



HAL
open science

Contribution à l'Estimation d'Etat à Horizon Glissant par Méthodes Ensemblistes : Applications à la Surveillance et Détection des Dysfonctionnements sur des Bioprocédés

Héctor-Moisés Valdes-Gonzalez

► **To cite this version:**

Héctor-Moisés Valdes-Gonzalez. Contribution à l'Estimation d'Etat à Horizon Glissant par Méthodes Ensemblistes : Applications à la Surveillance et Détection des Dysfonctionnements sur des Bioprocédés. Automatique / Robotique. Université Joseph-Fourier - Grenoble I, 2002. Français. NNT: . tel-00198362

HAL Id: tel-00198362

<https://theses.hal.science/tel-00198362>

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

N° attribué par la bibliothèque

□□□□□□□□□□□□□□□□

présentée à

L'Université Joseph Fourier - Grenoble I
Sciences - Technologie - Médecine

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER

Spécialité : Automatique - Robotique

préparée au Laboratoire d'Automatique de Grenoble
dans le cadre de l'**Ecole Doctorale Electronique, Electrotechnique, Automatique, Télécommunications, Signal**

présentée et soutenue publiquement

par

Héctor-Moisés VALDES-GONZALEZ

Ingénieur électricien et MSc de l'Ingénierie
Université de Santiago du Chili

le 19 décembre 2002

**Contribution à l'Estimation d'Etat à Horizon Glissant par
Méthodes Ensemblistes : Applications à la Surveillance et Détection
des Dysfonctionnements sur des Bioprocédés**

Directeur de thèse : Prof. Jean-Marie FLAUS

JURY

Mme. Sylviane GENTIL	Président
M. Jean-Luc GOUZE	Rapporteur
M. José RAGOT	Rapporteur
M. Jean-Marie FLAUS	Directeur de thèse
M. Lionel BOILLEREAUX	Examineur
M. Gonzalo ACUÑA	Examineur

Thèse préparée au sein de l'équipe CAPA au LAG-INPG-UJF - France

Ce travail est dédié très humblement, d'abord à mon meilleur ami "Je suis", à l'Alpha et l'Oméga pour la merveille d'avoir fait sa connaissance et pour le privilège qu'il m'a fait partager quand il m'a choisi et nommé comme son ami.

A "Je suis", à l'Alpha et l'Oméga et à mon Consolateur, pour ma vie pleine de bonheur, remplie par sa grâce et ses faveurs.

A mes parents Héctor et Adriana, qui on fait de moi l'enfant le plus heureux du monde, et parce qu'ils m'ont montré que l'effort, l'amour et la foi donnent toujours de bons fruits.

Pour être les meilleurs parents du monde !

A mes frères Eliseo, Gonzalo et Viviana pour l'amour, la joie et l'appui qu'ils m'ont donné chaque fois que j'ai eu besoin de soutien.

A chacun des frères de celui qui est, qui était et qui vient, particulièrement à mon bien aimé

le "Ministre" Javier Vasquez-Valencia au Chili.

A mon ami, le "Ministre" Salomon Bustamante, mon ami Sergio Iturriaga et à ma "famille" en France.

Aux "GBUssiens" de Grenoble.

A l'épouse et aux fils que je n'ai pas encore.

Enfin et tout particulièrement à toi Anne Westmacott, ma douce complice et ma bien aimée.

Femme qui avec ton cœur, tes mots, ton regard bleu, ta simplicité, ton soutien dans la foi et ton amour,

tu m'as montré un chemin qui m'a fait rêver de l'infini... Chaque fois que je suis dans tes bras.

Merci "Je suis"... Merci à tous. Je vous aime tous !

Este trabajo está dedicado humildemente, y ante todo a mi mejor amigo "Yo soy", el Alfa y la Omega, por la maravilla de haberle conocido y por el privilegio del que me ha hecho partícipe cuando me eligió y acogió como su amigo.

A "Yo soy", el Alfa y la Omega y a mi Consolador, por la vida llena de felicidad, inundada por su gracia y favores.

A mis padres Héctor y Adriana, que han hecho de mí, el hijo más feliz del mundo, y porque han sabido mostrarme que el amor, el esfuerzo y la fe dan siempre buenos frutos.

Por ser los mejores padres del mundo !

A mis hermanos Eliseo, Gonzalo y Viviana por el amor, alegría y apoyo que me han dado cada vez que les he necesitado.

A cada uno de los hermanos del que es y que era y que ha de venir, particularmente a mi amado "Ministro" Javier Vasquez Valencia en Chile.

A mi amigo y "Ministro", Salomón Bustamante, a mi amigo Sergio Iturriaga y a mi "familia" en Francia.

A todos los integrantes del GBU de Grenoble.

A la esposa e hijos que aún no tengo.

Finalmente y de forma particular a ti Anne Westmacott, mi dulce cómplice y mi bien amada.

Mujer que con tu corazón, palabras, mirada azul, simpleza, apoyo en la fe

y amor,

me haz mostrado un camino que me ha hecho soñar del infinito... Cada vez que estoy en tus brazos.

Gracias "Yo soy"... Gracias a todos. Les amo !

Avant propos

Ce travail a été réalisé au Laboratoire d'Automatique de Grenoble sous la direction de Monsieur Jean-Marie Flaus, Professeur à L'Université Joseph Fourier. Qu'il puisse trouver ici l'expression de ma profonde gratitude, pour son aide, pour la confiance qu'il m'a témoignée, pour les précieux conseils et les encouragements qu'il m'a prodigué, pour les intéressantes discussions, pour ses commentaires toujours positifs sur mes "idées" durant mon séjour au LAG et tout au long de cette étude.

Il a su me donner le goût des problèmes "réels" et me faire profiter de sa large culture en sciences appliquées, avec sa gentillesse et sa bonne humeur. Sur tout je tiens à le remercier pour sa patience à chaque fois que je n'ai pas voulu faire ce qu'il aurait voulait que je fasse.

Je tiens à remercier Madame Sylviane Gentil, Professeur à l'Institut National Polytechnique de Grenoble, pour l'honneur qu'elle m'a fait d'accepter la présidence de mon jury et pour les fructueuses et "constructives" discussions lors des réunions avec l'équipe CAPA (Conduite Avancée des Procédés Automatisés) voir S³D (Sûreté, Surveillance, Supervision et Diagnostic) au LAG.

Je remercie Monsieur José Ragot, Professeur à l'Institut National Politechnique de Lorraine pour avoir accepté d'être rapporteur de mon travail et pour les nombreuses remarques qui ont démontré le vif intérêt qu'il a porté à mes travaux de recherche.

A Monsieur Jean-Luc Gouzé, Directeur de Recherches à l'INRIA de Sophia-Antipolis, j'aimerais lui exprimer ma gratitude pour avoir accepté d'être rapporteur de mon travail et pour son honnêteté.

Je remercie Monsieur Lionel Boillereaux du E.N.I.T.I.A.A de Nantes, pour avoir accepté de participer au jury, et parce que je me suis largement inspiré de ses travaux de recherche.

Je remercie également Monsieur Gonzalo Acuña, de l'Université de Santiago du Chili, pour son amitié, pour avoir participé au jury de cette thèse et pour les nombreuses discussions qui ont été concrétisées par des publications de travaux conjoints, et parce que si je suis venu en France, c'est en grande partie grâce à lui.

Je ne peux pas oublier les thésards de mon équipe, les amis Chiliens et Mexicains de Grenoble... sans qui l'ambiance n'aurait pas été la même.

A propos de l'ambiance, je ne peux pas oublier de remercier toutes les personnes du laboratoire qui font régner une ambiance "sympathique" et "amicale" propice au travail de groupe. Comment quelqu'un pourrait-il oublier les "sourires" et les "bonjours" attentifs de chaque matin?

Pour ceux qui ne lissent pas entre les lignes : "Il y a un *problème intéressant* à résoudre, et ce n'est pas chez les thésards".

Ce travail de thèse a été réalisé avec le soutien financier du Gouvernement Français, du Gouvernement Chilien et de l'Université de Santiago du Chili. Merci beaucoup, je vous en suis très reconnaissant.



Chile, fértil provincia y señalada
en la región antártica famosa,
de remotas naciones respetada
por fuerte, principal y poderosa;
la gente que produce es tan granada,
tan soberbia, gallarda y belicosa,
que no ha sido por rey jamás regida
ni a extranjero dominio sometida.

Es Chile norte sur de gran longura,
costa del nuevo mar, del Sur llamado,
tendrá del este a oeste de angostura
cien millas, por lo más ancho tomado;
bajo el polo Antártico en altura
de veinte y siete grados, prolongado
hasta do el mar Océano y chileno
mezclan sus aguas por angosto seno.

LA ARAUCANA - Canto Primero

Alonso de Ercilla y Zúñiga

1569 - 1589 (selección)

Table des matières

Chapitre 1 Introduction	17
1.1 Le problème d'estimation classique	19
1.2 Brève revue bibliographique.....	21
1.2.1 <i>Observateurs linéaires</i>	21
1.2.2 <i>Observateurs non-linéaires</i>	22
1.3 Contributions de la thèse	28
1.3.1 <i>Objectif général</i>	28
1.3.2 <i>Objectifs spécifiques</i>	29
1.4 Organisation et présentation du travail réalisé	29
Chapitre 2 IMHSE : Un observateur numérique ensembliste	33
2.1 Caractérisation des incertitudes	34
2.1.1 <i>Quelques aspects à propos de l'analyse par intervalles</i>	35
2.2 Le problème de l'observateur ensembliste	39
2.2.1 <i>Définition d'un observateur ensembliste</i>	42
2.3 Solution du problème d'estimation ensembliste.....	44
2.3.1 <i>Principe de la méthode à horizon glissant par intervalles</i>	44
2.3.2 <i>L'algorithme pour la méthode IMHSE</i>	45
2.3.3 <i>L'algorithme d'obtention de trajectoires</i>	50
2.3.4 <i>Prédiction de l'espace de recherche</i>	50
2.4 L'observateur IMHSE et les mesures hors-ligne.....	52
2.5 L'observateur IMHSE et les systèmes décrits par des modèles hybrides... 53	
2.5.1 <i>Automates hybrides</i>	54
2.5.2 <i>Automate hybride ensembliste</i>	55

2.6	La longueur de l'horizon.....	57
2.6.1	<i>Effet d'une longueur d'horizon trop courte.....</i>	57
2.6.2	<i>Relation avec le temps de réponse du système</i>	58
2.6.3	<i>Bornes pour l'horizon.....</i>	58
2.7	IMHSE globalement convergente	58
2.7.1	<i>Besoin d'une technique globale d'optimisation.....</i>	59
2.8	Stabilité et convergence de la méthode IMHSE	59
2.8.1	<i>Cas 1 : Sans filtrage</i>	59
2.8.2	<i>Cas 2 : Avec filtrage</i>	61
2.9	Conclusions	63
Chapitre 3 Optimisation globale par intervalles		65
3.1	Survol général des méthodes d'optimisation globale.....	66
3.1.1	<i>Méthodes probabilistes.....</i>	66
3.1.2	<i>Méthodes déterministes</i>	68
3.2	Une méthode d'optimisation globale par intervalles.....	69
3.2.1	<i>Faisabilité ou non-faisabilité.....</i>	70
3.2.2	<i>Le test du point central.....</i>	71
3.2.3	<i>Test de monotonie.....</i>	71
3.2.4	<i>Test de non-convexité</i>	72
3.2.5	<i>Test de Newton par intervalles.....</i>	72
3.2.6	<i>L'algorithme G-Optimisation</i>	73
3.2.7	<i>Complexité de l'algorithme d'optimisation global</i>	76
3.3	Matlab et l'outil ia.jar	78
3.3.1	<i>Définition d'une fonction avec "ia.jar"</i>	78
3.4	Optimisation globale avec précision de sortie bornée	81
3.5	L'outil "ia.jar" et le problème optimal (2.10) sur un horizon de temps fini..	85
3.6	Conclusions	88
Chapitre 4 Observabilité numérique : Approche par intervalles		89
4.1	Etats indistinguables.....	90
4.2	Réduction de la précision ϵ autour de la référence.....	93
4.2.1	<i>Systèmes observables.....</i>	93
4.2.2	<i>Systèmes non observables : Cas 1</i>	94
4.2.3	<i>Systèmes non observables : Cas 2.....</i>	94

4.3	Indicateur de qualité ou indice d'observabilité	94
4.3.1	<i>Propriétés de l'indicateur de l'indicateur de discernabilité.....</i>	99
4.4	L'observabilité pas à pas avec IMHSE	100
4.5	Conclusions	101
Chapitre 5 Détection de dysfonctionnements du modèle : Un observateur multi-		
	modèles	103
5.1	IMHSE Multi-modèles.....	106
5.1.1	<i>L'algorithme multi-modèles de base</i>	<i>107</i>
5.1.2	<i>Remarques à propos de l'implémentation de base.....</i>	<i>108</i>
5.1.3	<i>Garantie sur tous les comportements prévus.....</i>	<i>109</i>
5.1.4	<i>Sélection du meilleur modèle.....</i>	<i>110</i>
5.1.5	<i>L'algorithme IMHSE Multi-Modèles</i>	<i>112</i>
5.2	Conclusions	113
Chapitre 6 Applications et discussions		115
6.1	Cas 1 : Application à un procédé batch générique	117
6.1.1	<i>Conditions générales du test</i>	<i>118</i>
6.1.2	<i>Résultats.....</i>	<i>119</i>
6.2	Cas 2 : Application à un bioprocédé générique décrit par un modèle hybride	138
6.2.1	<i>Modèle du système.....</i>	<i>138</i>
6.2.2	<i>Résultats.....</i>	<i>140</i>
6.3	Cas 3 : Application à la détection de systèmes non-observables ou faiblement observables.....	143
6.3.1	<i>Effort nécessaire pour distinguer certains états</i>	<i>147</i>
6.4	Cas 4 : Application à un modèle de fermentation sur substrat solide	149
6.4.1	<i>Conditions générales du test</i>	<i>150</i>
6.4.2	<i>Résultats.....</i>	<i>152</i>
6.5	Cas 5 : Détection de dysfonctionnement par observateur ensembliste multi-modèles	155
6.5.1	<i>Description du système</i>	<i>155</i>
6.5.2	<i>Conditions générales du test</i>	<i>156</i>
6.6	Cas 6 : Application à un modèle de fermentation Bioréacteur à échelle pilote	160
6.6.1	<i>Principe de fonctionnement.....</i>	<i>160</i>

6.6.2 Conditions générales du test	161
6.6.3 Résultats.....	162
Chapitre 7 Conclusions générales et perspectives.....	167
7.1 Perspectives	172
Bibliographie	173
Annexe A Analyse de la complexité de l'algorithme 3 d'optimisation globale	183
A.1 Complexité de l'algorithme	185
A.1.1 Détermination de la profondeur L	188
Annexe B Application de ia.jar à un problème différentiable d'optimisation globale	191
Annexe C L'algorithme de la méthode MHSE-BFGS	193
Annexe D L'algorithme du Filtre de Kalman Etendu	195
Annexe E L'algorithme heuristique MHSE-SA et une application.....	197
E.1 Procédure d'optimisation SA.....	198
E.2 Application.....	199

Liste des figures

Figure 2.1 : Problème d'estimation ensembliste	41
Figure 2.2 : Exemples de trajectoires bornées possibles pour le problème d'estimation ensembliste	43
Figure 2.3 : Principe de méthode ensembliste IMHSE	47
Figure 2.4 : La Prédiction de l'espace de recherche	51
Figure 2.5 : Mesures en ligne et hors ligne.....	53
Figure 2.6 : Automate hybride.....	55
Figure 2.7 : Etat intervalle appartenant à deux places	57
Figure 3.1 : Solution avec $B_\varepsilon = 0.05$ et $\tau = 20\%$	84
Figure 3.2 : Solution avec $B_\varepsilon = 0.05$ et $\tau = 10\%$	84
Figure 3.3 : Solution avec $B_\varepsilon = 0.05$ et $\tau = 5\%$	85
Figure 4.1 : Ensemble d'états intervalles en \mathbb{B} indistinguables.....	90
Figure 4.2 : α -voisinage sur la sortie du système	91
Figure 4.3 : δ -voisinage d'un état x sur un domaine admissible.....	92
Figure 4.4 : Trajectoire de sortie α -bornée et un δ -voisinage d'un état x_0	92
Figure 4.5 : Visualisation graphique de la propriété de ε -observabilité	93
Figure 4.6 : Visualisation graphique d'un système non observable : cas 1.....	95
Figure 4.7 : Visualisation graphique d'un système non observable : cas 2.....	96
Figure 4.8 : Sous-ensemble indiscernable \mathbb{B}	98
Figure 4.9 : Système observable sur tout le domaine admissible.....	101
Figure 4.10 : Système non-observable localement sur le domaine admissible.....	101
Figure 5.1 : a) Biomasse mesurée (trois possibilités) b) Gaz instantanés et Urea	104
Figure 5.2 : Modèle choisi de la biomasse, Urea et GA_3	105
Figure 5.3 : Méthode IMHSE avec commutation de modèle interne.....	106
Figure 5.4 : Domaines admissibles prédits pour tout modèle de la famille.....	107

Figure 5.5 : Union de tous les domaines admissibles compatibles en une seule boîte	110
Figure 6.1 : Sortie non bruitée	119
Figure 6.2 : Biomasse simulée (ligne continue) et estimations	119
Figure 6.3 : Substrat simulé (ligne continue) et estimations	120
Figure 6.4 : Taux spécifique de croissance simulé (ligne continue)	120
Figure 6.5 : Indicateurs d'observabilité numérique et de discernabilité	120
Figure 6.6 : Sortie bruitée	122
Figure 6.7 : Biomasse simulée (ligne continue) et estimations	122
Figure 6.8 : Substrat simulé (ligne continue) et estimations	123
Figure 6.9 : Taux spécifique de croissance simulé (ligne continue)	123
Figure 6.10 : Indicateurs d'observabilité numérique et de discernabilité	123
Figure 6.11 : Sortie bruitée considérée	125
Figure 6.12 : Biomasse simulée (ligne continue) et estimations	125
Figure 6.13 : Substrat simulé (ligne continue) et estimations	126
Figure 6.14 : Taux spécifique de croissance simulé (ligne continue).....	126
Figure 6.15 : Indicateurs d'observabilité numérique et de discernabilité	126
Figure 6.16 : Sortie bruitée considérée	128
Figure 6.17 : Biomasse simulée (ligne continue) et estimations	128
Figure 6.18 : Substrat simulé (ligne continue) et estimations	129
Figure 6.19 : Taux spécifique de croissance simulé (ligne continue).....	129
Figure 6.20 : Indicateurs d'observabilité numérique et de discernabilité	129
Figure 6.21 : Sortie bruitée considérée	131
Figure 6.22 : Biomasse simulée (ligne continue) et estimations	131
Figure 6.23 : Substrat simulé (ligne continue) et estimations	132
Figure 6.24 : Taux spécifique de croissance simulé (ligne continue).....	132
Figure 6.25 : Mesures hors ligne disponibles du substrat	132
Figure 6.26 : Indicateurs d'observabilité numérique et de discernabilité	133
Figure 6.27 : Sortie bruitée considérée	134
Figure 6.28 : Biomasse simulée (ligne continue) et estimations	134
Figure 6.29 : Substrat simulé (ligne continue) et estimations	135
Figure 6.30 : Taux spécifique de croissance simulé (ligne continue).....	135
Figure 6.31 : Mesures hors ligne disponibles du substrat	135
Figure 6.32 : Indicateurs d'observabilité numérique et de discernabilité	136
Figure 6.33 : Loi hybride entre le taux spécifique de croissance et le substrat.....	139
Figure 6.34 : Conditions de transitions du modèle hybride.....	139

Figure 6.35 : État intervalle dont plusieurs zones de commutation sont concernées	140
Figure 6.36 : Biomasse simulée (ligne continue) et estimée (lignes intervalles)	141
Figure 6.37 : Substrat simulé (ligne continue) et estimé (lignes intervalles)	141
Figure 6.38 : Taux spécifique de croissance simulé (ligne continue)	141
Figure 6.39 : Ensemble $[\hat{\Theta}]_{sh}$ (biomasse et substrat)	143
Figure 6.40 : Ensemble $[\hat{\Theta}]_{sh}$ d'états indiscernables déterminés en début d'horizon	145
Figure 6.41 : Concentration de biomasse et substrat estimés par FKE (lignes segmentés) et par MHSE-BFGS (grosses lignes continues)	145
Figure 6.42 : Divergence des estimations face à une perturbation dans la biomasse	145
Figure 6.43 : Ensemble $[\hat{\Theta}]_{sh}$ d'états indiscernables en début d'horizon	146
Figure 6.44 : Ensemble $[\hat{\Theta}]_{sh}$ d'états indiscernables en début d'horizon (k=0 hrs)	147
Figure 6.45 : Conditions de transitions du modèle hybride pour NI_k	150
Figure 6.46 : Biomasse simulée (ligne continue) et estimée (lignes intervalles)	152
Figure 6.47 : Biomasse mesurée simulée (ligne continue), estimée-intégrée	153
Figure 6.48 : Azote intermédiaire simulé (ligne continue) et estimé	153
Figure 6.49 : Taux spécifique de croissance simulé (ligne continue) et estimé	154
Figure 6.50 : Urea simulé (ligne continue) et (o) mesuré	154
Figure 6.51 : Une famille de Van-Der-Pol prévue	156
Figure 6.52 : Estimation bornée de l'état x1	157
Figure 6.53 : Estimation bornée de l'état x2	157
Figure 6.54 : Estimation bornée de l'état x1	158
Figure 6.55 : Estimation bornée de l'état x2	158
Figure 6.56 : Estimation bornée de l'état x1	159
Figure 6.57 : Estimation bornée de l'état x2	159
Figure 6.58 : La matière sèche (M_s) simulée et estimée	163
Figure 6.59 : Estimations de la quantité d'eau (X_w) simulée et estimée	163
Figure 6.60 : Modèles de dysfonctionnement	164
Figure 6.61 : Variations du paramètre δ_1	164
Figure 6.62 : Variations du paramètre δ_2	164
Figure 6.63 : Taux expérimental de CO_2	165
Figure 6.64 : Apports d'eau fraîche (F_w)	165
Figure 6.65 : Evaporation pendant la fermentation (W)	165

Chapitre 1

Introduction

L'automatique appliquée à des systèmes réels, souvent représentés par des modèles de différentes complexités, est confrontée au problème du manque d'instrumentation en ligne (l'état n'est pas entièrement accessible à la mesure), ce qui implique que dans de nombreux procédés, les variables de qualité permettant le suivi et la conduite ne sont pas mesurées. Les performances de la commande des procédés sont donc affectées, par exemple, parce que la commande d'un système nécessite normalement la connaissance complète de son état. Mais aussi parce que cette connaissance est aussi affectée par le retard avec lequel certaines valeurs mesurées sont fournies. C'est ce problème qui a motivé depuis des années la synthèse des observateurs d'état. Autrement dit, nous cherchons à synthétiser un système auxiliaire, qui puisse fournir une estimation de tout l'état ou au moins de celui qui n'est pas susceptible d'être mesuré. La synthèse se fait en partant des mesures connues du système, en général l'entrée et la sortie.

