1987.
- [150] Q. Zhang and A. Xu. Global adaptive observers for a class of nonlinear systems. In *Proceedings of the 40th Conference on Decision and Control*, pages 3360–3366, Orlando, Fl., 2001.
- [151] G. Zimmer. A new algorithm for approximating the state of nonlinear systems. *Int. Journal of Control*, 24:777–788, 1993.
- [152] G. Zimmer. State observation by on-line minimization. *Int. Journal of Control*, 60:595–606, 1994.



## Résumé :

Ce travail s'inscrit principalement dans le domaine de synthèse d'observateurs pour les systèmes non linéaires. D'une part le développement d'une nouvelle méthodologie de synthèse est proposée et d'autre part des résultats supplémentaires pour les observateurs à horizon glissant sont donnés.

La première partie est plus directement dédiée aux rappels des notions de base d'observabilité des systèmes linéaires et non linéaires, ainsi qu'à la description des principales techniques de synthèse d'observateurs. De cette étude, il a été possible d'énoncer les avantages et les inconvénients de chaque méthode.

Dans une deuxième partie et grâce à l'étude des différentes techniques de synthèse, une nouvelle méthodologie de synthèse d'observateur pour des systèmes non linéaires a été développée. Cette méthodologie divise le problème original en  $n$  problèmes d'optimisation scalaires, dénommés problèmes élémentaires, qui peuvent être résolus en parallèle, ce qui peut être très intéressant pour des systèmes de grande dimension.

La dernière partie est consacrée aux observateurs à horizon glissant plus classiques et on traite le cas où les hypothèses de régularité uniforme globale ne sont pas satisfaites. Le développement technique correspondant est effectué en utilisant les notions de rayon de régularité et de rayon d'observabilité.

**Mots clefs :** observateurs non linéaires, problème élémentaire, équation de Hamilton-Jacobi, horizon glissant, rayon de Régularité, rayon d'Observabilité.

---

## Some Contributions to Receding Horizon Nonlinear Observers

### Abstract:

This work mainly addresses the problem of observer design for nonlinear systems. On the one hand the development of a new methodology of design was proposed and on the other hand further results for receding horizon observers were given.

The first part is more directly dedicated to the recalls of the basic concepts of observability of linear and nonlinear systems, as well as the description of the principal design techniques for observers. From this study, it was possible to state the advantages and the disadvantages of each method.

In a second part and thanks to the study of the different design techniques, a new methodology of observer design for nonlinear systems was developed. This methodology divides the original problem into  $n$  scalar problems of optimization, called elementary problems, which can be solved in parallel which for systems of high dimension can be very interesting to apply it.

The last part is devoted on the more traditional receding horizon observers and the way to treat the case where the global uniform regularity assumptions are not supposed. The corresponding technical description is given through definitions of the regularity radius and the observability radius.

**Keywords:** nonlinear observer, elementary problem, Hamilton-Jacobi equation, receding horizon observer, Regularity radius, Observability radius.