Le problème de retard avec lequel certaines mesures sont fournies est un problème qui survient par exemple lorsque l'on travaille avec des systèmes biotechnologiques et/ou chimiques. En effet, il n'existe pas de capteurs fiables capables de mesurer en ligne certaines variables biologiques (comme la biomasse, le substrat...). Par conséquent, les mesures sur ce type de systèmes sont faites normalement hors-ligne. Leur analyse est souvent complexe et nécessite des manipulations en laboratoire longues et délicates car ces mesures sont difficiles à automatiser, ce qui affecte le système de contrôle.

D'autre part, la connaissance fiable des variables d'état est un élément essentiel aussi bien pour le contrôle des procédés que pour la surveillance (détection et diagnostic de défaut, pannes dans le système) [Gatzke et Doyle III, 2002].

Ce contexte général de la problématique nous oriente vers le besoin d'avoir des méthodes d'estimation d'états fiables et simples. En effet, nous sommes confrontés à une grande gamme d'outils théoriques d'analyse et d'observation des systèmes non-linéaires dont le "cadre" mathématique provoque toujours une certaine réserve à leur utilisation.

Dans un premier temps, les observateurs ont été développés sous forme de calcul formel, avec le seul objectif de fournir à l'utilisateur des résultats garantis au sens mathématique, en l'absence d'outils informatiques performants, avec toutefois quelques contraintes. Voir [Besançon, 1996] pour un survol général de l'état de l'art.

Actuellement, compte tenu du développement des outils informatiques et des logiciels de calcul numérique, un pont a été créé entre les outils mathématiques et informatiques donnant lieu aux observateurs numériques (voir par exemple [Boillereaux et Flaus, 2000] ou [Boillereaux, 1996]). Il s'agit donc d'algorithmes plus simples à utiliser pour des non-spécialistes, mais en contrepartie ces algorithmes demandent en général beaucoup plus de ressources de calcul.

Par ailleurs, un système physique peut avoir plusieurs modèles différents en fonction de l'utilisation considérée. Par exemple, nous pouvons trouver des modèles soumis à différents niveaux de précision, des modèles dynamiques ou encore statiques. De plus, la nature des phénomènes peut aussi influencer sur le modèle (phénomènes continus ou à événements discrets, discontinuités, équations algèbro-différentielles, etc.) ou le nombre d'éléments composant l'installation [Thévenon, 2000].

Il est donc également souhaitable que l'observateur repose sur une conception capable de prendre en compte ces formalismes mathématiques, indépendamment du contexte d'utilisation du modèle (systèmes réels bruités avec mesures hors-ligne ou modèle décrit par transitions hybrides par exemple). Cependant, il faut noter que la plupart des observateurs existant utilisant le calcul formel, sont généralement intégrés à une application particulière. Ils ont donc été développés spécifiquement pour un procédé, ce que fait d'eux des outils difficiles à mettre en œuvre hors des systèmes pour lesquels ils ont été envisagés [Boillereaux, 1996], [Hammouri et Othman, 2001].

Tout ceci nous conduit à proposer que l'un de nos objectifs dans ce travail soit de contribuer à l'estimation d'états numériques des systèmes non-linéaires, en s'appuyant

sur des méthodes qui puissent être d'un formalisme général (autant que possible), tout en garantissant les résultats obtenus.

1.1 Le problème d'estimation classique

Dans le cadre d'une approche générale pour le problème d'observation, nous allons considérer la représentation discrète suivante (nous ne considérerons pas dans cette thèse le cas continu), pour la représentation de systèmes non-linéaires :

$$\Sigma : \begin{cases} x(k+1) = f(x(k), u(k)) \\ y(k) = h(x(k)) \end{cases} \quad (1.1)$$

avec $x \in \mathbb{R}^n$ le vecteur d'état, $y \in \mathbb{R}^p$ la sortie (pour nous toujours mesurée), $u \in \mathbb{R}^m$ la commande ou vecteur d'entrées, $f(\cdot)$ et $h(\cdot)$ sont des fonctions non-linéaires connues (généralement complexes). Finalement k représente le temps discret.

Remarque 1: *L'utilisation d'une représentation discrète n'est pas une restriction substantielle, parce que les mesures obtenues sur un système réel sont presque toujours obtenues par échantillonnage.*

Nous nous intéressons à la reconstruction (par l'observateur $O(\cdot)$ de $\Sigma(\cdot)$), de l'état courant du système $\Sigma(\cdot)$, c'est-à-dire le calcul de la trajectoire des variables d'état du vecteur d'état estimé x , en utilisant le modèle (1.1), les mesures de l'entrée et la sortie du système et la valeur initiale $x(0)$. Un tel système auxiliaire qui utilise les entrées et sorties de $\Sigma(\cdot)$, et a comme sorties les états estimés \hat{x} du système est appelé classiquement observateur.

$$O : \begin{cases} \dot{z}(k) = f(z(t), u(k), y(k)) \\ \hat{x}(k) = h(z(t), u(k), y(k)) \end{cases} \quad (1.2)$$

L'observateur $O(\cdot)$ est tel qu'il doit vérifier que l'état estimé converge vers l'état réel, c'est à dire : $\|\hat{x}(k) - x(k)\| \rightarrow 0$ quand $k \rightarrow \infty$. On souhaite que si $\hat{x}(0) = x(0)$, alors pour tout $k \geq 0$, $\hat{x}(k) = x(k)$.

Remarque 2 : *Il n'y a pas une méthode systématique au sens du calcul formel, ou une méthodologie ordinaire de synthèse d'observateurs, qui puisse garantir un bon fonctionnement pour des systèmes non-linéaires sous la forme générale (1.1). Il y a seulement un vaste bagage d'observateurs disponibles (parfois avec réglages très délicats) sous des formes particulières ou qui vérifient des conditions spécifiques de structure. [Othman, 1992], [Besançon, 1996], [Boillereaux, 1996].*

Une seule condition est nécessaire à $O(\cdot)$ pour pouvoir être mis en œuvre efficacement. Cette condition est que le système soit observable. La garantie de cette condition se révèle particulièrement difficile à vérifier vis-à-vis des systèmes non-linéaires, voire impossible de façon formelle et générale. Rappelons-nous que la notion classique d'observabilité nous dit que, pour pouvoir construire un observateur pour un système ou procédé dont le modèle est $\Sigma(\cdot)$, il est nécessaire que son état puisse être reconstruit à partir de ses entrées et de ses sorties. Cette notion formalisée initialement pour des systèmes non-linéaires continus par [Hermann et Krener, 1977], et pour des systèmes non-linéaires discrets par [Sontag, 1984] ou [Nijmeijer, 1982], peut être résumée par la notion d'indistinguabilité d'un couple d'état (dans le cas d'un système non-linéaire général, l'observabilité dépend donc de l'entrée appliquée). C'est à dire qu'un système est dit observable, si tout couple d'états du système peut être distingué grâce aux sorties correspondantes, pour au moins une entrée. Si pour toute condition initiale $x_1(0) \neq x_2(0)$ de $\Sigma(\cdot)$, il existe $k > 0$ et $u(k)$ sur $[0, k]$, tel que les sorties des 2 trajectoires soient pas identiquement égales, alors $\Sigma(\cdot)$ est dit observable.

D'ailleurs quand un système n'admet que des entrées permettant de distinguer tout couple d'états, on dit que cette entrée est universelle ou non singulière et l'observabilité est dite uniforme [Sussmann, 1979], [Gauthier et Bornard, 1981].

1.2 Brève revue bibliographique

Notre travail est centré sur les observateurs non-linéaires numériques sous forme de *capteurs logiciels* pour la maîtrise des procédés au travers de *mesures indirectes*.

Considérant la diversité des observateurs développés, et compte tenu de la Remarque 2, nous donnerons d'abord un panorama général des approches existantes.

Nous verrons les approches d'estimation ponctuelle, dont la solution est habituellement un vecteur réel en \mathbb{R}^n (tel que les méthodes exactes ou garanties, stochastiques ou déterministes, locales ou globales). Ce type d'approche reste limité, parce qu'il n'est généralement pas possible de caractériser d'une bonne façon les incertitudes avec lesquelles la solution au problème d'estimation a été obtenue. Il n'est donc pas possible d'accorder une validité certaine aux résultats [Jaulin, 2000]. Ainsi, les résultats techniques ne sont pas présentés en détail, mais pourront être trouvés dans les références citées. Les résultats liés de façon directe avec notre travail seront évoqués tout au long de ce document.

1.2.1 Observateurs linéaires

Initialement les systèmes abordés ont été les systèmes linéaires, dont les observateurs de Kalman et Luenberger ont donné de bons résultats. Parmi les différences entre ces deux approches, nous remarquons le choix fait pour la modélisation des perturbations, déterministe ou stochastique respectivement [Kalman et Bucy, 1961] [Luenberger, 1971]. Dans les caractéristiques communes à ces approches, nous pouvons remarquer que ces algorithmes sont récursifs (à chaque nouvelle mesure, ils calculent la nouvelle valeur des variables d'état en fonction de l'estimation précédente), que la stabilité et la convergence sont garanties (il y a normalement un rapprochement de la valeur exacte à chaque pas de calcul). Malgré les bons résultats obtenus, les systèmes linéaires ne modélisent qu'un groupe très limité des systèmes existants du milieu pratique industriel. Ainsi, des extensions à des domaines non-linéaires ont été nécessaires, et les travaux de recherche se sont focalisés sur cet aspect.

1.2.2 Observateurs non-linéaires

Dans ce contexte, quand les systèmes sont non-linéaires, l'observation d'état est un peu plus délicate et les approches envisageables sont soit une approximation des algorithmes linéaires, soit des algorithmes non-linéaires spécifiques. Dans le premier cas, l'approximation est basée sur une linéarisation du modèle autour d'un point de fonctionnement. Malheureusement la convergence de ce type de méthode n'est pas assurée, sauf sous quelques conditions particulières comme en [Reif *et al.*, 1999]. Ceci a donné lieu par exemple, à l'observateur de Luenberger étendu et au filtre de Kalman étendu [Song et Grizzle, 1995], [Grewal *et Andrews*, 1993], [Brown *et Hwang*, 1997], [Brookner, 1998]. Dans le cas d'algorithmes non-linéaires spécifiques nous pouvons retrouver par exemple des approches tel que l'observateur adaptatif de [Bastin et Dochain, 1990] ou d'autres observateurs spécifiques tel que [Gauthier *et al.*, 1992], ou [Alvarez, 2000]. Ces observateurs ne servent qu'à la classe de systèmes pour lesquels ils ont été conçus, mais leur convergence a bien été démontrée. La synthèse de ces types d'observateurs est normalement complexe, car leur développement est mathématiquement complexe et difficile d'accès pour les non-spécialistes. De plus, ce type d'approches suppose que le modèle est parfaitement connu, hypothèse peu réaliste dans un contexte industriel, mais qui est une bonne approche théorique.

Tout ceci nous conduit à proposer qu'un autre de nos objectifs soit de contribuer à la représentation d'incertitudes spécifiques d'un modèle non-linéaire notamment pour des cas réels, au moment de faire l'estimation de l'état.

En résumé, pour résoudre le problème d'estimation non-linéaire, nous pouvons trouver dans la littérature parmi autres, les approches suivantes :

1.2.2.1 Observateurs étendus

Comme nous l'avons remarqué, il est possible d'étendre quelques techniques linéaires à des systèmes non-linéaires, ceci en calculant les gains des estimateurs à partir du modèle du procédé linéarisé tangent autour d'un point de fonctionnement. C'est par exemple le cas du filtre de Kalman étendu [Song et Grizzle, 1995], [Grewal *et Andrews*, 1993], [Brown *et Hwang*, 1997], [Brookner, 1998]. Malheureusement les preuves de stabilité et de convergence établies dans le cas de systèmes linéaires, ne peuvent pas être étendues de manière générale à ces systèmes. Malgré cela, ce filtre étendu a été appliqué avec succès sur différents types de procédés non-linéaires, par exemple [Sargantanis et Nazmul-Karim, 1994]. Un autre cas est l'observateur de

Luenberger étendu. Ici un modèle linéarisé est aussi nécessaire, et le gain est calculé par placement de pôles. Ce type d'observateur ne peut être utilisé que lorsque l'on est sûr que l'état restera au voisinage de l'état d'équilibre, c'est-à-dire, dans la zone "considérée comme linéaire" : zone pour laquelle le placement de pôles a été effectué. De ce fait, cette méthode n'est pas très utilisée, parce que son utilisation peut être compromise par les instabilités qui peuvent se révéler si l'on s'éloigne du point de fonctionnement. Aucune méthodologie de synthèse garantissant un bon fonctionnement n'est disponible pour ces estimateurs étendus, ce qui rend leur réglage parfois très délicat.

1.2.2.2 Observateurs adaptatifs

Pour des systèmes comme les bioprocédés par exemple, il est souvent difficile d'élaborer un modèle fiable. Cette contrainte a conduit à développer des estimateurs qui puissent estimer les paramètres du modèle en même temps que l'état [Chamilothoris, 1987]. Nous pouvons distinguer par exemple le filtre de Kalman adaptatif, pour lequel aucune preuve de stabilité n'est fournie. Il y a aussi des observateurs dont les applications sont plus restrictives mais dont la convergence a été démontrée, tel que l'observateur adaptatif de Bastin-Dochain ou la méthode RPE par exemple.

La caractéristique principale du filtre de Kalman adaptatif est que le vecteur d'état est augmenté du nombre de paramètres à estimer. Les paramètres cinétiques sont ainsi considérés comme des états supplémentaires. Dans le cas des bioprocédés, les paramètres (p) sont associés à l'équation différentielle $\frac{dp}{dt} = 0$ (une supposition de variations lentes dans le temps). Le problème de cette approche est l'absence de garantie de convergence et de stabilité, ainsi que la difficulté associée au réglage de leurs paramètres. Cette approche a toutefois été mise en œuvre sur plusieurs procédés, voir par exemple [Dimitratos *et al*, 1991].

La méthode RPE (Recursive Prediction Error), est classée parmi les techniques adaptatives dont la convergence est démontrée, dans le cas particulier où le système est linéaire par rapport à l'état. [Ljung et Soderström, 1983] et [Ljung, 1979] ont mis au point cette méthode comme une amélioration du filtre de Kalman adaptatif. Elle permet en même temps d'estimer l'état et les paramètres d'un procédé en découpant le problème en deux parties. L'état est estimé par un filtre de Kalman et l'estimation des paramètres est réalisée de manière récursive en minimisant une fonction

quadratique de l'erreur de prédiction, par la méthode du gradient. Des applications de cette méthode ont été réalisées notamment par [Charbonnier, 1994]. Des variantes de la méthode RPE utilisée conjointement avec d'autres méthodes d'optimisation, telles que l'heuristique Recuit Simulé, ont été testées pour surmonter les problèmes de minimum locaux par [Valdés-González *et al.*, 2000b].

Un autre exemple est l'observateur de Bastin et Dochain [Bastin et Dochain, 1990], qui proposent un observateur d'état et de paramètres cinétiques appliqués à des bioprocédés, pour les cas où l'on dispose uniquement d'une mesure partielle de l'état. Des conditions à vérifier sur les gains d'estimations K_x et K_p permettent d'assurer la stabilité et la convergence de l'estimation. L'état manquant peut ensuite être calculé par bilan de matière [Dochain, 1986].

1.2.2.3 Observateurs à grand gain

Ce type d'observateurs est relativement classique en observation de systèmes non-linéaires. Son nom est dû au fait que le gain d'observation choisi est suffisamment grand pour compenser les non-linéarités du système. Sous certaines hypothèses, la convergence exponentielle de cet observateur a été établie et appliquée à des procédés biotechnologiques dans le travail de [Gauthier *et al.*, 1992], [Farza *et al.*, 1998].

1.2.2.4 Approches basées sur l'optimisation

L'observateur non-linéaire a été aussi présenté sous forme d'un problème d'optimisation, c'est-à-dire, sous forme des solutions numériques obtenues par la minimisation d'un critère d'erreur [Zimmer, 1993], [Zimmer, 1994]. La difficulté majeure de cette approche est liée directement au problème d'optimisation non-linéaire associé. Le travail de [Moraal et Grizzle, 1995] présente un observateur pour des systèmes discrets, dont une application de la méthode d'optimisation de Newton a été incluse pour résoudre un système d'équations non-linéaires. A partir de ce type d'observateurs, plusieurs travaux se sont développés, essentiellement les approches par *horizon glissant* ou *MHSE*. Nous pouvons, parmi d'autres travaux retrouver ceux de [Michalska et Mayne, 1995], [Muske et Edgar, 1997], [Alamir, 1999], [Boillereaux et Flaus, 2000] et [Rao *et al.*, 2000], ainsi qu'un survol de l'état de l'art dans [Allgöwer *et al.*, 1999]. Cette technique consiste à minimiser l'écart entre la mesure et sa prédiction sur un horizon de temps glissant. Elle est particulièrement fiable, et

applicable sur une grande variété de systèmes différents, tel que des systèmes à événement discret [Rao *et al.*, 2000], des systèmes continus [Boillereaux, 1996], décrit par modèles hybrides [Ferrari-Trecate *et al.*, 2000] et [Valdés-González *et al.*, 2001b], ou modélisés par des réseaux de neurones [Flaus et Boillereaux, 1995], ou encore à de multiples modèles et à la détection de défaut [Gatzke et Doyle III, 2002] etc. Rappelons que la difficulté de ce type d'approche est de garantir que le minimum local trouvé soit bien global. Pour essayer de surmonter ce problème, la méthode heuristique Recuit Simulé a été appliquée cette fois-ci à la technique d'estimation MHSE, présentée en [Valdés-González *et al.*, 2000a]. Cependant la convergence au sens probabiliste de cette méthode d'optimisation, et l'impossibilité de trouver plus d'un minimum global à la fois (lorsque c'est le cas), font qu'une partie du problème persiste.

Malgré les problèmes de minimums locaux, l'approche à horizon fuyant est applicable sur une grande variété de systèmes différents. Ainsi l'observateur MHSE est intéressant car il est indépendant de toute structure du système observé [Alamir, 2001]. Nous présentons dans ce travail les algorithmes numériques, la théorie et des détails additionnels sur cette méthode d'estimation à *horizon fuyant* en y consacrant un chapitre.

Comme nous l'avons vu, les approches ponctuelles restent limitées par le fait qu'il n'est pas toujours possible de caractériser les incertitudes liées à la solution qui a été obtenue (hors des minimums locaux), et donc il n'est pas toujours possible de valider ces résultats. Cependant il existe une approche garantie, qui nous permet de trouver un ensemble, solution au problème d'estimation. Il s'agit de l'approche d'estimation ensembliste où la solution est contenue de façon certaine dans l'ensemble trouvé, et dont les incertitudes propres peuvent être caractérisées aussi à coup sûr [Jaulin, 2000].

1.2.2.5 Approches ensemblistes

Cette approche a ses bases théoriques dans le calcul ensembliste ou calcul par intervalle, dont notamment l'arithmétique par intervalles. La principale caractéristique de ce type d'analyse demeure dans sa capacité à appréhender les incertitudes sur les paramètres d'un modèle. La question était de savoir comment les incertitudes évoluent et comment quantifier l'erreur finale. [Moore, 1966], [Moore, 1979], [Neumaier, 1990] et [Hansen, 1992] sont les principaux contributeurs à la théorie

ensembliste. Ils ont mis en place les fondements de ces outils, en passant par l'analyse par intervalles, la résolution de systèmes d'équations et l'optimisation globale.

Il s'agit donc d'une théorie qui permet de pallier aux incertitudes de certaines données intervenant dans le problème ponctuel, et aux erreurs de calcul numérique introduites [Jaulin, 2000]. Dans ces imprécisions sont incluses aussi celles qui proviennent des outils informatiques (calcul approximatif), des données issues des capteurs (liées à la qualité et précision de l'instrumentation disponible) et des incertitudes des paramètres d'un modèle (liées au fait qu'un modèle est souvent une approximation grossière de la réalité). Nous attendons donc des résultats garantis. Ceci parce que l'ensemble solution du problème contiendra toujours la ou les solutions exactes du problème associé [Hansen, 1992].

L'apparition d'approches ensemblistes survient aussi dans le domaine de l'estimation paramétrique ensembliste. Dans une formulation ensembliste du problème, nous ne cherchons pas une solution ponctuelle, mais un *ensemble solution* pouvant contenir une infinité de vecteurs. C'est-à-dire, l'estimation d'un ensemble de valeurs acceptables de paramètres cohérents avec les mesures et le modèle d'un système [Milanese et Vicino, 1996].

Ce type d'estimation, avec une technique de sous pavages, a été vu comme un problème d'inversion ensembliste [Jaulin *et al.*, 1993]. Une recherche vaste et approfondie a été faite sur le sujet, ce qui a donné suite à [Jaulin, 2000], [Jaulin *et al.*, 2001]. Dans le cadre des observateurs ensemblistes, nous retrouvons par exemple pour une représentation discrète que la technique du filtre de Kalman classique a été étendue à l'analyse par intervalle (IKF). La version IKF a la même optimalité et la même structure récursive que la version classique [Chen *et al.*, 1997]. Un autre estimateur non-linéaire récursif d'état, a été présenté dans [Kieffer *et al.*, 1998]. Ici, pour un temps donné, l'estimateur d'état retourne un ensemble garanti qui contient toutes les valeurs de l'état. Ces valeurs explicitent l'information disponible de la mesure sur le système. D'autres types d'observateurs par intervalles peuvent être trouvés dans l'application d'une méthode bornée pour une installation de traitement des eaux usées [Hadj-Sadok et Gouzé, 1999], [Hadj-Sadok et Gouzé, 2001]. Par exemple [Hadj-Sadok et Gouzé, 2001] présente un observateur par intervalles appliqué à un modèle avec quatre variables d'état dont quelques-unes sont incertaines ou inconnues. Pour le problème d'estimation posé, seules sont connues les limites supérieures et inférieures des variables. Sous ces conditions (incertitude des entrées ou des paramètres) un observateur par intervalles a été proposé pour faire l'estimation

garantie de l'état. L'incertitude correspondant à l'estimation est caractérisée par la largeur des bornes obtenues. Cette largeur dépend directement de l'incertitude des variables inconnues. De plus, l'étude montre qu'il est possible dans quelques cas d'estimer exactement les variables inconnues du modèle. L'observateur par intervalles converge vers les intervalles bornés qui contiennent avec certitude les états estimés.

Les deux dernières approches d'estimation, *Approches basées sur l'optimisation et Approches ensemblistes*, semblent utiles pour atteindre les objectifs que nous nous sommes fixés d'une manière informelle jusqu'à présent. D'abord parce qu'une approche garantie ensembliste nous permet d'aborder d'une manière numérique des problèmes traités communément d'une manière analytique en trouvant un ensemble qui contienne la ou les solutions du problème. De plus, si nous nous concentrons maintenant sur les aspects numériques, les approches basées sur l'optimisation ont l'avantage d'être applicables sur une grande variété de systèmes car l'approche à horizon fuyant se révèle indépendante des structures du modèle observé.

L'utilisation conjointe de ces deux approches sera adoptée dans notre travail. En effet dans les problèmes d'estimation, trouver les solutions de manière certaine est extrêmement important. Pour accomplir ce but, nous allons travailler avec une méthode d'estimation d'état non-linéaire à horizon fuyant basée sur une méthode d'optimisation globale par intervalles, appelée ici IMHSE (Interval Moving Horizon State Estimation).

L'idée principale est de manipuler des pavages sur un domaine admissible (intervalle vectoriel), pour trouver un ensemble de tous les sous-pavages (états intervalles) qui soit la solution globale du problème d'estimation. Cet ensemble permettra d'interpréter l'information disponible de la mesure effectuée sur le système.

La version ensembliste de l'observateur à horizon fuyant va nous permettre de contribuer à l'amélioration de l'estimation d'état numérique sur des systèmes non-linéaires, dont la convergence globale est garantie. La simplicité de cette approche va nous conduire à proposer un algorithme ensembliste capable d'approcher convenablement un ensemble solution du problème d'estimation non-linéaire, ceci tout en étant indépendant de la structure du modèle observé. Nous nous intéresserons à titre d'application à la résolution de quelques problèmes d'estimation non-linéaires. Parmi ceux-ci, quelques procédés théoriques pour valider notre approche, et notamment deux bioprocédés issus de problèmes réels dans le domaine de la fermentation sur substrat solide.

1.3 Contributions de la thèse

Il est clair que la synthèse d'un observateur repose sur une théorie bien éprouvée lorsque les modèles impliqués sont linéaires. Cependant, pour les systèmes non-linéaires, des solutions ne peuvent être trouvées que pour des cas particuliers. D'une façon générale, la plupart des méthodes d'estimation non-linéaires sont intéressantes sur le principe, mais leur synthèse est très délicate voire impossible si on veut garantir la convergence de l'estimation. Ceci reste un problème ouvert pour la recherche en automatique. De plus ces observateurs reposent sur des méthodes mathématiques très spécialisées, d'accès difficile aux non-spécialistes.

Une méthode originale d'estimation d'état à horizon fuyant ou MHSE pour le sigle anglais (Moving Horizon State Estimation) a été développée au sein de l'équipe CAPA du LAG (Laboratoire d'Automatique de Grenoble) pour essayer de surmonter ces difficultés à la fois théoriques et de mise en œuvre pratique, pour des procédés le plus souvent non-linéaires [Boillereaux, 1996] et [Boillereaux et Flaus, 2000]. Cependant cette méthode d'estimation présentait encore une limitation, celle de ne pas être en mesure d'assurer la convergence globale.

Dans cette thèse, nous présentons un algorithme d'estimation ensembliste à partir de mesures indirectes, appelé méthode d'estimation à horizon fuyant par intervalles, dont la convergence globale est assurée. Nous l'appellerons dorénavant IMHSE [Valdés-González *et al.*, 2002b].

1.3.1 Objectif général

Dans cette thèse, nous présentons une méthodologie numérique d'estimation non-linéaire d'état basée sur une approche ensembliste d'optimisation globale. Cette approche résout le problème d'estimation de l'état d'un système dynamique, par un problème statique d'optimisation globale non-linéaire par intervalles. Cette approche permet d'utiliser un modèle d'un procédé non-linéaire sous forme générale.

Notre but est de contribuer au développement d'un *algorithme d'estimation d'état globalement convergent*, capable de caractériser un ensemble d'état acceptable, à partir de mesures indirectes. Par ailleurs, cette approche ensembliste nous permettra de prendre en compte d'éventuels problèmes d'observabilité local, de plus nous aurons une indication sur l'observabilité pas à pas du système observé.

Un de nos objectifs sera aussi d'étudier les performances et les limites de ce type d'approche en termes de temps de calcul et de complexité informatique des algorithmes.

Les principaux axes de recherche de notre travail se regroupent autour des thèmes détaillés ci-après.

1.3.2 Objectifs spécifiques

- Du point de vue de l'observateur ensembliste
 - Mise au point du couplage d'un observateur robuste type MHSE avec une technique globale d'optimisation non-linéaire par intervalles.
 - Capacité de recalage en ligne à partir de mesures hors-ligne.
 - Fournir des indications en ligne à propos de l'observabilité du système, de façon à utiliser les estimations avec prudence.
- Du point de vue du modèle
 - Que la méthode IMHSE soit capable de travailler avec des modèles issus de représentations différentes.
 - Pouvoir utiliser (si nécessaire) plusieurs modèles d'un même procédé.
- Du point de vue du capteur logiciel
 - Simple à mettre en œuvre, tout en nécessitant un minimum de réglages.
- Du point de vue de l'expérimentation
 - Nous cherchons la performance de l'estimation dans des procédés réels biotechnologiques.

1.4 Organisation et présentation du travail réalisé

Avec l'approche IMHSE, nous développons dans ce travail ce qui est appelé couramment un *capteur logiciel*. Cet algorithme associé à un capteur physique (voire plusieurs) qui mesure la sortie, délivre une estimation par intervalles fiable de l'état du système, que l'algorithme reconstruit en temps réel. Pour reconstruire cet état, l'algorithme utilise la connaissance a priori donnée par un modèle, les mesures de l'entrée et de la sortie du procédé, ainsi que les mesures hors ligne disponibles des variables états.

Nous présentons au *chapitre 2* l'observateur numérique ensembliste IMHSE. Tout d'abord nous présentons la notion d'analyse par intervalles, indispensable au

développement de nos algorithmes ensemblistes. Ensuite nous présentons la définition de notre observateur ensembliste. Une fois notre observateur défini, nous développons le principe de la méthode IMHSE, la théorie sous-jacente, les algorithmes ensemblistes utilisés, les types de modèles avec lesquels nous travaillons et le type d'information prise en compte pour notre observateur dans le cas d'un procédé non-linéaire. Cette étude va permettre d'identifier les différentes étapes algorithmiques sur lesquelles repose la méthode d'estimation proposée. Une fois que les algorithmes de base liés à IMHSE seront établis, nous verrons la nécessité d'une technique globale ensembliste d'optimisation non-linéaire pour cette approche.

Le *chapitre 3* sera consacré à la technique d'optimisation globale qui est utilisée au cœur de notre observateur. Cette méthode permet d'aborder le problème d'optimisation non-linéaire avec ou sans contraintes. L'intérêt de cette méthode est sa capacité à trouver sur un domaine admissible, tous les ensembles d'états qui minimisent globalement un critère donné. Cet algorithme est basé sur l'arithmétique des intervalles. Nous examinerons tout particulièrement :

- l'algorithme de la technique d'optimisation utilisée
- le test de sélection
- la complexité informatique impliquée
- un outil développé en Java couplé au logiciel Matlab, pour obtenir de façon automatique et par intervalles; le gradient et le hessien d'un critère différentiable quelconque, sur un horizon de temps fini

Nous étudierons au *chapitre 4* l'observabilité numérique par intervalles. Ce formalisme est basé sur la notion d'indistinguabilité ensembliste. Nous illustrerons ensuite l'utilisation de ce formalisme pour des cas théoriques généraux, tout en donnant un indicateur pour IMHSE à propos de l'observabilité pas à pas du système estimé.

L'approche IMHSE et l'utilisation de plusieurs modèles est développée au *chapitre 5*. Avec cette approche nous verrons la détection de dysfonctionnements non prévus par le modèle interne de notre observateur. Ceci consiste à commuter le modèle interne de notre observateur, parmi une famille de modèles de dysfonctionnement prévus, lorsque celui-ci est incompatible avec l'évolution observée du système sur l'horizon choisi. Cette approche est nommée IMHSE-Multi-modèle.

Enfin, plusieurs exemples d'applications de la méthode d'estimation sont présentés au *chapitre 6* :

- un bioprocédé générique

- un bioprocédé réel modélisé de façon hybride. Il s'agit d'un procédé de fermentation sur substrat solide ou SSC (Solid Substrate Culture), dont nous disposons les résultats expérimentaux dans une base de données
- un bioprocédé réel issu d'un réacteur à échelle pilote (aussi avec des données expérimentales)
- un système non-linéaire théorique classique

Ce groupe d'exemples qui possèdent des caractéristiques différentes, va nous permettre d'étudier et de valider les capacités de notre approche dans l'estimation d'état numérique ensembliste.

Finalement nous dressons le bilan de ce travail de recherche en montrant comment la méthode IMHSE répond aux objectifs fixés au début de ce travail de recherche, et comment celle-ci répond aux besoins d'estimation d'état dans un contexte industriel.

Chapitre 2

IMHSE : Un observateur numérique ensembliste

De nombreuses théories modernes en rapport avec la commande des systèmes dynamiques sont basées sur la représentation d'état. Le comportement d'un système est décrit au travers des variations de son état, comme le système défini en (1.1). Comme nous l'avons déjà mentionné au chapitre précédent, le besoin d'un observateur d'état vient du fait que la commande de ce système nécessite la connaissance du vecteur d'état à chaque instant, mais aussi cette connaissance est nécessaire pour la surveillance du procédé ou pour sa conduite manuelle. Pour reconstruire cet état, nous pouvons parcourir théoriquement tout un espace de recherche, au sens de points réels, pour trouver la solution du problème. Cependant dans la réalité une approche ponctuelle ne permet d'analyser qu'une infime parcelle de cet espace. Il n'est alors pas possible de conclure avec certitude que la seule solution ou toutes les solutions du problème ont été retrouvées de façon garantie, particulièrement dans des cas de systèmes non-linéaires. Il faut remarquer que les méthodes ayant recours au calcul formel, arrivent aussi à obtenir des résultats garantis. Toutefois la classe des problèmes à traiter, relevant de ce type d'observateurs est très réduite. De plus, il existe certaines catégories de problèmes, pour lesquelles il est très difficile voire presque impossible, d'obtenir des résultats avec des méthodes formelles.

Pour essayer de surmonter cette difficulté, dans le cadre d'observateurs non-linéaires, nous nous intéresserons à un observateur présenté sous forme d'un problème d'optimisation globale, appelé IMHSE. Cette technique consiste à minimiser globalement l'écart entre la mesure et sa prédiction sur un horizon de temps glissant prédéfini et sur un domaine admissible pour le problème d'estimation. Ceci correspond à la minimisation de l'erreur de prédiction au sens de moindres carrés. Cette approche est particulièrement fiable et applicable sur une grande variété de systèmes différents, grâce à son indépendance vis-à-vis de la structure du système.

Dans ce chapitre nous définirons ce que nous entendons par problème d'observation ensembliste d'un système dynamique. Ensuite nous présenterons la nouvelle méthode d'estimation ensembliste IMHSE, pour montrer l'intérêt de notre approche.

2.1 Caractérisation des incertitudes

Considérons une variable quelconque. Nous voulons caractériser avec rigueur les incertitudes sur cette valeur. La variable peut être représentée par un nombre intervalle censé la contenir.

Définition 1. *Un intervalle $[x]$ est un ensemble borné fermé de nombres réels, telle que : $\forall a, b \in \mathbb{R}$*

$$[a, b] = \{x / a \leq x \leq b\}$$

Exemple 2.1 : Dans le cas d'une variable b quelconque, qui est fixé à $b = 0.5$ ($b \in \mathbb{R}$), et qui a désormais une variabilité de 50% ; un ensemble tenant compte de cette variabilité serait l'intervalle dénoté par $\mathbb{B} = [b] = [0.25 \ 0.75]$, où $\mathbb{B} \in \mathbb{R}$ représente un ensemble de nombres réels. Il faut remarquer que cette représentation nous dit tout simplement que $b \in \mathbb{B}$, et rien d'autre. C'est-à-dire une variable et sa variabilité peuvent être représentée par un seul ensemble.

Notons qu'en prenant un nombre contenu dans cet ensemble, nous avons une représentation ponctuelle d'une valeur, dont l'incertitude est complètement caractérisée par l'ensemble. Cependant cette représentation est aussi pauvre, parce

qu'il peut avoir des zones où b ne soit pas vraiment contenu, bien que la densité d'appartenance soit uniforme.

Exemple 2.2 : Supposons deux domaines réels $\mathbb{X} = [-1, 3]$ et $\mathbb{Y} = [5, 6]$. Soit une variable réelle z censée être contenue dans le domaine $\mathbb{Z}_1 = \mathbb{X} \cup \mathbb{Y}$. Pour simplifier supposons $\mathbb{Z}_2 = [-1, 6]$, tel que $\mathbb{Z}_1 \subset \mathbb{Z}_2$. Malgré le fait que $z \in \mathbb{Z}_2$, il y a une zone de \mathbb{Z}_2 où z ne peut pas être contenue. C'est la zone $]3, 5[$.

Il est donc possible d'affirmer que les représentations ensemblistes permettent une manipulation plus simple des variables aléatoires avec densité uniforme. Elles demandent moins de connaissances statistiques sur ces variables et engendrent des calculs plus simples et avec moins d'hypothèses que l'approche probabiliste. De plus, avec l'approche ensembliste, les manipulations non-linéaires sont plus simples, voir par exemple [Jaulin, 2000].

Un approche de modélisation semi-qualitative du système (1.1), que nous nommerons $\mathcal{I}\Sigma(\cdot)$, a été développée pour des bioprocédés dans [Romero-Jimenez, 1996]. Cette méthodologie nous permet de recueillir les différents types d'information disponibles, même si ses paramètres sont connus de façon incertaine. La structure du modèle est donnée par un modèle de connaissances, dont les paramètres et les variables peuvent être définis sous la forme d'intervalles. Considérons par exemple l'ensemble $[y](k)$ en (1.1), comme un vecteur intervalle de mesures acceptables \mathbb{Y} , centrées en y , $y \in \mathbb{Y}$. Ceci prend en compte le fait que l'erreur de la mesure d'une variable d'un procédé est fréquemment donnée en terme de la tolérance de l'appareil de mesure, dans un contexte expérimental. Ce qui fait de la représentation ensembliste une approche naturelle pour traiter les mesures fournies par le capteur.

Avant de continuer avec le problème de l'observateur ensembliste, faisons un survol de la théorie de l'analyse par intervalles, et en particulier des outils que nous allons utiliser tout au long de cette thèse.

2.1.1 Quelques aspects à propos de l'analyse par intervalles

2.1.1.1 Arithmétique par intervalles

L'extension des quatre opérations arithmétiques définies sur les nombres réels, à l'arithmétique d'intervalles est définie comme suit :

$$\begin{aligned}
[a, b] + [c, d] &= [a + c, b + d] \\
[a, b] - [c, d] &= [a - d, b - c] \\
[a, b] \times [c, d] &= [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \\
[a, b] \div [c, d] &= [a, b] \times [1/d, 1/c] \quad \text{si } 0 \notin [c, d]
\end{aligned}$$

Nous pouvons aussi avoir un *vecteur intervalle* $[x]$ (aussi appelé pavé), $x \in \mathbb{R}^n$, constitué de n intervalles réels tel que : si $[x] \subset \mathbb{I}\mathbb{R}^n$ représente un ensemble de nombres réels par intervalle, alors :

$$[x] = [[a_1, b_1], [a_2, b_2], \dots, [a_n, b_n]]^T = [[x_1], [x_2], \dots, [x_n]]^T$$

Plusieurs autres concepts ont été développés par Moore [Moore, 1966], [Moore, 1979] dont nous pouvons distinguer les caractéristiques suivantes. Ces caractéristiques se rapportent aux intervalles. Nous les utiliserons tout au long de ce mémoire. Le lecteur pourra avantageusement consulter les références mentionnées dans ce chapitre pour de plus amples détails :

- _ La *largeur ou taille d'un intervalle* est défini comme le plus grand de ses côtés, c'est-à-dire :

Si $[x]$ est un intervalle : $w([x]) = b - a$

Si $[x]$ est un vecteur intervalle :

$$w([x]) = \max \{w([x_1]), w([x_2]), \dots, w([x_n])\} \quad (2.1)$$

Notez que l'intervalle nul a une largeur égale à zéro.

- _ Le *centre ou milieu d'un intervalle* est défini par :

Si $[x]$ est un intervalle : $c([x]) = \frac{b+a}{2}$

Si $[x]$ est un vecteur intervalle, $c([x]) \in \mathbb{R}^n$:

$$c([x]) = [c([x_1]), c([x_2]), \dots, c([x_n])]^T \quad (2.2)$$

- _ La *valeur absolue* d'un intervalle est définie de la façon suivante :

Si $[x]$ est un intervalle : $|[x]| = \max \{ |a|, |b| \}$

Si $[x]$ est un vecteur intervalle :

$$|[x]| = \max \{ |[x_1]|, |[x_2]|, \dots, |[x_n]| \} \quad (2.3)$$

2.1.1.2 Quelques opérations logiques ensemblistes de base

Côté opérations logiques, nous pouvons aussi opérer sur des sous-ensembles de \mathbb{R}^n , en accord avec les définitions suivantes. Nous donnons celles qui seront utilisées tout au long de ce document :

$$\mathbb{A} \cap \mathbb{B} = \{[x] \subset \mathbb{R}^n / [x] \subset \mathbb{A} \wedge [x] \subset \mathbb{B}\} \quad (2.4)$$

$$\mathbb{A} \cup \mathbb{B} = \{[x] \subset \mathbb{R}^n / [x] \subset \mathbb{A} \vee [x] \subset \mathbb{B}\} \quad (2.5)$$

$$\mathbb{A} \subset \mathbb{B} \Leftrightarrow \{[x] \subset \mathbb{R}^n / \forall [x] \subset \mathbb{A} \Rightarrow [x] \subset \mathbb{B}\} \quad (2.6)$$

$$\mathbb{A} = \mathbb{B} \Leftrightarrow \{\forall [x] \subset \mathbb{A} / \mathbb{A} \subset \mathbb{B} \wedge \mathbb{B} \subset \mathbb{A}\} \quad (2.7)$$

Il faut aussi noter que les opérations arithmétiques de base sur les intervalles, telles que l'addition et la multiplication d'ensembles, ont des propriétés d'associativité et de commutativité. L'élément neutre de l'addition et de la multiplication sont aussi définis, ainsi que l'inclusion monotone et une relation d'ordre partiel. Pour avoir plus détails, exemples ou autres opérations sur les ensembles et test d'inclusion, voir [Moore, 1966], [Moore, 1979], [Romero-Jimenez, 1996], [Adrot, 2000] ou [Jaulin, 2000].

2.1.1.3 Fonctions à intervalles ou fonction d'inclusion

Quand nous travaillons avec des ensembles, nous devons remplacer les variables du modèle par des intervalles, tel que si $f: \mathbb{R} \rightarrow \mathbb{R}$ est une fonction réel, alors l'extension intervalle de $f(\cdot)$ dans les nombres réels par intervalles (\mathbb{IR}) peut être définie comme :

$$\begin{aligned} F: \mathbb{IR} &\rightarrow \mathbb{IR} \\ [x] &\rightarrow F([x]) \end{aligned}$$

où la fonction $F(\cdot)$ est telle que :

$$F([x]) : \{y = f(x) / x \in [x]\} \quad (2.8)$$

Dans le cas des fonctions élémentaires continues, l'extension intervalle $F(\cdot)$ correspond à l'image de $[x]$ par $f(\cdot)$, noté : $F([x]) = f([x])$.

D'ailleurs le calcul de la valeur d'une fonction intervalle peut se révéler délicat, principalement parce que l'arithmétique d'intervalles diffère de l'arithmétique réelle. La loi de distributivité, en particulier, n'est pas toujours vérifiée. Cependant, le calcul de l'image d'une fonction par intervalles est très intéressant parce que la propriété des résultats garantis repose sur ce calcul [Moore, 1966], [Moore, 1979].

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ une fonction quelconque, alors $F(.)$ est son extension intervalle, définie telle qu'en (2.8) comme une fonction inclusion de $f(.)$ si : $\forall [x] \subset \mathbb{I}\mathbb{R}^n$ alors $f(.) \subset F(.)$. Le fait que $f(.) \neq F(.)$ est dû au pessimisme provoqué par les occurrences multiples d'une variable intervalle dans l'expression d'une fonction, aussi appelé *problème de dépendance*. En effet, l'écriture d'une expression ensembliste oublie les relations de dépendance qui peuvent exister entre les variables représentées par les ensembles [Moore, 1979]. L'évaluation ensembliste est alors dite pessimiste, parce qu'en général $f(.) \subset F(.)$.

La méthode la plus simple pour trouver une fonction d'inclusion est l'obtention de la fonction d'inclusion naturelle. Celle-ci consiste à remplacer les variables réelles dans l'expression algébrique de $f(.)$, par ses variables intervalles correspondantes, et à faire l'évaluation de $f(.)$ en utilisant l'arithmétique d'intervalles (remplacement de $f(.)$ par son extension à intervalles $F(.)$). Cependant la plupart du temps cette fonction n'est pas minimale.

Dans ce contexte, il est nécessaire de remarquer qu'il n'existe pas une fonction d'inclusion unique pour une fonction réelle. L'évaluation de n'importe quelle fonction d'inclusion d'une fonction réelle $f(.)$ conduit à un majorant de l'intervalle minimal [Moore, 1979]. De plus la longueur des intervalles obtenus dépend de l'expression initiale de la fonction $f(.)$, comme cela est montré sur des exemples illustratifs dans la thèse de [Romero-Jimenez, 1996].

La fonction $F(.)$ qui donne comme résultat l'intervalle le plus étroit possible est appelé une extension optimale. D'autre part, une fonction d'inclusion $F(.)$ est dite convergente si pour toute suite $[x](k)$, nous trouvons que :

$$\lim_{k \rightarrow \infty} w([x](k)) = 0 \Rightarrow \lim_{k \rightarrow \infty} w(F([x](k))) = 0 \quad (2.9)$$

Autrement dit, plus la largeur de l'intervalle $[x]$ est petit, plus l'évaluation de la fonction d'inclusion sera proche de l'image optimale (intervalle minimal).

Un résultat intéressant décrit en [Moore, 1979], est le suivant : une fonction d'inclusion $F(.)$ de $f(.)$, dans laquelle la variable intervalle n'apparaît qu'une seule fois et à la puissance un, permet de déterminer l'intervalle minimal contenant les valeurs de $f(.)$. Autrement dit $F(.)$ est optimale (évaluation sans pessimisme).

Plusieurs travaux ont déjà été menés avec l'objectif de trouver des fonctions d'inclusions efficaces (proches de l'optimal) pour différents types de fonctions $f(.)$. Voir par exemple [Moore, 1979], et une revision de l'état de l'art sur différentes techniques en [Jaulin, 2000] et [Adrot, 2000].

2.2 Le problème de l'observateur ensembliste

Considérons un système tel que (1.1), où la sortie est bornée (tolérance du capteur). Nous cherchons à caractériser un ensemble d'état \mathbb{X} dans un domaine admissible, où $y = \Sigma(\hat{x}, u)$. C'est-à-dire, l'ensemble \mathbb{X} qui puisse contenir tous les états \hat{x} sur un domaine admissible de façon garantie, et tel que la trajectoire issue de cet ensemble soit contenue dans le vecteur de mesures borné $[y]$.

Il s'agit ici d'un problème d'estimation ensembliste. Contrairement aux techniques ponctuelles classiques, la caractéristique principale de cette approche est qu'elle permet d'approcher la solution du problème d'estimation par un ensemble qui la contient à coup sûr. Posé de cette façon, le problème reste théorique, car on ne sait pas calculer effectivement avec les ensembles. Ainsi, déterminer l'ensemble exact de tous les états que donnent une trajectoire à l'intérieur de la mesure bornée de la sortie, n'est pas une tâche évidente à réaliser. Ceci est principalement dû à la surévaluation du résultat dans le calcul ensembliste. Nous allons maintenant tenter d'explorer trois possibilités du problème d'estimation par intervalles. Ci-dessous, \mathbb{S} représente l'ensemble de solutions indiscernables du problème d'estimation ensembliste, \mathbb{S}^* représente l'ensemble solution minimal tel que $\mathbb{S}^* \subset \mathbb{S}$. Toutes les trajectoires issues de l'ensemble \mathbb{S}^* restent à l'intérieur de la mesure bornée $[y]$. Finalement nous considérerons l'état intervalle $[x]$, un pavé de l'espace de recherche :

- a) Si un pavé $[x] \not\subset \mathbb{S}$: $[x]$ n'est pas contenu dans l'ensemble \mathbb{S} de solutions du problème d'estimation, c'est-à-dire, $[x] \cap \mathbb{S} = \emptyset$. Nous pouvons l'exclure.
- b) Si un pavé $[x] \subset \mathbb{S}$, cependant $[x] \not\subset \mathbb{S}^*$: $[x]$ appartient à l'ensemble \mathbb{S} de solutions indiscernables du problème d'estimation, mais que $[x]$ n'est pas une solution du problème d'estimation, nous sommes obligés de garder $[x]$.

- c) Si un pavé $[x] \subset \mathbb{S}^*$, $[x]$ appartient à l'ensemble de solutions minimal, alors $[x]$ est solution du problème d'estimation. Nous gardons le pavé. $[x]$.

Les trois opérations qui forment la base des algorithmes ensemblistes sont :

- 1) Trouver un ensemble qui contienne l'image de $[x]$ par une fonction d'inclusion $F(\cdot)$, tel que $\{F(x) / \forall x \in [x], F(x) \in F([x])\}$.
- 2) Montrer l'appartenance d'un pavé $[x]$ à un ensemble \mathbb{A} , c'est-à-dire que $[x] \subset \mathbb{A}$ ou $[x] \cap \mathbb{A} \neq \emptyset$. Cette opération est appelée test d'inclusion.
- 3) Réduire $[x]$ par rapport à \mathbb{A} . L'idée est de retrouver un pavé $[z]$, tel que $[z] \subset [x]$ et que $[x] \cap \mathbb{A} = [z] \cap \mathbb{A}$.

Supposons un problème d'estimation sur un système observable (1.1), tel que :

- _ Nous avons un domaine admissible \mathbb{X} pour les état du système (1.1).
- _ Après d'avoir appliqué des techniques de réduction et d'inclusion, nous avons trois zones d'états initiales, $\mathbb{S}_1, \mathbb{S}_2$ et \mathbb{S}_3 , où $\mathbb{S}_1 \subset \mathbb{S}_2$.
- _ Ces zones nous donnent des images bornées $\mathbb{Y}_1, \mathbb{Y}_2$ et \mathbb{Y}_3 , tel que $\mathbb{Y}_1 \subset \mathbb{Y}_2$, $\mathbb{Y}_1 \cap \mathbb{Y}_3 = \emptyset$ et $\mathbb{Y}_2 \cap \mathbb{Y}_3 = \emptyset$.
- _ Les bornes qui correspondent à la mesure de la sortie sont respectées seulement par l'ensemble \mathbb{Y}_1 , c'est à dire, $Sup(\mathbb{Y}_1) < \varepsilon_1$ où ε_1 représente la tolérance du capteur.
- _ Enfin, \mathbb{Y}_1 peut être reconstruite seulement à partir de la zone \mathbb{S}_1 , c'est-à-dire que la zone \mathbb{S}_1 est l'ensemble solution minimal du problème d'estimation tel que si $y_1 = \Sigma(x, u)$, $\forall x \in \mathbb{S}_1$ alors $y_1 \in \mathbb{Y}_1$.

Le problème d'estimation ensembliste est illustré dans la figure 2.1. Nous pouvons exclure l'ensemble \mathbb{S}_3 parce qu'on ne peut pas reconstruire \mathbb{Y}_1 à partir de \mathbb{S}_3 . L'image de l'ensemble \mathbb{S}_2 ne respecte pas la tolérance du capteur, cependant cet ensemble reste indiscernable de \mathbb{S}_1 . Ceci est principalement dû, comme nous l'avons déjà mentionné auparavant, à la surévaluation du résultat dans le calcul ensembliste.

La question qui se pose après avoir appliqué des techniques de réduction et des tests d'inclusion est : "*Qu'est-ce qui empêche le sous-ensemble \mathbb{S}_2 d'être réduit à \mathbb{S}_1 ?*" La réponse nous pouvons la trouver dans le pessimisme au problème de dépendance qui est née du calcul ensembliste et de la fonction d'inclusion utilisée. C'est à dire, plus la surévaluation d'un ensemble est grande, plus grand est l'effort nécessaire pour

distinguer les états. De plus, bien souvent cette fonction n'est pas minimale. Ainsi la longueur des intervalles obtenus dépend de la qualité de cette fonction.

Notez qu'il y a deux cas possibles : D'un part un système peut être observable, mais ces états seront quand même difficiles à distinguer. D'une autre part les états d'un système qui n'est pas observable ne peuvent pas être distingués (voir chapitre 4).

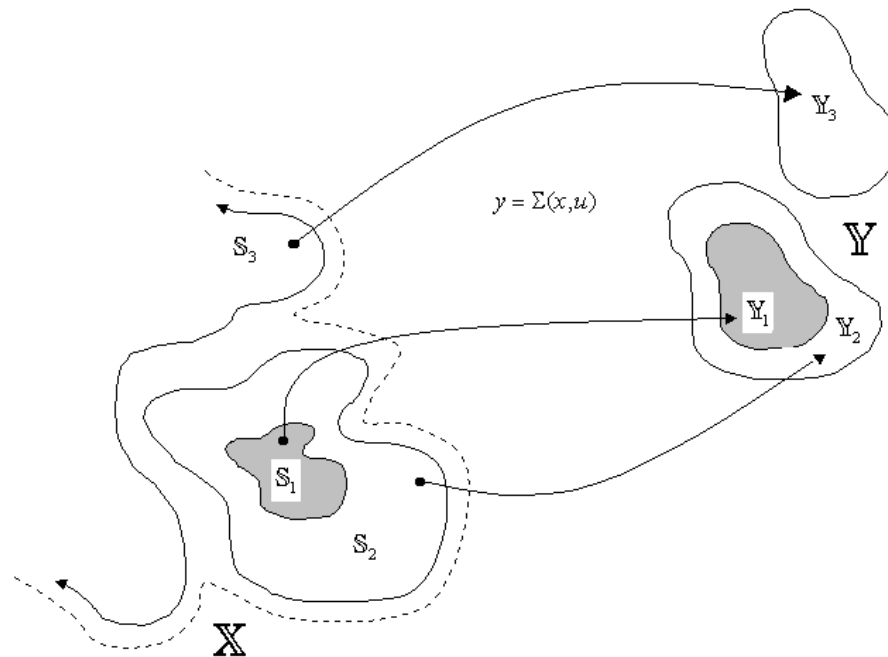


Figure 2.1 : Problème d'estimation ensembliste

Regardons l'exemple suivant :

Exemple 2.3 : Prenons la fonction : $f(x) = x^2 - x$ et sa fonction d'inclusion naturelle $F([x]) = [x]^2 - [x]$ comme un exemple numérique lié au problème d'estimation ensembliste. Supposons aussi que nous avons trois états liés à trois zones d'un domaine admissible, tel que :

- _ $[x_1] = [0, 1]$, tel que $[x_1] \subseteq \mathcal{S}_1$.
- _ $[x_2] = [1, 2]$, tel que $[x_2] \subseteq \mathcal{S}_2$ et $[x_2] \cap \mathcal{S}_1 = \emptyset$.
- _ $[x_3] = [3, 4]$, tel que $[x_3] \subseteq \mathcal{S}_3$.

Supposons aussi que le problème d'estimation d'état soit défini comme une approche basée sur l'optimisation, dont nous cherchons à trouver les états qui minimisent la fonction $f(\cdot)$ sur un domaine admissible $\mathbb{X} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3$, et que la précision du capteur est tel que $\mathbb{Y}_1 = [-1, 1]$.

Si nous adoptons une procédure d'élimination dans la recherche de solutions, telle que si la limite inférieure de l'image d'un état intervalle $[a]$ est plus grande que la limite supérieure de l'image d'un état intervalle $[b]$, alors il est clair que la solution (minimum) ne se trouve pas dans le premier : nous pouvons éliminer l'état intervalle $[a]$. Sinon nous ne pouvons rien affirmer (au moins avec cette fonction d'inclusion et en absence d'autres tests).

Dans notre exemple, $F([x_1]) = [-1, 1]$, $F([x_2]) = [-1, 3]$ et $F([x_3]) = [5, 13]$. Alors de toute évidence $[x_3]$ peut être exclue, mais $[x_1]$ et $[x_2]$ restent indistinguables. Notons que $[x_1]$ permet de reconstruire complètement $\mathbb{Y}_1 = [-1, 1]$. Toutefois nous ne pouvons rien dire sur $[x_2]$, parce que son image est contenue dans \mathbb{Y}_1 . Finalement la solution à ce problème d'estimation ensembliste est donné pour l'ensemble $[\hat{x}] \subseteq [x_1] \cup [x_2]$.

2.2.1 Définition d'un observateur ensembliste

De façon générale nous nous intéressons à la reconstruction par intervalles de la variable de l'état courant du système $\Sigma(\cdot)$, c'est-à-dire, des états estimés \hat{x} , tels que la trajectoire issue de ces états reste incluse dans les bornes $[y]$ données par un capteur qui mesure la sortie du système. Pour cet observateur, nous utilisons un modèle de procédé du type (1.1), les mesures de l'entrée et de la sortie du système (notons que dans le cas de mesures précises à 100%, un intervalle a une largeur nulle), et un domaine admissible \mathbb{X} pour les états \hat{x} . Notre observateur ensembliste est alors défini par :

Définition 2. *Observateur ensembliste : Il s'agit d'un système auxiliaire capable de déterminer sur un domaine admissible \mathbb{X} l'ensemble d'états estimés $[\hat{x}]$, tel que les trajectoires $[\hat{y}]$ issues de $[\hat{x}]$ restent à l'intérieur de la mesure bornée de la sortie $[y]$ (tolérance du capteur). Les caractéristiques propres de notre observateur sont :*

- *Nous ne cherchons pas à minimiser infiniment l'écart sur les trajectoires de sortie, mais à rester dans les bornes de tolérance données, c'est-à-dire que nous cherchons que les états estimés soient tels que : $[\hat{x}] : \{[\hat{x}] / \forall [\hat{x}_i] \subset \mathbb{X}, [\hat{y}_i] \subseteq [y]\}$*
- *Les trois cas qui sont possibles en considérant le système (1.1) sont :*
 - *$[x_1] \subset \mathbb{X}$ et $[y_1] = h([x_1])$ tel que $[y_1] \subseteq [y]$, alors $[x_1]$ est solution du problème d'estimation, $[x_1] \subset [\hat{x}]$.*

- $[x_2] \subset \mathbb{X}$ et $[y_2] = h([x_2])$ tel que $[y_2] \not\subset [y]$, mais $[y_2] \cap [y] \neq \emptyset$ donc nous ne pouvons rien dire par rapport à $[x_2]$ et donc $[x_2] \subset [\hat{x}]$. Cependant comme il existe certains états de $[x_2]$ qui sont solutions du problème, $[x_2]$ pourrait être encore réduit.
- $[x_3] \subset \mathbb{X}$ et $[y_3] = h([x_3])$ tel que $[y_3] \not\subset [y]$ et $[y_3] \cap [y] = \emptyset$, pourtant $[x_3]$ ne peut pas faire partie des solutions et nous pouvons l'éliminer sans autres considérations.

Remarque 3: Si la largeur de la mesure de sortie bornée tend vers zéro, c'est-à-dire si $w([y]) \rightarrow 0$ (Capteur précis à 100%), alors l'estimation ensembliste converge vers un cas d'estimation d'état réel classique.

Remarque 4: Notons qu'en prenant un vecteur ponctuel à l'intérieur de l'ensemble solution $[\hat{x}]$, et avec le système (1.1) nous aurons une estimation ponctuelle de l'état, et l'incertitude sur cette estimation restera complètement caractérisée par l'ensemble $[\hat{x}]$.

Depuis cette définition, nous pouvons illustrer les trois cas possibles décrits dans la figure suivante.

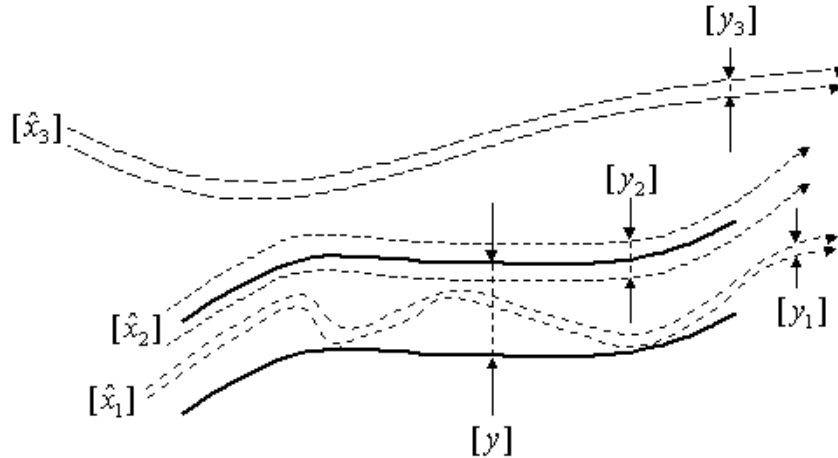


Figure 2.2 : Exemples de trajectoires bornées possibles pour le problème d'estimation ensembliste

2.3 Solution du problème d'estimation ensembliste

Dans cette partie, et dans le cadre d'approches d'estimation basées sur l'optimisation, nous développons une stratégie basée sur une méthode globale d'optimisation non-linéaire par intervalles, avec des contraintes d'inégalité comme moyen pour faire des estimations d'état, avec la méthode d'estimation à horizon glissant nommée IMHSE [Valdés-González *et al*, 2002b].

2.3.1 Principe de la méthode à horizon glissant par intervalles

La méthode IMHSE résout le problème d'estimation d'état d'un système dynamique par un problème statique d'optimisation globale non-linéaire. La stratégie consiste à minimiser globalement l'écart entre la mesure du système et sa prédiction sur un horizon de temps prédéfini. L'estimation de l'état est exécutée en trouvant la valeur de tous les vecteurs de l'état au début de l'horizon de temps prédéfini, qui minimise globalement un critère sur un domaine admissible. Le but est donc que la trajectoire de la sortie estimée du système, se rapproche autant que possible de celle qui est mesurée à partir du système réel. Une fois le vecteur initial déterminé, on calcule l'état courant à la fin de l'horizon, avec l'aide du modèle dynamique (1.1). Le critère à minimiser est l'erreur de prédiction de la sortie au sens de moindres carrés, c'est-à-dire la somme du carré des erreurs, comme en (2.12).

Soit un système dynamique non-linéaire décrit tel qu'en (1.1). Considérons aussi h observations bornées du vecteur de sortie effectuées à des intervalles de temps réguliers Δt , tel que la longueur lh de l'horizon soit $lh = (h-1) \cdot \Delta t$. Celles-ci sont résumées dans la suite $\{[y]\}_{k=sh}^{sh+lh-1}$. Dans cette suite définie sur l'horizon de temps, sh correspond au début de l'horizon et le terme $\ell = sh + lh - 1$ représente la fin de cet horizon.

Pour déterminer l'ensemble du début de l'horizon, sur un domaine admissible $[\Omega]$, nous formulons alors le problème d'estimation optimale avec contraintes, comme la solution du problème d'optimisation globale non-linéaire du critère $V(\cdot)$ suivant :

$$\begin{aligned} \text{Minimisation Globale } & V\left(x_k, \Sigma(\cdot), \{[y]\}_{k=sh}^{\ell}\right) \\ & x_k \in [\Omega] \\ & [\Omega] \subset \mathbb{R}^n \end{aligned} \tag{2.10}$$

Cette formulation générale prend en compte la représentation de perturbations et bruits typiques d'un système industriel appelé $L(\cdot)$. La formulation du critère répond à la forme classique suivante :

$$V(\cdot) = \frac{1}{2} \sum_{k=sh}^{\ell} L(\omega_k, v_k, u_k) + \Gamma(\hat{x}_{sh}) \quad (2.11)$$

Où l'expression ω_k représente les perturbations, définie comme une séquence $\{\omega_k\}_{k=sh}^{sh+lh-1}$ sur l'horizon prédéfini, si un bruit additif est choisi comme perturbation. Nous définissons v_k comme : $v_k = y_k - h(\hat{x}_k)$. De plus, $\Gamma(\cdot)$ est nommé le coût de l'arrivée du problème. Ce coût est une pénalisation qui résume les données antérieures au début de l'horizon, non contenues dans le critère. D'après l'expression (2.11), nous observons que l'information liée aux données à traiter, augmente avec le temps et la longueur de l'horizon. Notons qu'en utilisant un horizon de temps infini ($\ell \rightarrow \infty$), la solution en temps réel du problème d'optimisation globale non-linéaire (2.10), ne sera pas accessible à cause de la complexité informatique impliquée.

Une stratégie pour résoudre ce problème dans un domaine expérimental, est de travailler avec des dimensions fixes, c'est-à-dire d'utiliser une approximation d'horizon fixe de longueur lh glissante [Boillereaux, 1996], [Allgöwer *et al.*, 1999]. En d'autres termes, l'utilisation d'un critère d'optimisation sur un horizon de temps fini du type suivant.

$$V(\cdot) = \frac{1}{2} \sum_{k=sh}^{sh+lh-1} v_k^2 \quad (2.12)$$

2.3.2 L'algorithme pour la méthode IMHSE

L'algorithme itératif suivant nous donne une estimation de l'état du système dynamique (1.1) à la fin de l'horizon choisi, c'est-à-dire au temps ℓ . Cet algorithme utilise la terminologie suivante :

- T_k : Temps maximal d'estimation (durée de l'expérience).
- $[\Omega]$: Domaine admissible pour l'espace d'état en $\mathbb{I}\mathbb{R}^n$.
- sh : Début d'horizon.
- lh : Longueur de l'horizon.

B_ε	: Précision pour le problème d'optimisation (2.10). Voir chapitre 3.
$\Sigma(\cdot)$: Modèle du système dynamique.
$\{[u]\}_{sh}^\ell$: Commande bornée (si elle existe), sur l'horizon prédéfini.
$\{[y]\}_{sh}^\ell$: Mesure bornée de la sortie du système, sur l'horizon prédéfini.
$[\hat{\Theta}]_{sh}$: Etat borné solution du problème d'optimisation globale (2.10).
$\beta_{sh}(\cdot)$: Fonction de régularisation ou filtre de début d'horizon.
$[\hat{x}]_{sh}$: Etat borné estimé régularisé en début d'horizon.
$[\hat{x}]_{sh/sh-1}$: Estimation passée $[\hat{x}]_{sh-1}$ actualisée à temps sh .
$[\hat{x}]_\ell$: Etat borné estimé à l'instant courant $\ell = sh + lh - 1$, généré depuis $[\hat{x}]_{sh}$.

Algorithme 1 : IMHSE(.)

Entrée : $T_k, [\Omega], lh, B_\varepsilon, \Sigma(\cdot), \{[u]\}_{sh}^\ell, \{[y]\}_{sh}^\ell$

Sortie : $[\hat{x}]_\ell$

Initialisation : $sh := 1, \ell := sh + lh - 1$

WHILE $\ell \leq T_k$

1. Définition du domaine admissible $[\Omega]_{sh} := [\Omega]$
2. Résolution du problème global d'optimisation (2.10) par intervalles

$$[\hat{\Theta}]_{sh} := \text{G-Optimisation}([\Omega]_{sh}, B_\varepsilon, V(\cdot), \{[u]\}_{sh}^\ell, \{[y]\}_{sh}^\ell, lh)$$

3. Régularisation ou filtre

$$[\hat{x}]_{sh} := \beta_{sh}([\hat{\Theta}]_{sh}, [\hat{x}]_{sh/sh-1})$$

4. Calcul de séquence d'états jusqu'à la fin de l'horizon : $[\hat{x}]_\ell$

$$\{[\hat{x}]\}_{sh}^\ell := \text{I-Trajectoire}([\hat{x}]_{sh}, \Sigma(\cdot), sh, lh, \{[u]\}_{sh}^\ell)$$

5. Faire glisser l'horizon d'un pas pour calculer la prochaine estimation : $[\hat{x}]_{\ell+1}$

$$sh := sh + 1$$

END

Pour analyser les résultats correctement, nous devons prendre en considération que la première estimation obtenue grâce à cette méthode est possible uniquement à partir de $k = lh$ qui est la longueur prédéfinie pour l'horizon. La méthode IMHSE correspondant à l'algorithme qu'on vient de détailler est illustrée par la figure 2.3.

Le coût d'arrivé en (2.11) (ou arrival cost en anglais) est un concept théorique de la méthode à horizon fuyant qui permet de transformer le problème d'un horizon infini en un problème à horizon fini (figure 2.3). La littérature montre que la clef pour préserver la stabilité de cet observateur basé sur un problème d'optimisation et sur un

Notre algorithme utilise aussi une fonction de régularisation $\beta_{sh}(\cdot)$, qui peut être utilisée en présence de bruit dans les mesures, comme il est indiqué dans l'étape 3 de l'algorithme 1. Cette fonction met à jour l'information de l'estimation précédente quand l'horizon est décalé d'un pas. Donc, à chaque changement d'horizon, ces informations sont prises en considération grâce à une moyenne pondérée entre la déviation de l'estimation passée actualisée $[\hat{x}]_{sh/sh-1}$, et la solution $[\hat{\Theta}]_{sh}$ du problème (2.10), en début d'horizon, cf. figure 2.3.

$$\beta_{sh}([\hat{\Theta}]_{sh}, [\hat{x}]_{sh/sh-1}) = \alpha \cdot [\hat{\Theta}]_{sh} + (1 - \alpha) \cdot [\hat{x}]_{sh/sh-1} \quad (2.13)$$

Le paramètre de pondération α est tel que $\{\alpha/\alpha \in \mathbb{R}, \alpha \in [0,1]\}$. Si $\alpha = 1$, aucun filtrage n'est effectué. Si $\alpha = 0$, l'estimation est égale à la prédiction obtenue par le modèle. Le réglage du paramètre est fait de façon expérimentale [Boillereaux, 1996], [Flaus, 1998]. De manière théorique, le filtrage n'est possible qu'à partir du deuxième point estimé. Mais dans la pratique lorsqu'il y a des mesures ou conditions initiales des états, celles-ci peuvent être utilisées pour démarrer le filtre depuis le premier point estimé.

Avec cette étape de mise à jour ou régularisation, nous transférons l'information antérieure à la fenêtre courante (temps d'horizon considéré). Ceci conditionne les estimations en début de l'horizon. Nous remarquons que cette régularisation ou filtre de premier ordre (2.13), peut être vue à la fois comme filtre et $\Gamma(\cdot)$ [Rao *et al*, 2001].

Remarque 6 : *Observabilité. La solution au problème (2.10) est unique sur un domaine admissible si et seulement si le système est observable. Si le système est observable et que l'entrée est une entrée non-singulière sur un horizon de temps donné, alors nous pouvons distinguer des états différents avec l'algorithme IMHSE pour une certaine précision d'optimisation. Inversement si l'algorithme IMHSE ne peut pas distinguer des solutions, soit le système n'est pas observable, soit la précision est insuffisante. En d'autres termes, si le système (1.1) impliqué n'est pas observable alors la solution du problème d'optimisation global n'est pas unique, l'algorithme fourni l'ensemble de tous les états indiscernables solutions du problème (2.10). Voir section 2.2 de ce chapitre, et le chapitre 4, pour de plus amples informations.*

D'après la remarque 6, l'analyse des résultats de (2.10) nous donne des informations sur l'observabilité locale du système. Autrement dit la taille de l'ensemble $[\hat{\Theta}]_{sh}$ peut nous renseigner sur la possibilité de distinguer différents états du système impliqué, pour une marge donnée de la trajectoire de sortie. Ainsi dans le cas de non-observabilité locale, nous pourrions la détecter par exemple grâce à un programme superviseur, et fixer par défaut (sous quelques conditions), la valeur de l'estimation $[\hat{x}]_{sh}$ à celle de l'estimation antérieure actualisée, c'est-à-dire $[\hat{x}]_{sh/sh-1}$.

Notons aussi que dans la méthode IMHSE le seul paramètre à ajuster est la longueur d'horizon lh . Dans [Boillereaux, 1996], [Boillereaux et Flaus, 2000] et [Flaus, 1998] nous pouvons trouver des relations théoriques et expérimentales entre la longueur de l'horizon, la constante de temps du système, le nombre de points suffisant sur l'horizon pour distinguer les états et l'observabilité numérique impliqué (voir section 2.6).

Remarque 7 : *Propriété de convergence globale. Il est important de noter que la convergence globale de cet algorithme IMHSE d'estimation ensembliste est donnée par l'usage d'une technique globale d'optimisation. Ceci est fondé sur la capacité algorithmique de trouver tous les états \hat{x} (états indiscernables d'un ensemble $[\hat{x}]$), qui minimisent globalement la fonction objectif (2.12) sur un domaine admissible $[\Omega] \subset \mathbb{R}^n$, avec une précision d'optimisation qui peut être arbitrairement petite.*

D'autre part, cet algorithme peut être divisé en deux parties principales :

- _ Le problème d'optimisation globale par intervalles (2.10), pour résoudre l'étape 2 du dernier algorithme. Voir *chapitre 3* pour la méthode d'optimisation globale utilisée et l'algorithme nommé G-optimisation(.).
- _ Le problème de prédiction. Notons que pour appliquer notre algorithme d'estimation, nous avons besoin d'évaluer les trajectoires de l'état du système (1.1) par intervalles. C'est-à-dire de calculer sur l'horizon de temps défini, la suite $\{[\hat{x}]_{sh}^\ell$ qui contient toutes les valeurs possibles de l'état engendrées depuis le vecteur de l'état intervalle initial en début d'horizon $[\hat{x}]_{sh}$.

2.3.3 L'algorithme d'obtention de trajectoires

Pour évaluer les trajectoires estimées par intervalles du système discret (1.1), entre deux instants de temps et une commande appliquée à partir d'un état intervalle initial, nous utiliserons l'algorithme suivant que considère l'extension intervalle $F(\cdot)$ et $H(\cdot)$ de $f(\cdot)$ et $h(\cdot)$ du système (1.1) respectivement (il faut noter le besoin des méthodes d'intégration dans le cas de systèmes continus).

Algorithme 2 : I-Trajectoire(.)

Entrée : $[\hat{x}]_{sh}$, $\Sigma(\cdot)$, sh , lh , $\{[u]\}_{sh}^\ell$

Sortie : $[\hat{x}]_\ell$, $\{[\hat{x}]\}_{sh}^\ell$, $\{[\hat{y}]\}_{sh}^\ell$

Initialisation : $\ell := sh + lh - 1$

FOR $k = sh$ to ℓ

1. $[\hat{x}]_{k+1} := F([\hat{x}]_k, [u]_k)$

2. $[\hat{y}]_k := H([\hat{x}]_k)$

END

Remarque 8 : *La fonction de l'inclusion utilisée est l'extension naturelle. (Voir section 2.1.1.3 de ce chapitre pour plus d'informations).*

Remarque 9 : *L'effet enveloppant. Le calcul de l'ensemble exact qui contient la trajectoire estimée, est une tâche difficile pour un système non-linéaire en général. Le problème principal est lié à l'effet de majoration, car chaque résultat intermédiaire est approché par un vecteur de l'intervalle qui le contient : l'erreur augmente à chaque pas. Heureusement, dans le cas de notre observateur ensembliste, nous considérons seulement des horizons de temps courts, et donc le calcul des bornes de la trajectoire ainsi obtenu est satisfaisant.*

2.3.4 Prédiction de l'espace de recherche

Une variante intéressante de la méthode IMHSE présentée dans l'algorithme 1, est l'utilisation d'une prédiction de l'espace de recherche $[\Omega]_{sh}$ pour la prochaine estimation. Cet espace est égal à la prédiction en un pas de $[\hat{x}]_{sh-1}$ obtenue par le modèle (1.1), augmenté d'un pourcentage δ (choisi arbitrairement entre 5% et 10% voir figure 2.4). Cette prédiction réduit la complexité informatique nécessaire dans le

problème d'estimation et notamment le temps de calcul (voir *section 3.2.7*). Elle autorise une réduction de la longueur de l'horizon jusqu'à la dimension de l'état [Valdés-González et Flaus, 2001a].

Une expansion appelé $Expansion([x], \delta)$, de $\delta\%$ sur tous les axes d'une boîte $[x] \in \mathbb{IR}^n$, tient compte des erreurs de modèle et des perturbations (bruit de mesure). Il est défini comme :

$$Expansion([x], \delta) = [[a_1 - \delta, b_1 + \delta] \times \dots \times [a_n - \delta, b_n + \delta]] \quad (2.14)$$

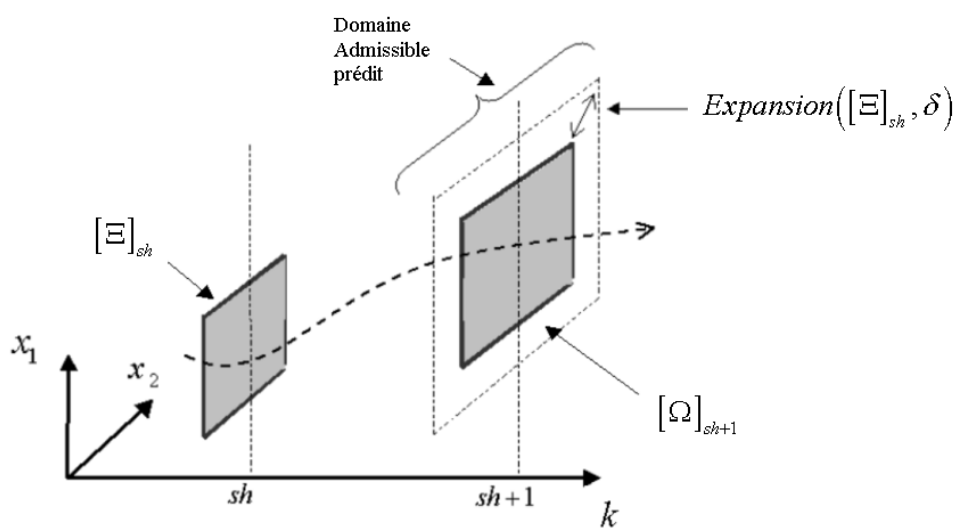


Figure 2.4 : La Prédiction de l'espace de recherche

Pour effectuer cette prédiction, un changement dans la première étape de l'algorithme $IMHSE(\cdot)$ peut être effectué.

1. Définition du domaine admissible $[\Omega]$

IF $[\hat{x}]_{sh-1}$ existe (solution précédente)

$$[\Xi]_{sh} := \text{I-Trajectoire}([\hat{x}]_{sh-1}, \mathbf{I}\Sigma(\cdot), sh-1, sh, \{[u]\}_{sh-1}^{sh})$$

$$[\Omega]_{sh} := \text{Expansion}([\Xi]_{sh}, \delta)$$

ELSE

$$[\Omega]_{sh} := [\Omega]$$

END

Notons que lorsque cette prédiction de l'espace de recherche est utilisée, la propriété de convergence globale sur le domaine admissible est perdue (au moins en dehors de l'espace de recherche prédit). Cependant quand la solution en début de l'horizon du problème (2.10) n'est pas contenue dans l'espace de recherche prédit $[\Omega]_{sh}$, autrement dit cette solution est vide, nous pouvons utiliser tout le domaine admissible prédéfini $[\Omega]$.

2.4 L'observateur IMHSE et les mesures hors-ligne

Dans certains procédés, comme les procédés biotechnologiques, il peut y avoir plus d'une sortie pour le système à mesurer. Une mesure partielle des états impliqués peut être faite de temps en temps, mais ces mesures ne sont pas toujours disponibles en ligne. Supposons la sortie bornée du système (1.1) tel que $[y]_k \in \mathbb{R}^p$ avec une période d'échantillonnage que nous nommerons Δt_1 . Quelques mesures partielles sur l'état du système sont obtenues avec une autre période d'échantillonnage, que nous nommerons ici Δt_2 (avec $\Delta t_1 \ll \Delta t_2$). Les mesures faites avec la période d'échantillonnage Δt_2 , sont telles que $x_k^j \in \mathbb{R}^b$. Cela peut être une mesure partielle ou totale de l'état, borné ou exacte mais hors ligne. Un diagramme représentant ce principe est donné dans la figure 2.5, où (\bullet) représente la mesure de la sortie du système, disponible en ligne, et (\diamond) représente la mesure partielle ou totale de l'état, mais hors ligne.

Ces mesures hors ligne sont prises en compte par la méthode IMHSE, de la même façon que les mesures en ligne. Nous utilisons pour cela la méthode décrite par l'algorithme 1. Notons donc que notre observateur peut être recalé sans problème, et que ces nouvelles informations simplifient d'autant plus le problème d'optimisation globale (2.10).

Sous ces conditions, le critère (2.11) du problème d'estimation optimale (2.10), peut être reformulé comme suit :

$$V(.) = \frac{1}{2} \sum_{j=1}^b \sum_{k=sh}^{\ell} \rho_j \cdot L(\omega_k, v_k, u_k) \quad (2.15)$$

Ou plus simplement :

$$V(.) = \frac{1}{2} \left(\sum_{k=sh}^{\ell} (v_k)^2 + \sum_{j=1}^b \rho_j \cdot (\psi_k^j)^2 \right) \quad (2.16)$$

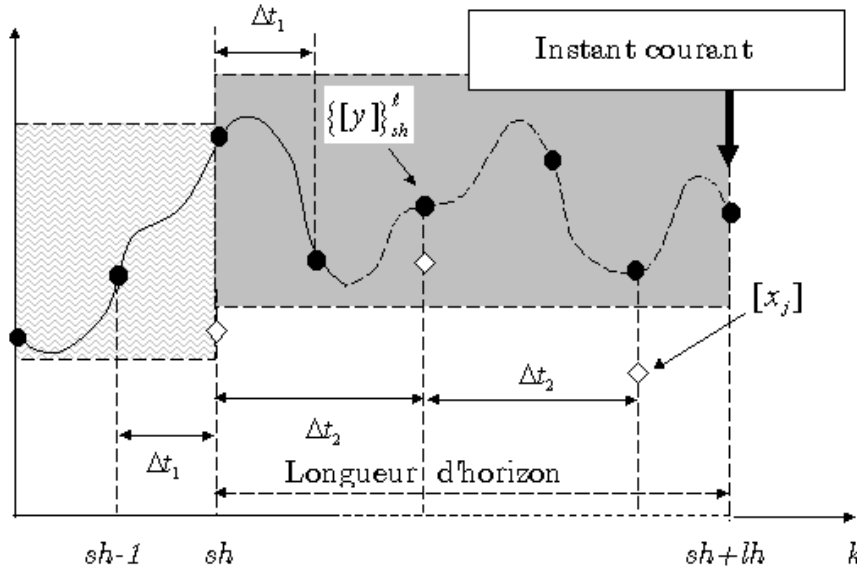


Figure 2.5 : Mesures en ligne et hors ligne

Avec $v_k = y_k - h(x_k)$ et $\psi_k^j = x_k^j - \hat{x}_k^j$ qui représentent l'erreur de prédiction de la sortie en ligne et hors ligne respectivement. ψ_k^j représente l'écart entre la composante j de l'état estimé \hat{x}_k et la composante j de l'état x_k mesuré hors ligne.

Comme il s'agit d'un système à événements discrets, considérons pour simplifier que l'expression $\Delta t_2 / \Delta t_1 \in \mathbb{N}$. Alors le paramètre ρ est un vecteur constant, qui indique si la mesure hors ligne de la composante j du vecteur d'état est disponible ou non à l'instant k . La notation suivante est utilisée :

$$\rho_j = \begin{cases} 1, & \text{quand } x_k^j \text{ est disponible à l'instant } k \\ 0, & \text{quand } x_k^j \text{ n'est pas disponible à l'instant } k \end{cases} \quad (2.17)$$

Notez que le critère (2.12) est un cas particulier de (2.16) quand il n'y a pas de mesures hors ligne.

2.5 L'observateur IMHSE et les systèmes décrits par des modèles hybrides

Un système dynamique est susceptible d'être modélisé par un système hybride. Un système hybride est une combinaison de plusieurs modèles qui sont commutés en fonction de la trajectoire que suit le système. La trajectoire évolue d'une part en

fonction des changements instantanés de modèles (places) et d'autre part en fonction de l'état évoluant suivant l'inclusion différentielle de chaque place [Flaus, 1998].

Ce type de modèles est très utilisé dans des procédés tels que les processus chimiques [Pérez-Correa et Agosin, 1999], [Gelmi, 1999]. Des modèles de ce type ont été utilisés avec succès pour modéliser le comportement d'un paramètre dans un procédé biotechnologique de type batch. Ce paramètre est le taux spécifique de croissance, particulièrement difficile à modéliser. Modélisé par une loi de Monod, il a été remplacé par un modèle hybride [Valdés-González et Flaus, 2001b]. Ce type de modèle a aussi été utilisé pour un procédé de fermentation sur substrat solide, dont un état intermédiaire a été modélisé de façon hybride (voir *chapitre 6*) [Valdés-González et al, 2002b].

2.5.1 Automates hybrides

Un automate hybride est un outil de modélisation informatique. Il est constitué d'un ensemble de places, et d'arcs entre chaque place. A chaque place est associée une équation différentielle quelconque (équations aux différences finies dans le cas à temps discret). Chaque arc est étiqueté par une condition de transition et une relation de saut ou de réinitialisation. A chaque instant, l'état de l'automate hybride est donné par la paire (l, x) où l est la place, $x \in \mathbb{R}^n$ est l'état continu et (l, x_0) est l'état initial. La trajectoire de l'automate évolue d'une part en fonction des changements instantanés de places, et d'autre part en fonction de l'état continu évoluant suivant l'inclusion différentielle dans chaque place. La définition formelle [Flaus, 1998] [Thévenon, 2000] est la suivante :

Définition 3. *Un automate hybride est défini par la donnée de $H = (L, X, E, I, F)$ où :*

- *L est un ensemble fini de sommets appelés places*
- *$X \subset \mathbb{R}^n$ est l'espace d'état continu de l'automate. Un état est une paire (l, x_k) composé d'une place $l \in L$ et d'un vecteur $x_k \in X$.*
- *E est un ensemble de transitions représentées par un arc entre les places. Une transition est définie par le quadruplet $(l, Guard, Jump, l')$. La transition est validée lorsque $x_k \in Guard$, et lors du franchissement de la transition l'état x subit un saut et passe à la valeur définie par la relation $Jump(x_k, x_k')$.*

- $I = \{inv(l) \subset X | l \in L\}$ est une collection d'ensembles appelés invariants de la place. Lorsque le système est dans la place l , l'état discret doit vérifier $x_k \in Inv(l)$.
- F est un ensemble d'applications associant à chaque vecteur d'état un sous ensemble de l'espace d'état, noté $f_l(x_k)$. Une application est associée à chaque place. La dynamique discrète est définie par une inclusion différentielle $x_{k+1} \in f_l(x_k)$, où $f_l(x_k)$ est appelée l'activité de la place l .

Remarquons que ce type de modèle permet de représenter des sauts et des commutations aussi bien autonomes que contrôlés. La figure 2.6 donne un exemple d'automate hybride représentant un système dynamique. Le principe de fonctionnement du système est de maintenir l'état $x_k(1)$ dans un intervalle donné : $[\gamma_1, \gamma_2]$. Dans la place l_1 la dynamique est définie par l'inclusion $f_1(x_k, u_k)$, et dans la place l_2 pour $f_2(x_k, u_k)$. Dans la place l_1 , l'état $x_k(1)$ vérifie $x_k(1) < \gamma_2$ et dans la place l_2 il vérifie $x_k(1) > \gamma_1$. Ces 2 transitions permettent de passer d'une place à une autre en fonction de l'état $x_k(1)$ évoluant suivant l'inclusion correspondante.

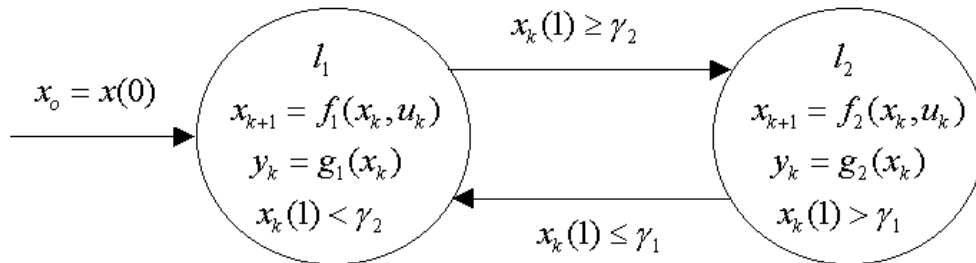


Figure 2.6 : Automate hybride

2.5.2 Automate hybride ensembliste

Un automate hybride dont l'état de la place est défini par intervalles, est appelé automate hybride ensembliste. Cet automate ensembliste, présente la caractéristique suivante : une commutation peut avoir lieu entre deux instants simultanément, c'est-à-dire, qu'un état intervalle appartiendra à plus d'une place à la fois.

Ceci implique qu'à chaque instant de l'évolution de l'état d'une place, l'état doit être intersecté avec son domaine et avec sa frontière. Ce qui nous emmène d'une part à évaluer une partie de l'ensemble selon l'inclusion d'une place, et d'autre part, à évaluer une autre partie du même ensemble selon l'inclusion d'une autre place.

Pour illustrer d'une façon simple la problématique : supposons le problème optimal (2.10) de la méthode IMHSE lié à un système dynamique discret (1.1), décrit de façon générale par un modèle hybride de la forme :

$$\begin{aligned} x(k+1) &= f_m(x(k), u(k)) \\ y(k) &= g_m(x(k)) \end{aligned} \quad (2.18)$$

m représente le nombre de places du modèle hybride. Lorsque les états sont bornés, il peut arriver lors de la résolution du problème (2.10) qu'un état soit concerné par plus d'une place, particulièrement au moments des transitions. Pour illustrer ceci, considérons un système 2D décrit par un modèle hybride vérifiant (2.18) et représenté par la figure 2.6, dont l'état $[x] \in \mathbb{R}^2$ est tel que $[x] = [x_1] \times [x_2]$. Cet état est représenté par la figure 2.7.

Sous ces conditions, l'état $[x]$ dans la figure 2.7 appartient bien aux deux places du système représenté par la figure 2.6.

Pour traiter ce cas, nous pouvons renommer la composante $[x_1]$ de $[x]$ telle que $[x_1] = [x_1]^* \cup [x_1]**$, où $[x_1]^* = [a_1, \gamma_2]$ et $[x_1]** = [\gamma_1, b_1]$.

L'état de l'automate hybride est donné par les paires $(l_1, [x_1]^*)$ et $(l_2, [x_1]**)$. La trajectoire de l'automate évolue alors d'une part en fonction de la place et d'autre part en fonction de l'état intervalle.

Cette manière d'aborder le problème borné nous conduit à redéfinir et généraliser (2.18) de la façon suivante :

$$x(k+1) = \bigcup_{i=1}^m f_i(\{l_i, x(k)\}, u(k)) \quad (2.19)$$

Dans notre cas, $[x]_{k+1} = F_1([x]_k^*, u_k) \cup F_2([x]_k**, u_k)$ est l'évolution de la trajectoire de l'automate, où $[x]_k^* = [x_1]_k^* \times [x_2]_k$, $[x]_k** = [x_1]_k** \times [x_2]_k$ et où $F_1(\cdot)$ et $F_2(\cdot)$ sont les fonctions d'inclusion de $f_1(\cdot)$ et $f_2(\cdot)$ respectivement.

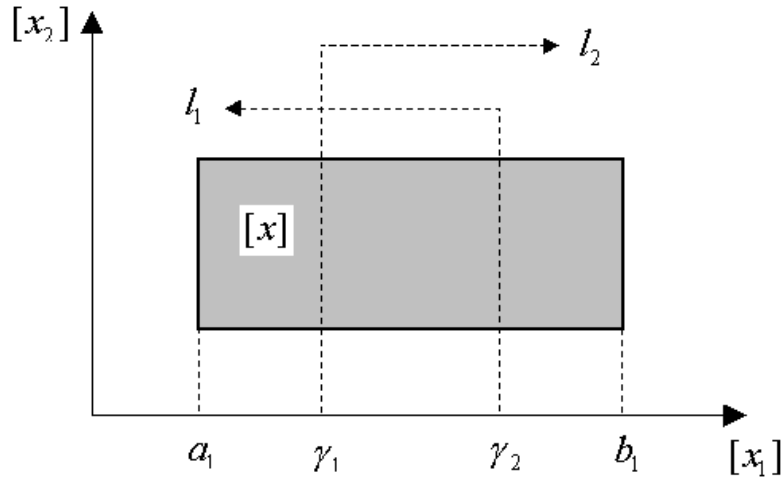


Figure 2.7 : Etat intervalle appartenant à deux places

2.6 La longueur de l'horizon

La méthode d'estimation IMHSE est intéressante car nous avons besoin seulement d'un modèle du système, et non de le linéariser, ni de connaître a priori les perturbations. Il nous reste donc seulement à régler la longueur de l'horizon lh .

Si n est la dimension du vecteur d'état, un nombre de points h sur l'horizon égal à $2 \cdot n + 1$ est suffisant pour recouvrir l'état [Aeyels, 1981]. De plus, une longueur de l'horizon lh plus grande que n donnera une plus faible sensibilité au bruit de mesure [Boillereaux, 1996]. Soit $lh = (h - 1) \cdot \Delta t$ la longueur de l'horizon. Nous allons étudier l'effet de la longueur de l'horizon sur notre observateur.

2.6.1 Effet d'une longueur d'horizon trop courte

Classiquement un système dont il n'existe pas de points indiscernables est dit observable. C'est-à-dire :

Définition 4. *Un système du type (1.1) est dit observable sur un intervalle de temps $[0, T]$, si pour deux conditions initiales différentes, les sorties associées ne sont pas identiquement égales sur $[0, T]$. Autrement dit :*

Si $x_1(k)$ et $x_2(k)$ sont deux trajectoires du système (1.1), telles que $x_1(k) \neq x_2(k)$, alors il existe $k \in [0, T]$, et $u(k)$ sur $[0, k]$ tel que $h(x_1(k)) \neq h(x_2(k))$ [Hermann et Krener, 1977], [Othman, 1992], [Hammouri et Othman, 2001].

Ainsi, pour tout $lh > k$ le système est observable. Mais il peut exister une longueur d'horizon $lh^- < k$, telle que les trajectoires ne soient plus distinctes, et donc le système n'est plus observable.

2.6.2 Relation avec le temps de réponse du système

Si l'horizon est trop long, $lh > k_r$ (k_r est le temps de réponse du système) la méthode à horizon fuyant se comporte comme un simulateur du procédé et non comme un observateur [Boillereaux, 1996], [Flaus, 1998]..

2.6.3 Bornes pour l'horizon

Il est maintenant possible de fournir des bornes pour la longueur de l'horizon :

$$lh^- < lh < k_r$$

Plus l'horizon sera court, plus les perturbations sur le système seront prises en compte par l'observateur et le système peut devenir numériquement non observable. D'autre part, avec une longueur d'horizon trop importante, l'observateur peut se comporter comme un simulateur du procédé [Boillereaux, 1996], [Flaus, 1998]. Il faut donc faire un compromis.

2.7 IMHSE globalement convergente

Tout d'abord nous remarquons ici le fait que si la solution du problème d'optimisation globale (2.10) existe et est unique, alors elle conduit bien à l'estimation de l'état du système (1.1). Ensuite, l'algorithme global d'optimisation est utilisé pour la minimisation du critère (2.12). IMHSE converge globalement vers l'optimum sur un domaine admissible. C'est-à-dire que lorsque le minimum global du critère (2.12) est trouvé pour un modèle du système du type (1.1), la méthode IMHSE fournit l'estimation de l'état en début d'horizon.

A partir de la Remarque 6 à propos de l'observabilité du système, la solution ensembliste au problème d'optimisation du critère (2.10) est unique sur un domaine admissible, si et seulement si, le système est observable. Dans le cas où le système (1.1) ne serait pas observable, alors la solution du problème d'optimisation globale n'est pas unique (l'algorithme retourne un ensemble de tous les états indiscernables).

Grâce à la technique globale d'optimisation par intervalles utilisé pour résoudre le problème optimal (2.10), les conditions imposées sur le système (1.1) seront bien peu exigeantes. En effet, la propriété de convergence dépendra uniquement de la méthode d'optimisation utilisée. C'est-à-dire, la méthode d'estimation IMHSE converge globalement parce que l'algorithme d'optimisation globale utilisé est capable de trouver de façon garantie tous les états indiscernables, qui minimisent globalement le critère (2.12) sur un domaine admissible (voir Chapitre 3).

Remarque 10: *En utilisant une méthode garantie d'optimisation globale ensembliste, nous pouvons trouver tous les minimum globaux du critère (2.12), sur un domaine admissible même dans le cas où le critère ne serait pas convexe. Nous verrons dans le chapitre suivant que la méthode d'optimisation peut éliminer les minimum locaux du problème.*

2.7.1 Besoin d'une technique globale d'optimisation

Remarquons que pour fournir l'estimation de l'état en début d'horizon, pour le problème (2.10), nous avons besoin de trouver le ou les minimum globaux du critère (2.12). De plus, l'observabilité du système (1.1) est déterminée, au sens des intervalles, par l'ensemble des solutions du problème d'optimisation obtenues pour une marge de la trajectoire de sortie.

Nous allons donc nous intéresser dans le chapitre suivant à une méthode efficace d'optimisation globale par intervalles développée par [Leclerc, 1992].

2.8 Stabilité et convergence de la méthode IMHSE

Dans cette partie nous étudions la convergence de la méthode IMHSE en nous appuyant sur les résultats donnés en [Leclerc, 1992] et [Boillereaux, 1996].

2.8.1 Cas 1 : Sans filtrage

- Hypothèses:

H_1 : Le système est observable

H_2 : Pas de bruit sur la sortie du système

H_3 : $\partial(x, k_1, k_2)$ correspond à la fonction de transition du système (1.1), c'est-à-dire, un équivalent de l'algorithme 2 qui évalue la trajectoire du système (1.1) en partant de l'état x entre les temps k_1 et k_2 .

Théorème 1 : Si H_1, H_2 sont vérifiés et le système (1.1) est tel que $\partial(\cdot)$ est Lipschitz, c'est-à-dire :

$$\left\| \partial(x_1, sh, sh+lh) - \partial(x_2, sh, sh+lh) \right\| \leq a \cdot \|x_1 - x_2\| \quad \forall x_1, x_2$$

Alors la méthode IMHSE sans filtrage est stable et converge globalement vers l'état réel x_{sh+lh} .

Démonstration : Si l'hypothèse H_1 est vérifiée, alors l'algorithme d'optimisation global **G-Optimisation**(.) converge vers le vecteur d'état solution \hat{x}_{sh}^V qui minimise globalement le critère (2.12) sur un domaine admissible $[\Omega]$ pour l'état. En d'autres termes nous pouvons trouver dans un nombre fini d'itérations (noté par un "**") un vecteur d'état solution $\hat{\Theta}_{sh}^*$ tel que :

$$\left\| \hat{\Theta}_{sh}^* - \hat{x}_{sh}^V \right\| \leq \varepsilon_1$$

où $\varepsilon_1 \in \mathbb{R}^+$. De plus ε_1 peut être aussi petit que l'on veut sous l'hypothèse H_2 .

Comme le minimum de $V(\cdot)$ est unique par l'hypothèse H_1 d'observabilité, et si nous désignons le vecteur x_{sh} comme le vecteur d'état réel en début de l'horizon, nous pouvons écrire que :

$$\hat{x}_{sh}^V = x_{sh}$$

La condition de Lipschitz sous l'hypothèse H_3 nous permet alors de réécrire :

$$\left\| \partial(\hat{\Theta}_{sh}^*, sh, sh+lh) - \partial(x_{sh}, sh, sh+lh) \right\| \leq a \cdot \left\| \hat{\Theta}_{sh}^* - x_{sh} \right\|$$

$$\left\| \hat{x}_k^* - x_k \right\| \leq a \cdot \left\| \hat{\Theta}_{sh}^* - x_{sh} \right\|$$

Par conséquent, avec ε_1 aussi petit que l'on veut, la méthode IMHSE sans filtrage converge globalement vers l'état courant réel du système :

$$\left\| \hat{x}_k^* - x_k \right\| \leq a \cdot \varepsilon_1$$

En effet si le système est observable, le minimum global du critère existe et est unique. Alors ceci sera toujours obtenu et l'erreur d'estimation converge vers 0.

2.8.2 Cas 2 : Avec filtrage

Théorème 2 : Si H_1, H_2 sont garanties et le système (1.1) est tel que $\partial(\cdot)$ est Lipschitz, c'est-à-dire :

$$\left\| \partial(x_1, sh+lh-1, sh+lh) - \partial(x_2, sh+lh-1, sh+lh) \right\| \leq b \cdot \|x_1 - x_2\| \quad \forall x_1, x_2$$

Alors la méthode IMHSE avec filtrage est stable et converge globalement vers l'état réel x_k , avec $k = sh + lh$.

Démonstration : Le résultat du cas 1 (système sans filtrage), nous permet d'écrire que :

$$\left\| \hat{x}_k^* - x_k \right\| \leq a \cdot \varepsilon_1$$

Notons que le filtre est donné par l'expression :

$$\hat{x}_k = \alpha \cdot \hat{\Theta}_k^* + (1-\alpha) \cdot \hat{x}_{k/k-1}$$

Si x_k est l'état courant réel du système nous pouvons écrire que :

$$\hat{x}_k - x_k = \alpha \cdot (\hat{\Theta}_k^* - x_k) + (1-\alpha) \cdot (\hat{x}_{k/k-1} - x_k)$$

Si pour un instant précédent l'état vérifie que :

$$\left\| \hat{x}_{k-1}^* - x_{k-1} \right\| \leq \varepsilon_{k-1}$$

où $\varepsilon_{k-1} \in \mathbb{R}^+$. De plus ε_1 peut être aussi petit que l'on veut sous l'hypothèse H_2 , et si $\alpha \in [0,1]$ alors nous pouvons vérifier l'inégalité suivante :

$$\left\| \hat{x}_k - x_k \right\| \leq \alpha \cdot \left\| \hat{\Theta}_k^* - x_k \right\| + (1-\alpha) \cdot \left\| \hat{x}_{k/k-1} - x_k \right\|$$

Notons que $x_k = F(x_{k-1}, sh+lh-1, sh+lh)$ alors :

$$\left\| \hat{x}_k - x_k \right\| \leq \alpha \cdot \left\| \hat{x}_k^* - x_k \right\| + (1-\alpha) \cdot \left\| \hat{x}_{k/k-1} - x_{k/k-1} \right\|$$

Avec la condition de Lipschitz, nous obtenons :

$$\left\| \hat{x}_k - x_k \right\| \leq \alpha \cdot a \cdot \varepsilon_1 + (1-\alpha) \cdot b \cdot \varepsilon_{k-1}$$

Soit $\varepsilon_k = \alpha \cdot a \cdot \varepsilon_1 + (1-\alpha) \cdot b \cdot \varepsilon_{k-1}$, avec $\varepsilon_k \in \mathbb{R}^+$. Un réel positif strictement aussi petit que l'on veut. Alors la méthode IMHSE avec filtrage converge globalement vers l'état courant réel du système.

Remarquons que le coefficient α a une influence sur la vitesse de convergence, et corrige l'influence des valeurs a et b .

- Commentaires

Notons aussi que les preuves apportées sont basées sur des hypothèses fortes. Si l'hypothèse d'absence de bruit n'est pas accomplie, alors le minimum du critère, n'est pas nécessairement la solution du problème d'estimation. Autrement dit :

$$\hat{x}_{sh}^V \neq x_{sh}$$

La méthode est stable mais l'erreur d'estimation converge vers une valeur bornée.

2.9 Conclusions

Dans ce chapitre, nous avons fait d'abord un survol de l'analyse par intervalles, afin de poser le problème d'un observateur ensembliste. Nous avons proposé en particulier une méthode d'estimation d'états numérique par intervalles, ayant la caractéristique d'être globalement convergente. Cette méthode est basée sur une approche d'estimation à horizon glissant local classique, mais couplée à une méthode d'optimisation globale par intervalles, appelée IMHSE.

Cette nouvelle méthode d'estimation proposée fournit l'état estimé par intervalles, à partir d'un ensemble de solutions trouvées sur un domaine admissible. Cet ensemble correspond à la solution d'un problème d'optimisation globale.

Cette approche nous permet de détecter les problèmes d'observabilité du système étudié, pour une précision d'optimisation donnée et pour une marge obtenue de la trajectoire de sortie. C'est-à-dire que si le système n'est pas observable, la solution du problème d'optimisation globale n'est pas unique : l'*algorithme 1* retourne un ensemble de tous les états indiscernables $[\hat{\Theta}]_{sh}$, qui minimisent globalement le critère (2.12).

En général, nous ne cherchons pas à minimiser infiniment l'écart de l'erreur de prédiction de la sortie, mais à rester dans des bornes de tolérance τ données. Nous "*cherchons*" donc les états en début d'horizon, dont la trajectoire issue est contenue dans le vecteur de mesures à erreurs bornées $[y]$.

Bien que la définition des critères (2.11) et (2.12), soit assez générale, nous nous sommes intéressés à modifier le critère d'optimisation du problème optimal pour recueillir des mesures hors ligne (2.16). Ces mesures sont souvent disponibles dans des systèmes tel que les bioprocédés. De plus, notre approche peut aussi être utilisée pour des systèmes décrits par modèles des hybrides. En ce qui concerne les systèmes hybrides dont les états sont représentés par des vecteurs intervalles, nous avons résolu le problème des états qui sont concernés par plus d'une place, particulièrement au moment des transitions.

Nous allons nous intéresser dans le chapitre suivant à la solution du problème optimal global par intervalles (2.10), base de notre approche.

Chapitre 3

Optimisation globale par intervalles

La problématique liée à l'optimisation globale d'une fonction non-linéaire quelconque, a été considérée pendant quelques années comme un problème très difficile à traiter, voire impossible à traiter avec un nombre fini de pas. Actuellement quelques chemins mènent à l'optimum global d'une fonction, quand celui-ci daigne bien exister !

Chercher l'optimum global d'une fonction à plusieurs variables réelles, avec ou sans contraintes est un problème d'actualité. En effet, il n'y a aucune méthode exacte pour résoudre de façon générale ce problème [Leclerc, 1992], [Van-Iwaarden, 1996] et [Jaulin, 2000].

Si nous étudions la bibliographie sur le sujet, nous trouverons par exemple, qu'il y a des problèmes d'optimisation globaux NP complet dans la pratique [Murty et Kabadi, 1987], voir aussi [Pardalos et Vavasis, 1991] pour des problèmes NP-difficiles en programmation quadratique. Ces problèmes se retrouvent dans presque tous les domaines scientifiques et industriels [Cherruault, 1998], [Díaz *et al*, 1996] [Cherruault, 1999].

Parmi les exemples de problèmes non-linéaires dont l'optimum global est nécessaire ou souhaitable, nous pouvons trouver ceux qui sont liés aux mélanges corrects lors des opérations optimales sur des procédés chimiques [Pérez-Correa et Agosin, 1999], des procédés biotechnologiques et en biologie [Ammar, 1995], [Bendiab, 1995], [Cherruault, 1998].

Dans ce travail, nous ne nous intéressons pas à la classification des problèmes, mais à leur résolution, et en particulier au problème global optimal d'estimation IMHSE présenté en (2.10).

3.1 Survol général des méthodes d'optimisation globale

En général, nous pouvons trouver deux approches d'optimisation globale.

- L'approche probabiliste
- L'approche déterministe

Nous allons présenter dans cette partie une brève description des deux approches nommées ci-dessus.

Dans le premier cas, l'approche probabiliste consiste à traiter un ensemble de points avec une stratégie donnée, pour lequel une fonction densité de probabilité mathématique est donnée. De cette façon nous pouvons obtenir le meilleur minimum de la fonction, en nous approchant de façon successive de la valeur de la fonction où l'optimum global se trouve.

La deuxième approche, est caractérisée par le fait qu'elle donne comme résultat tous les optimum globaux d'un critère donné.

3.1.1 Méthodes probabilistes

Les méthodes probabilistes pour l'optimisation globale sont des processus infinis, c'est-à-dire que pour que la probabilité d'avoir rencontré l'optimum global se rapproche de 1, le nombre de pas (visites) tend vers l'infini.

3.1.1.1 Recherche Aléatoire (*Random Search*)

Dans ce contexte, la méthode la plus basique est appelée de *Recherche Aléatoire* (Random Search). Celle-ci est une technique générale d'optimisation stochastique, qui converge vers le minimum global sur \mathbb{R}^n , la procédure converge vers la valeur minimale de la fonction sur l'espace de recherche au sens probabiliste [Atkinson, 1989], [Díaz *et al*, 1996]. Si la fonction est évaluée pour des points qui sont tirés d'une distribution de probabilité sur une région S, alors il peut être montré que la plus

petite valeur de la fonction est trouvée avec une probabilité égale à 1, et que l'algorithme converge vers la valeur minimale globale y^* .

3.1.1.2 Recuit Simulé (Simulated Annealing)

La méthode *Recuit Simulé* est une autre méthode probabiliste qui a été utilisée avec succès en optimisation, pour la programmation entière par exemple [Johnson, 1988] et plus récemment pour des problèmes généraux d'optimisation globale [Dekker et Aarts, 1991], [Díaz *et al*, 1996].

Cette technique est basée sur le principe physique de refroidissement des métaux pour trouver l'état d'énergie le plus bas de ce métal. Pour cette raison *Recuit Simulé* utilise un vocabulaire en rapport avec cette interprétation physique [Kirpatrick *et al*, 1983]. L'interprétation mathématique de *Recuit Simulé* est un algorithme qui simule le refroidissement, et qui est une fonction décroissante en fonction du temps. Sa convergence globale au sens probabiliste a été montrée par [Lundy *et Mees*, 1986]. Une discussion sur cette convergence globale qui implique un temps de convergence exponentiel est donné en [Eglese, 1990]. Des commentaires à propos de différents types d'implémentations plus performants sont faits en [Heine, 1994].

3.1.1.3 Recuit Simulé Adaptatif

La méthode *Recuit Simulé* a été développée et modifiée par beaucoup de personnes différentes. Une de ces modifications est appelée *Recuit Simulé adaptatif*. Le logiciel a été appelé *Recuit Simulé très rapide* ou (VFSA - Very Fast Simulated Annealing) et publié dans [Ingber, 1989].

Un problème avec la méthode *Recuit Simulé* standard est que chaque paramètre ou variable est "refroidie" au même taux, sans se soucier de sa grandeur, valeur ou sensibilité. Cependant la méthode *Recuit Simulé Adaptatif* permet de tenir compte des différents paramètres et variables au moment du refroidissement.

La méthode *Recuit Simulé* utilise un programme de refroidissement fixe, dont la température baisse de façon monotone vers 0 (ou à une température proche de 0). Pour un cas réel, il paraîtrait raisonnable d'allonger la gamme sur laquelle les paramètres relativement insensibles sont cherchés. Par exemple, si la fonction objectif est assez insensible aux changements d'une variable x_i particulière, l'algorithme *Recuit Simulé Adaptatif* diminue le taux de refroidissement de la variable x_i . Cela permet d'explorer plus en détails la région pour cette variable particulière. Cela est

fait par un réglage périodique du temps de refroidissement, appelé reannealing [Ingber, 1989].

3.1.1.4 Méthode à multiples niveaux (*Clustering Methods*)

La méthode à multiples niveaux appartient à un groupe de méthodes stochastiques qui sont souvent appelées *Clustering Methods*. L'algorithme commence avec une recherche uniforme sur une région S prédéfinie, et crée ensuite des groupes de points proches correspondant à une région commune d'attraction. Un groupe C est un ensemble de points qui correspondent à une région d'attraction et qui ont les propriétés suivantes :

- C contient exactement un minimum local x^* .
- Tout algorithme de descente, qui commence avec un point $x \in C$ converge au point x^* .
- Les deux premières propriétés impliquent que C peut contenir exactement un point stationnaire.

La méthode a été proposée par [Rinnooy-Kan, 1984], dont la convergence globale est seulement probabiliste.

3.1.2 Méthodes déterministes

Les algorithmes déterministes sont différents des algorithmes stochastiques pour plusieurs raisons. Tout d'abord, les algorithmes stochastiques ne donnent une garantie à 100% que si les optimaux globaux sont localisés, et qu'ils peuvent les atteindre dans un temps fini. Deuxièmement, les algorithmes déterministes s'exécutent toujours en un temps fini (même si celui-ci peut être très long). Ils donnent un ensemble garanti (une partie d'une région), dans lequel les minimaux globaux sont contenus. Enfin, les algorithmes stochastiques ne peuvent pas garantir de trouver tous les minimums globaux d'une fonction : en général, ils ne peuvent en trouver qu'un lorsqu'il y en a plusieurs différents. Les algorithmes déterministes donnent la garantie que les optima globaux sont contenus dans les régions qui sont retournées à la fin de l'algorithme.

En général, une procédure d'optimisation dite globale pourrait trouver les optima globaux d'une fonction quelconque. Pour cela on suppose que les bornes initiales des variables sont données. Cependant, d'un point de vue théorique, il est aussi acceptable que les bornes soient infinies.

Une méthode générale et idéale d'optimisation globale déterministe est en quelque sorte une "boîte noire" capable de travailler avec la description d'une fonction, les contraintes associées au problème, et les liens entre les variables. La sortie de cette "boîte noire" donne lieu aux deux caractéristiques suivantes :

- Les valeurs de l'approximation de l'optimum global, sont obtenues avec une erreur maximale mathématiquement vérifiable.
- La valeur approximative de l'image de la fonction qui atteint les optima a une erreur bornée qui est aussi mathématiquement vérifiable.

Ces deux caractéristiques donnent comme résultat un ensemble qui contient de façon garantie tous les optimum globaux d'un critère donné.

Dans cette section nous n'essayons pas d'établir un guide complet de l'histoire de l'optimisation globale. Nous allons plutôt nous concentrer sur une méthode en particulier, présentée en [Leclerc, 1992]. Cependant, plusieurs autres méthodes ont été développées et une intéressante discussion bibliographique peut être trouvée dans [Van-Iwaarden, 1996], [Hansen, 1992] ou [Cherruault, 1999].

3.2 Une méthode d'optimisation globale par intervalles

Dans cette partie, nous considérons une variante de l'algorithme proposé par [Leclerc, 1992] pour résoudre le problème d'optimisation non-linéaire général suivant, lié au problème optimal (2.10) de la méthode d'estimation IMHSE :

$$\begin{aligned}
 & \text{Minimiser (globalement)} \quad V([x]) \\
 & [x] \in [\Omega] \\
 & \varphi([x]) \leq 0 \\
 & [\Omega] \subset \mathbb{IR}^n
 \end{aligned} \tag{3.1}$$

La formulation (3.1) répond à la terminologie suivante :

$V(\cdot)$: Fonction objectif ou critère à minimiser.

$[\Omega]$: Domaine admissible du problème sur \mathbb{IR}^n .

$[x]$: Etat borné (pavé) sur \mathbb{IR}^n .

$\varphi(\cdot)$: Ensemble de restrictions du problème (si disponibles).

L'algorithme fonctionne de manière itérative pour éliminer des parties de la boîte initiale $[\Omega]$, dans lesquelles il n'y a pas de valeurs qui minimisent globalement la fonction objectif $V(\cdot)$. Des parties de $[\Omega]$ sont éliminées quand elles ne contiennent pas le minimum global de la fonction ou quand ces parties ne vérifient pas l'ensemble des restrictions données par $\varphi(\cdot)$.

Si certaines parties de $[\Omega]$ ne sont pas éliminées, elles seront alors incorporées dans une liste. Cette liste contiendra toutes les sous boîtes de $[\Omega]$ dans lesquelles le minimum global de la fonction objectif peut être contenu. Par conséquent, l'union de tous les éléments contenus dans la liste, incluent tous les minimums globaux du critère $V(\cdot)$ qui doit être minimisé.

Dans la suite, nous allons considérer les aspects nécessaires pour formuler l'algorithme global d'optimisation par intervalles, en considérant un intervalle $[\Omega]$ comme domaine admissible.

- La largeur d'une boîte $w([x])$, est définie comme le plus long côté de cette boîte (Voir (2.1)).
- L'algorithme stocke dans une liste toutes les sous boîtes de $[\Omega]$ qui contiennent les minimums globaux (nous prenons en considération l'effet d'arrondi des calculs, de façon à obtenir des intervalles garantis qui contiennent la solution mathématique exacte). Par conséquent, l'union de toutes celles-ci contient tous les minimums globaux faisables de la fonction $V(\cdot)$ qui est minimisée.
- Le processus de recherche itérative de base s'arrête lorsque la largeur de la première boîte dans la liste est plus petite qu'une certaine tolérance prescrite, dénoté comme B_ε ($B_\varepsilon \in \mathbb{R}$).

Pour prendre en considération les aspects ci-dessus, nous allons utiliser les tests suivants pour éliminer les boîtes dans lesquelles le minimum global n'est pas contenu.

3.2.1 Faisabilité ou non-faisabilité

Soit $[x] \in \mathbb{IR}^n$. Nous pouvons alors tester si $[x]$ vérifie l'ensemble des restrictions $\varphi_1(\cdot), \varphi_2(\cdot), \dots, \varphi_r(\cdot)$ en utilisant l'arithmétique par intervalles, et grâce au test suivant :

Si pour toutes les $i^{\text{ème}}$ composantes, nous trouvons que $\varphi_i([x]) \subset]0 \text{ } +\infty[$, alors $[x]$ est sans aucun doute infaisable (il ne contient pas de points faisables) et peut donc

être effacé sans autres considérations. Dans tous les autres cas, nous ne pouvons rien dire (il y a peut-être des points faisables) et on devra garder la boîte $[x]$ dans la liste.

3.2.2 Le test du point central

Si $[x]$ est faisable, nous pouvons alors calculer l'image du centre de $[x]$ appelé $\beta_{[x]}$ avec la fonction objectif. C'est-à-dire, $V(\beta_{[x]}) = UV_{\beta_{[x]}}$ (Notez que $V([x]) \in \mathbb{IR}$ et $V(\beta_{[x]}) \in \mathbb{R}$). De plus, si V^* est la valeur minimale de $V(\cdot)$, nous pouvons être sûrs que $UV_{\beta_{[x]}} \geq V^*$, c'est-à-dire, que $UV_{\beta_{[x]}}$ est une borne supérieure pour la valeur minimale de $V(\cdot)$ sur dans le domaine admissible [Ichida et Fujii, 1979].

Si nous évaluons le critère $V(\cdot)$ dans une autre boîte faisable $[z] \in [\Omega]$ avec la même procédure, nous obtenons $V([z]) = [LV_{[z]} UV_{[z]}]$. Nous pouvons alors effectuer le test suivant :

Soit UV^* le plus petit $UV_{\beta_{[x]}}$ trouvé ; Si $LV_{[z]} > UV^*$, alors $[z]$ ne contient pas de candidats au minimum global dans $[\Omega]$. $[z]$ peut donc être effacé sans autres considérations.

En utilisant les deux tests présentés ci-dessus, il est possible de construire un algorithme simple pour minimiser globalement des critères différentiables et non-différentiables indistinctement [Leclerc, 1992]. Cependant il y a aussi la possibilité d'appliquer d'autres tests qui rendent l'algorithme d'optimisation globale plus performant. Nous pouvons trouver par exemple des tests qui prennent en compte la monotonie du critère ou les tests qui analysent la convexité de celui-ci. Ceci bien entendu, quand le critère à minimiser $V(\cdot)$ est différentiable.

3.2.3 Test de monotonie

Supposons une sous boîte $[x]$ faisable dans $[\Omega]$, et supposons que le gradient de $V(\cdot)$ est évalué en $[x]$.

Si pour une composante quelconque i , avec $i = 1, \dots, n$, nous trouvons que $0 \notin \nabla V_i([x])$, alors de toute évidence le gradient de $V(\cdot)$ ne s'annule pas en $[x]$. Si le gradient ne s'annule pas sur $[x]$, cette boîte peut être effacée, parce que le minimum global ne peut pas être contenu dans cette boîte.

3.2.4 Test de non-convexité

Supposons une sous boîte $[x]$ faisable dans $[\Omega]$. Si $[x]$ est une solution globale du problème d'optimisation (3.1), alors $V(\cdot)$ doit être convexe dans un voisinage de la solution. Autrement dit, le hessien de $V(\cdot)$ doit être non négatif sur $[x]$. Si nous pouvons montrer que $\nabla^2 V([x])$ n'est pas semi-défini positif sur $[x]$, alors cette boîte peut aussi être effacée.

Si $\nabla^2 V([x])$ est semi-défini positif sur $[x]$, cela implique que les éléments de la diagonale de $\nabla^2 V([x])$ sont des éléments non négatifs, c'est à dire :

$$\nabla^2 V_{ii}([x]) \geq 0 \quad \text{pour } i = 1, \dots, n$$

3.2.5 Test de Newton par intervalles

Nous pouvons aussi essayer d'éliminer une boîte ou au moins de la réduire, en utilisant la technique de Newton par intervalles [Kearfott, 1998].

Supposons une sous boîte $[x]$, $c([x])$ étant le centre de $[x]$ ($c([x]) \in \mathbb{R}^n$, voir (2.2)), et le critère $V(\cdot)$ du problème (3.1), nous pouvons alors définir l'opérateur de Newton par :

$$N(V(\cdot), [x], c([x])) = c([x]) - \frac{\nabla V(c([x]))}{\nabla^2 V([x])} \quad (3.2)$$

Notons que : $\nabla^2 V([x])$ est une matrice intervalle, et son inversion est faite en utilisant la technique de Gaus-Seidel ou élimination gaussienne par intervalles [Kearfott, 1998].

- a) Les minimums de $[V](\cdot)$ se trouvent inclus dans $N(V(\cdot), [x], c([x])) \cap [x]$
- b) Si $N(V(\cdot), [x], c([x])) \cap [x] = \{\emptyset\}$, alors il n'y a pas de solutions de $V(\cdot)$ sur $[x]$.
- c) Si $[x]$ contient des solutions de $V(\cdot)$, et si la largeur des composantes de $[x]$ sont suffisamment petites, la largeur de $N(V(\cdot), [x], c([x]))$ est proportionnelle au carré de la largeur des composants de $[x]$.
- d) Si $N(V(\cdot), [x], c([x])) \subset \text{Int}([x])$, où $\text{Int}([x])$ représente l'intérieur de $[x]$, alors il y a une solution unique pour $V(\cdot) = 0$.

3.2.6 L'algorithme G-Optimisation

Maintenant, un algorithme très simple minimisant globalement une fonction non-linéaire peut être formulé. Cet algorithme est détaillé ci-dessous :

- La "liste" de boîtes générées (gardées dans la "liste") est techniquement parlant une file d'attente.
- Les boîtes sont toujours ajoutées à la fin de la "liste" et sont retirées toujours du début de la "liste".
- Dans chaque itération, une boîte est retirée de la liste.
- La première boîte est toujours coupée en deux (bissectée) par son côté le plus large.
- Différents tests sont faits sur chaque demi boîte coupée ; si elle ne peut pas être effacée, elle est placée à la fin de la "liste".
- La première boîte est toujours la plus large, et si sa largeur est inférieure à la tolérance prescrite B_ε , alors la largeur de tout le reste des boîtes de la "liste" sera aussi plus petite ou égale que B_ε (critère d'arrêt).
- A chaque pas du processus, les boîtes contenues dans la "liste" contiennent tous les points qui minimisent globalement la fonction.
- Il est possible d'évaluer la complexité de l'algorithme en comptant le nombre d'opérations à effectuer et donc la durée des calculs (voir *section 3.2.7*).
- L'union des boîtes contenues dans la liste aura tous les points faisables qui minimisent globalement le critère $V(\cdot)$ sur $[\Omega]$.
- Plus la tolérance B_ε définie est petite, plus la capacité de la méthode à supprimer des minimum locaux de $V(\cdot)$ est grande sur $[\Omega]$.

Cette procédure est illustrée dans l'algorithme suivant :

Algorithme d'OPTIMISATION GLOBALE

Algorithme qui trouve une "liste" qui contient tous les points minimisant globalement un critère avec une tolérance donnée :

Algorithme 3 : G-Optimisation(.)

Entrées : $[\Omega], B_\varepsilon, V(\cdot), \varphi(\cdot)$
 $[\Omega]$: Domaine initial admissible sur \mathbb{R}^n .
 B_ε : Tolérance ou précision donnée d'optimisation.
 $V(\cdot)$: Critère à optimiser.
 $\varphi(\cdot)$: Ensemble des restrictions du problème (si disponibles).
 Sortie : Liste $\{[x]\}$, Liste de boîtes dans lesquelles le minimum global est contenu.

1. Ajouter $[\Omega]$ à la liste

Liste $\{[x]\} := \{[\Omega]\}$

2. Calculer le côté le plus large de la première boîte dans la liste

$d := w([\Omega])$ (Voir (2.1))

3. Calculer $V(\cdot)$ sur $[\Omega]$

$V([\Omega]) = [LV_{[\Omega]} UV_{[\Omega]}]$

$UV := UV_{[\Omega]}$ (Une borne supérieure de la valeur minimale de $V(\cdot)$)

4. Boucle principale

4.1 WHILE $d > B_\varepsilon$

4.2 Enlever la première boîte $[x]$ de la liste Liste $\{[x]\}$

4.3 Bissection

Couper en deux la boîte $[x]$ par son côté le plus large tel que $[x] = [x_1] \cup [x_2]$

4.4 Test de faisabilité de $[x_1]$ et $[x_2]$ (Si $\varphi(\cdot)$ est disponible)

Calculer $\varphi(\cdot)$ sur $[x_1]$

IF $\varphi([x_1]) \subset]0 +\infty[$ THEN effacer $[x_1]$ sans autres considérations

Calculer $\varphi(\cdot)$ sur $[x_2]$

IF $\varphi([x_2]) \subset]0 +\infty[$ THEN effacer $[x_2]$ sans autres considérations

4.6 IF Liste $\{[x]\}$ reste est vide, THEN RETURN 'Aucune solution dans $[\Omega]$ '

'Domaine $[\Omega]$ n'est pas faisable'

Fin du processus

4.7 Tests généraux : Déterminer si $[x_1]$ ou $[x_2]$ contiennent des minimums faisables

Test de Newton par intervalles

Déterminer s'il est possible d'éliminer la boîte $[x_1]$ ou de la réduire en utilisant : $[x_1] := N(V(\cdot), [x_1], c([x_1])) \cap [x_1]$

Calculer $V(\cdot)$ sur $[x_1]$ pour obtenir $V([x_1]) = [LV_{[x_1]} UV_{[x_1]}]$

Calculer $V(\cdot)$ sur $[x_1]$

Calculer $\nabla^2 V(\cdot)$ sur $[x_1]$

IF $LV_{[x_1]} > UV$ ou $0 \notin \nabla V_i([x_1])$ ou $\nabla^2 V_{ii}([x_1]) < 0$

THEN effacer $[x_1]$ sans autres considérations

ELSE ajouter $[x_1]$ à la fin de la Liste $\{[x]\}$

IF $[x_1]$ est listé

Calculer le point central $\beta_{[x_1]}$ de $[x_1]$ en $V(\cdot)$

IF $UV_{\beta_{[x_1]}} < UV$ THEN $UV := UV_{\beta_{[x_1]}}$

ENDIF

ENDIF

ENDIF

Test de Newton par intervalles

Déterminer s'il est possible d'éliminer la boîte $[x_2]$ ou de la réduire en utilisant: $[x_2] := N(V(\cdot), [x_2], c([x_2])) \cap [x_2]$

Calculer $V(\cdot)$ sur $[x_2]$ pour obtenir $V([x_2]) = [LV_{[x_2]} UV_{[x_2]}]$

Calculer $\nabla V(\cdot)$ sur $[x_2]$

Calculer $\nabla^2 V(\cdot)$ sur $[x_2]$

IF $LV_{[x_2]} > UV$ ou $0 \notin \nabla V_i([x_2])$ ou $\nabla^2 V_{ii}([x_2]) < 0$

THEN effacer $[x_2]$ sans autres considérations

ELSE ajouter $[x_2]$ à la fin de la Liste $\{[x]\}$

IF $[x_2]$ est listé

Calculer le point central $\beta_{[x_2]}$ de $[x_2]$ en $V(\cdot)$

IF $UV_{\beta_{[x_2]}} < UV$ THEN $UV := UV_{\beta_{[x_2]}}$

ENDIF

ENDIF

ENDIF

4.8 Calculer le côté le plus large de la première boîte $[x]$ dans la Liste $\{[x]\}$

$d := w([x])$

4.9 ENDWHILE

Remarque 11: *Le problème d'optimisation global (3.1) peut être aussi un problème sans restrictions. Dans ce cas la seule différence est que le test de faisabilité doit être éliminé (pas d'étape 4.4 dans l'algorithme précédent).*

Remarque 12: *D'un point de vue pratique, il est toujours possible de faire une normalisation du modèle utilisé (problème (3.1)) pour travailler plus facilement de façon numérique.*

Remarque 13: *Il est important de noter que cet algorithme d'optimisation global est valide aussi bien pour des critères $V(\cdot)$ différentiables que non-différentiables. Dans le cas de critères non-différentiables, les tests de monotonie, de non-convexité et de Newton ne sont pas utilisés.*

Remarque 14: *Une des caractéristiques importantes de cet algorithme, et des algorithmes d'optimisation global en général, est qu'avec une précision croissante d'optimisation, il éliminera tous les états (minimums locaux). Cependant, les états indiscernables restent dans la "liste" (voir section 2.2).*

Théorème : *Pour toutes valeurs positives de B_ϵ , l'algorithme d'optimisation global converge toujours avec certitude sur un domaine admissible, et résout le problème de l'optimisation (3.1).*

Preuve: *La démonstration est basée sur une recherche exhaustive de la solution, faite avec les techniques d'intervalles, sur tout le domaine admissible, et peut être trouvé pour ce type d'algorithme dans [Piazzzi et Visioli, 2000] ou [Piazzzi et Visioli, 1998].*

3.2.7 Complexité de l'algorithme d'optimisation global

Il est possible d'évaluer la complexité informatique d'un algorithme en comptant le nombre d'opérations à exécuter [Papadimitriou et Steiglitz, 1998]. Le temps de calcul peut ainsi être estimé. Il peut être démontré (voir annexe A) que la complexité de cet algorithme, pour un critère non différentiable et dans le pire des cas est donnée par :

$$O(f(n)) \approx O(L \cdot (n + nj + n\varphi)) \quad (3.3)$$

Où L est la longueur maximale de la Liste $\{[x]\}$ (nombre de boîtes), n correspond à la dimension du domaine admissible ou dimension de variable d'état ($[\Omega] \in \mathbb{I}\mathbb{R}^n$), nj est le degré de la fonction objectif, et $n\varphi$ le nombre de restrictions associées à la fonction objectif. Si nous considérons nj et $n\varphi$ comme constantes par rapport à L et n , la complexité est donnée par $O(f(n)) \approx O(L \cdot n)$.

Alors dans le pire des cas, le nombre de boîtes produites par λ divisions (bissection de la boîte initiale) et correspondant à la longueur L de la Liste $\{[x]\}$ est :

$$L = 2^{\sum_{i=1}^n \lambda_i} \quad (3.4)$$

Finalement la complexité informatique de l'algorithme est :

$$O(f(n)) \approx O\left(2^{\sum_{i=1}^n \lambda_i} \cdot n\right) \quad (3.5)$$

Remarque 15: *Limites de l'approche.* Cette complexité exponentielle montre que la technique d'optimisation globale peut être appliquée seulement à certains types de systèmes dynamiques. Bien que la complexité informatique de l'algorithme soit exponentielle, il ne faut pas oublier que cette complexité a été obtenue dans le cas le plus mauvais, c'est-à-dire quand le critère a une forme aplatie (le domaine admissible $[\Omega]$ tout entier est un minimum global). En d'autres termes, il y a une infinité de minimums globaux qui satisfont le critère pour toute précision d'optimisation. Cependant, ceci n'est seulement vrai que dans quelques cas de systèmes non-observables pour IMHSE. Par conséquent, la longueur L de la Liste $\{[x]\}$, est loin d'être une fonction exponentielle de la dimension du vecteur d'état.

De plus ces contraintes peuvent être réduites grâce à l'apparition de nouveaux processeurs, de programmes à base de codes compilés plus rapides. Par exemple, avec l'aide des langues informatiques tel que C++ [Jaulin et al, 2001] ou Java

[Thévenon et Flaus, 2000], et l'usage de techniques plus efficaces pour calculer les bornes des fonctions évaluées par intervalles.

Enfin, ce type d'algorithme d'optimisation peut être appliqué sans contraintes, quand la dimension de l'état n'est pas trop grande et quand le processus est lent. Ce qui est souvent le cas dans les bioprocédés industriels, tel que les procédés biotechnologiques.

3.3 Matlab et l'outil ia.jar

Pour l'implémentation de l'algorithme 3 d'optimisation globale nous ferons appel au logiciel Matlab pour sa souplesse et sa facilité de programmation. Cependant dans le cas d'un critère différentiable $V(\cdot)$ en (3.1), nous avons besoin des informations du gradient, du hessien par intervalles. Le but est d'implémenter les tests de monotonie (section 3.2.3), de non-convexité (section 3.2.4) et le test de Newton par intervalles (section 3.2.5) respectivement.

Les procédures propres de l'arithmétique par intervalles (section 2.1.1), ainsi que le gradient $\nabla V(\cdot)$ et le hessien $\nabla^2 V(\cdot)$ par intervalles seront évalués à l'aide d'un outil appelé "ia.jar" (Interval Arithmetic). Il est développé en langage JAVA et couplé à MATLAB version 6.1.

3.3.1 Définition d'une fonction avec "ia.jar"

Considérons la fonction suivante, et déclarons-la en utilisant "ia.jar", afin de manipuler les variables comme des intervalles sur un domaine admissible donné.

$$\begin{aligned}
 V(x_1, x_2) &= \frac{1}{2} \sum_{j=1}^2 (x_j^4 - 16 \cdot x_j^2 + 5 \cdot x_j) \\
 x_i &\in [\Omega] \\
 [\Omega] &\subset \mathbb{R}^n
 \end{aligned}
 \tag{3.6}$$

Exemple 3.1: La déclaration de (3.6) sera faite dans un fichier MATLAB d'extension ".m" défini comme suit :

```

import ia.*
import SExp.*
global sCrit
sCrit=SymbolicCriterion;
sCrit.addSExp('V:=0.5*(x[1]^4-16*x[1]^2+5*x[1])+0.5*(x[2]^4-
16*x[2]^2+5*x[2])');
sCrit.setNameOfVariable('x',2);
sCrit.setNameOfFunction('V');

```

A la suite de la déclaration de l'expression qui définit la fonction et ses variables, nous devons identifier le nom de la fonction 'V', et celle de la variable 'x'. Les résultats seront obtenus pour une boîte donnée $[x] \in [\Omega]$.

Exemple 3.2 : L'évaluation de $[x]$ et par la suite, l'obtention du gradient et du hessien seront faites en utilisant la déclaration de la fonction suivante dans un fichier ".m".

```

function [r,ri,G,H]=IV(a)
import ia.*
global sCrit
[n,m]=size(a);
r=sCrit.eval(midhpoint(a));
xxi=javaArray('ia.Interval',n);
for j=1:n
    xxi(j)=ia.Interval(a(j,1),a(j,2));
end
grad=javaArray('ia.Interval',n);
hess=javaArray('ia.Interval',n,n);
G=IntervalVector(n);
H=IntervalMatrix(n,n);
X=IntervalVector(xxi);
ri=sCrit.eval(X,G,H);

```

Ici "a" est une boîte, c'est-à-dire : $a = [x]$. Cette fonction donnera comme résultats les valeurs r, ri, G et H. Qui sont respectivement la valeur du critère $V(\cdot)$ évalué dans le point central de la boîte "a" ($r \in \mathbb{R}$), le critère $V(\cdot)$ évalué par intervalles ($ri \in \mathbb{IR}$), le gradient du critère $V(\cdot)$ par intervalles ($G \in \mathbb{IR}^n$) et le hessien du critère $V(\cdot)$ par intervalles ($H \in \mathbb{IR}^{n \times n}$).

Exemple 3.3 : L'évaluation en utilisant MATLAB de la fonction de l'exemple 3.2, sur une boîte $[x]$ tel que $[x] = [-3.0, -2.5] \times [-3.0, -2.5]$ nous donnera :


```

>> a=[-3.0 -2.5; -3.0 -2.5];
>> [r,ri,G,H]= IV (a)
r =
  -77.55859375000000
ri =
  [-119.9375, -31.5]
G =
  { [-11.5, 19.25], [-11.5, 19.25] }
H =
  { [21.5, 38.0] [0.0, 0.0] ; [0.0, 0.0] [21.5, 38.0] }
>>

```

Notons le pessimisme qu'il y a entre le résultat obtenu par "ri" par rapport à celui de "r". De plus $0 \in G$ et $H_{ii} > 0$, ce qui confirme qu'un minimum se trouve dans cette boîte $[x]$.

Remarque 16: *Les manipulations des intervalles seront faites en général en utilisant le logiciel MATLAB de manière classique, et toujours à l'aide de l'outil ia.jar développé en JAVA.*

Exemple 3.4 : Soit le domaine admissible $[\Omega] = [-4, 4] \times [-4, 4]$, et une précision donnée $B_\varepsilon = 0.05$ pour le problème d'optimisation globale sans contraintes suivant :

$$\begin{aligned} \text{Minimiser (Globalement)} \quad V(x_1, x_2) &= \frac{1}{2} \sum_{j=1}^2 (x_j^4 - 16 \cdot x_j^2 + 5 \cdot x_j) \\ x_j &\in [\Omega] \end{aligned} \quad (3.7)$$

Les solutions sur \mathbb{R}^2 , dont le gradient $V(\cdot)$ s'annule pour quatre minimums sont :

1. $x_1^* = (2.746, 2.746)$
2. $x_2^* = (2.746, -2.903)$
3. $x_3^* = (-2.903, 2.746)$
4. $x_4^* = (-2.903, -2.903)$

La solution globale au problème par intervalles, avec $B_\varepsilon = 0.05$ est présentée ci-dessous. La solution graphique est présentée dans l'annexe B.

$$[x]^* = \begin{bmatrix} [-2.9375 & -2.8750] \\ [-2.9375 & -2.8750] \end{bmatrix}$$

3.4 Optimisation globale avec précision de sortie bornée

Nous allons maintenant considérer la formulation du problème (3.1). Cependant nous cherchons maintenant à caractériser un ensemble de solutions en $[\Omega]$, qui puisse contenir tous les pavés $[x]$, tel que l'image issue de ces sous pavés soit plus petite qu'un seuil donné τ . Nous ne cherchons pas à minimiser infiniment l'écart des bornes sur l'image de $V(\cdot)$, mais à rester dans la borne de tolérance τ donnée, c'est-à-dire telle que $V(\cdot) \leq \tau$. Nous cherchons donc un ensemble $[x]$ qui minimise le problème d'optimisation globale (3.1) tel que :

$$[x] : \{[x] / \forall [x_i] \in [\Omega], V([x]) \leq \tau\}$$

La reformulation du problème (3.1) nous donne :

$$\begin{aligned} & \text{Minimiser (globalement)} \quad V([x]) \\ & [x] \in [\Omega] \\ & \varphi([x]) \leq 0 \\ & V([x]) \leq \tau \\ & [\Omega] \subset \mathbb{I}\mathbb{R}^n, \tau \in [0, 100]\% \end{aligned} \tag{3.8}$$

Pour résoudre le problème ci-dessus, nous allons modifier l'algorithme d'optimisation décrit dans la section 3.2.6 (Algorithme G-Optimisation(.)). De cette façon, notre algorithme pourra d'un côté optimiser globalement un critère quelconque, mais aussi chercher les ensembles sur le domaine admissible tels que leur image reste à l'intérieur d'une tolérance donnée.

Deux listes seront maintenant utilisées, d'une part la liste classique (Liste $\{[x]\}$) utilisée pour résoudre le problème (3.1), et d'autre part une liste appelée Liste-2 $\{[x]\}$ et qui sert à garder les boîtes dont l'image vérifie la tolérance spécifiée. Remarquons maintenant qu'un autre paramètre (τ) est nécessaire à l'entrée de notre algorithme modifié, et que la sortie est maintenant composée de deux listes de boîtes.

Algorithme 3 : G-Optimisation(.)

Entrée : $[\Omega], B_\varepsilon, V(\cdot), \varphi(\cdot), \tau$

$[\Omega]$: Domaine initial admissible sur $\mathbb{I}\mathbb{R}^n$.

B_ε : Tolérance ou précision donnée d'optimisation.
 $V(\cdot)$: Critère à optimiser.
 $\varphi(\cdot)$: Ensemble de restrictions du problème (si disponibles).
 τ : Tolérance de l'image bornée

Sortie:

Liste $\{[x]\}$: Liste de boîtes où le minimum global est contenu.

Liste-2 $\{[x]\}$: Liste de boîtes qui vérifient la tolérance τ .

La partie à modifier dans l'algorithme 3 est la section 4.6. Nous indiquons uniquement les modifications pour les tests sur la boîte $[x_1]$, cependant il faudrait effectuer les mêmes modifications pour les tests sur la boîte $[x_2]$.

Algorithme 3 : G-Optimisation(\cdot)

⋮

4.7 Tests généraux : Déterminer si $[x_1]$ ou $[x_2]$ contiennent minima faisables

Test de Newton par intervalles

Déterminer s'il est possible d'éliminer la boîte $[x_1]$ ou de la réduire en utilisant : $[x_1] := N(V(\cdot), [x_1], c([x_1])) \cap [x_1]$

Calculer $V(\cdot)$ sur $[x_1]$ pour obtenir $V([x_1]) = [LV_{[x_1]} UV_{[x_1]}]$

Calculer $\nabla V(\cdot)$ sur $[x_1]$

Calculer $\nabla^2 V(\cdot)$ sur $[x_1]$

IF $Sup(V([x_1])) \leq \tau$

 Ajouter $[x_1]$ à la fin de la Liste-2 $\{[x]\}$

END

IF $LV_{[x_1]} > UV$ ou $0 \notin \nabla V_i([x_1])$ ou $\nabla^2 V_{ii}([x_1]) < 0$

 THEN effacer $[x_1]$ sans autres considérations

 ELSE ajouter $[x_1]$ à la fin de la Liste $\{[x]\}$

 IF $[x_1]$ est listé

 Calculer le point central $\beta_{[x_1]}$ de $[x_1]$ en $V(\cdot)$

 IF $UV_{\beta_{[x_1]}} < UV$ THEN $UV := UV_{\beta_{[x_1]}}$

 ENDIF

 ENDIF

ENDIF

⋮

Avec ces modifications de l'algorithme, certaines boîtes sont stockées dans la liste2 (Liste-2 $\{[x]\}$), mais elles peuvent encore exister dans la liste1 d'optimisation (Liste $\{[x]\}$). Finalement la liste1 contiendra toujours les optimums globaux du critère, tandis que la liste2 contiendra les états qui respectent la tolérance. L'algorithme s'arrêtera encore avec le critère donné en la section 3.2.6. Dans ce cas et si \mathbb{L}_1 est l'ensemble union des états contenu dans la liste1, et si \mathbb{L}_2 est l'ensemble union des états contenu dans la liste2, alors $\mathbb{L}_1 \subseteq \mathbb{L}_2$. Autrement dit \mathbb{L}_1 est une borne interne pour \mathbb{L}_2 .

Une autre variante possible et plus performante est celle qui supprime de la liste liste1 la boîte si celle-ci est déjà contenue dans la liste liste2. Dans ce cas l'algorithme s'arrêtera quand la liste1 sera vide. Autrement dit, toutes les boîtes qui minimisent globalement le critère seront déjà placées dans la liste2, parce qu'elles vérifient la tolérance donnée. La modification à effectuer est la suivante. Cependant il faudrait effectuer les mêmes modifications pour les test sur la boîte $[x_2]$.

Algorithme 3 : G-Optimisation(.)

⋮

4.7 Tests généraux : Déterminez si $[x_1]$ ou $[x_2]$ contiennent minima faisables

Test de Newton par intervalles

Déterminer s'il est possible d'éliminer la boîte $[x_1]$ ou de la réduire en utilisant : $[x_1] := N(V(\cdot), [x_1], c([x_1])) \cap [x_1]$

Calculer $V(\cdot)$ sur $[x_1]$ pour obtenir $V([x_1]) = [LV_{[x_1]} UV_{[x_1]}]$

Calculer $\nabla V(\cdot)$ sur $[x_1]$

Calculer $\nabla^2 V(\cdot)$ sur $[x_1]$

IF $Sup(V([x_1])) \leq \tau$

THEN Ajouter $[x_1]$ à la fin de la Liste-2 $\{[x]\}$

ELSE

IF $LV_{[x_1]} > UV$ ou $0 \notin \nabla V_i([x_1])$ ou $\nabla^2 V_{ii}([x_1]) < 0$

THEN effacer $[x_1]$ sans autres considérations

ELSE ajouter $[x_1]$ à la fin de la Liste $\{[x]\}$

IF $[x_1]$ est listé

Calculer le point central $\beta_{[x_1]}$ de $[x_1]$ en $V(\cdot)$

IF $UV_{\beta_{[x_1]}} < UV$ THEN $UV := UV_{\beta_{[x_1]}}$

ENDIF

```

ENDIF
ENDIF
ENDIF
:

```

Regardons maintenant quelques résultats graphiques sur l'exemple suivant. En noir sont montrées les boîtes qui ont été incluses dans la liste1 (ensemble \mathbb{L}_1), et en blanc les boîtes de la liste2.

Exemple 3.5: Il s'agit du même exemple 3.4, avec une précision d'optimisation B_ε et une tolérance τ donnée, pour appliquer la méthode proposée.

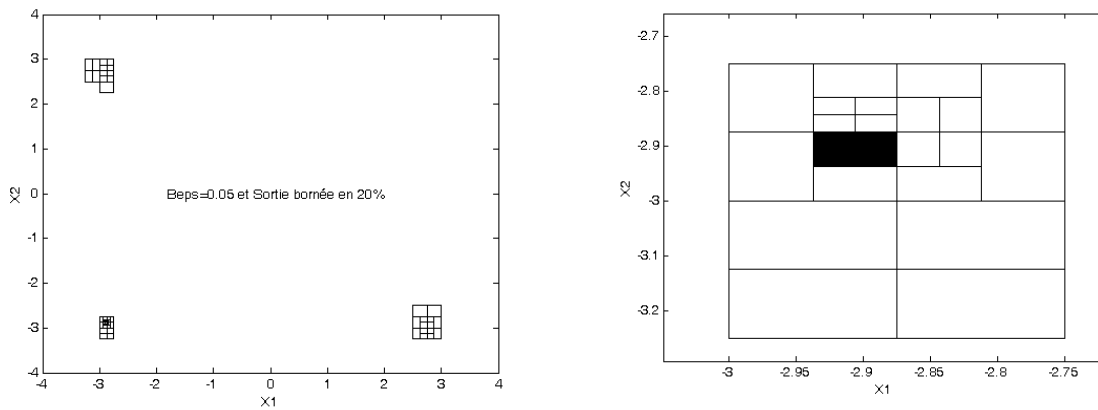


Figure 3.1 : Solution avec $B_\varepsilon = 0.05$ et $\tau = 20\%$

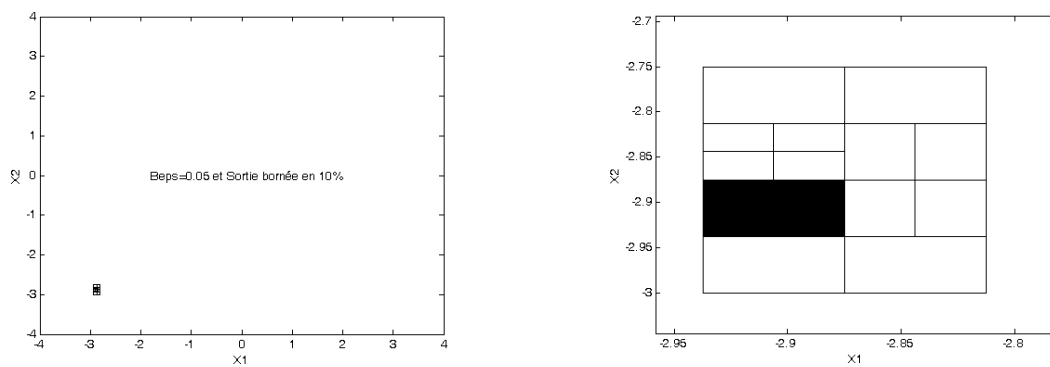


Figure 3.2 : Solution avec $B_\varepsilon = 0.05$ et $\tau = 10\%$

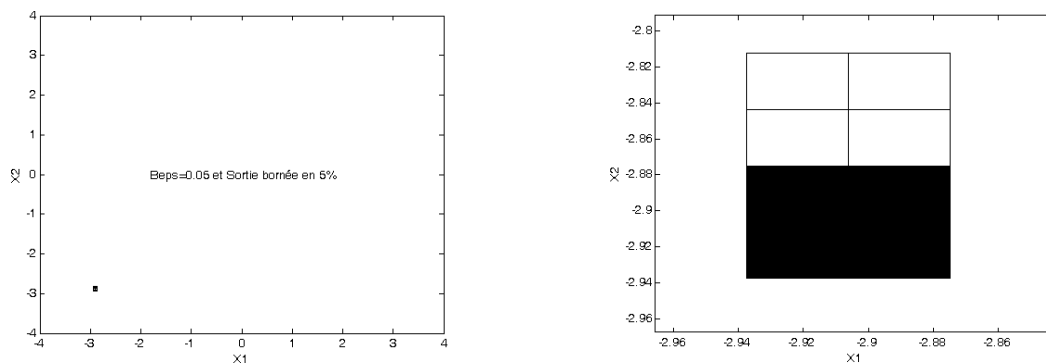


Figure 3.3 : Solution avec $B_\varepsilon = 0.05$ et $\tau = 5\%$

3.5 L'outil "ia.jar" et le problème optimal (2.10) sur un horizon de temps fini

Tout ce qui a été présenté dans ce chapitre, vise à résoudre le problème d'optimisation global (3.1) et par conséquent le problème d'estimation optimal sur un horizon de temps prédéfini, comme en (2.10). Cependant il nous reste à manipuler un système dynamique de la forme (1.1) avec *ia.jar* et à gérer son évolution sur un horizon de temps prédéfini avec intervalles.

Ce problème se révèle d'une grande importance pour notre méthode. En effet, nous avons besoin d'évaluer les trajectoires du système dynamique du type (1.1) dans l'étape 4 de l'algorithme 1. Mais nous avons aussi à évaluer plusieurs fois ces trajectoires dans l'algorithme 3, quand le problème général (3.1) est en réalité le problème d'optimisation (2.10). Ainsi, lors de l'étape 4.7 de l'algorithme 3, nous aurons besoin de calculer sur une boîte $[x]$ quelconque, la valeur du critère $V(\cdot)$, celui du gradient $\nabla V(\cdot)$, celle du hessien $\nabla^2 V(\cdot)$ et aussi la valeur de l'opérateur de Newton $N(V(\cdot), [x], c([x]))$. Dans tous ces cas, le critère $V(\cdot)$ sera celui présenté en (2.12).

Exemple 3.6 : Considérons le système non-linéaire suivant, dont les variables d'état sont X_k et S_k . μ_k est un paramètre et la sortie est donnée par y_k . Déclarons ce système avec *ia.jar*.

$$\begin{aligned} X_{k+1} &= X_k \cdot (1 + 0,1 \cdot \mu_k) \\ S_{k+1} &= S_k - 0,1 \cdot \frac{\mu_k}{0,8} \cdot X_k \\ \mu_k &= 0,4 \cdot \frac{S_k}{5 + S_k} \\ y_k &= (\mu_k + 0.01) \cdot X_k \end{aligned}$$

La représentation du système est faite, en utilisant la déclaration suivante dans un fichier ".m".

```
import ia.*
import SExp.*
global sCrit
sCrit=SymbolicCriterion;
sl=SExpList;
sl.add('u[1]:=0.4*x[2]/(5+x[2])');
sl.add('p1[k]:=x[1]*(1+0.1*u[1])');
sl.add('p2[k]:=x[2]-0.1*u[1]*x[1]/0.8');
sl.add('y[k]:=(u[1]+0.01)*x[1]');
sl.add('e2[k]:=(ym[k]-y[k])^2');
sl.add('J:=J+e2[k]');
sl.add('x[1]:=p1[k]');
sl.add('x[2]:=p2[k]');
```

Remarquons que le critère $V(\cdot)$ donné en (2.12) est remplacé par "J" dans la représentation ci-dessus.

Exemple 3.7 : L'expansion du système, représentée à l'aide de *ia.jar* dans le dernier exemple, sur un horizon de temps lh , est donné par l'instruction : `sl.expand('k',1,lh)`, incluse aussi dans un fichier ".m". Ici $lh=10$, donc :

```
slc=sl.expand('k',1,10);
```

Ce qui nous donne :

```

slc =
#0 u[1]:=(0.4*x[2])/(5+x[2])
#1 p1[1]:=(x[1]*(1+(0.1*u[1])))
#2 p2[1]:=(x[2]-((0.1*u[1])*x[1])/0.8)
#3 y[1]:=(u[1]+0.01)*x[1]
#4 e2[1]:=pow((ym[1]-y[1]),2)
#5 J:=(J+e2[1])
#6 x[1]:=p1[1]
#7 x[2]:=p2[1]
#8 u[1]:=(0.4*x[2])/(5+x[2])
#9 p1[2]:=(x[1]*(1+(0.1*u[1])))
#10 p2[2]:=(x[2]-((0.1*u[1])*x[1])/0.8)
#11 y[2]:=(u[1]+0.01)*x[1]
#12 e2[2]:=pow((ym[2]-y[2]),2)
#13 J:=(J+e2[2])
#14 x[1]:=p1[2]
#15 x[2]:=p2[2]
#16 u[1]:=(0.4*x[2])/(5+x[2])
#17 p1[3]:=(x[1]*(1+(0.1*u[1])))
#18 p2[3]:=(x[2]-((0.1*u[1])*x[1])/0.8)
#19 y[3]:=(u[1]+0.01)*x[1]
#20 e2[3]:=pow((ym[3]-y[3]),2)
#21 J:=(J+e2[3])
#22 x[1]:=p1[3]
#23 x[2]:=p2[3]
:
:
#64 u[1]:=(0.4*x[2])/(5+x[2])
#65 p1[9]:=(x[1]*(1+(0.1*u[1])))
#66 p2[9]:=(x[2]-((0.1*u[1])*x[1])/0.8)
#67 y[9]:=(u[1]+0.01)*x[1]
#68 e2[9]:=pow((ym[9]-y[9]),2)
#69 J:=(J+e2[9])
#70 x[1]:=p1[9]
#71 x[2]:=p2[9]
#72 u[1]:=(0.4*x[2])/(5+x[2])
#73 p1[10]:=(x[1]*(1+(0.1*u[1])))
#74 p2[10]:=(x[2]-((0.1*u[1])*x[1])/0.8)
#75 y[10]:=(u[1]+0.01)*x[1]
#76 e2[10]:=pow((ym[10]-y[10]),2)
#77 J:=(J+e2[10])
#78 x[1]:=p1[10]
#79 x[2]:=p2[10]

```

Remarquons que $p1[10]$ et $p2[10]$ ont la valeur de l'état courant en fin d'horizon.

Pour ce cas particulier, l'affectation des variables pour évaluer le système sur l'horizon de temps choisi peut être faite comme suit (fichier ".m") :

```
[n,m]=size(Boite);  
sCrit.setNameOfVariable('x',n);  
sCrit.setNameOfFunction('J');  
sCrit.add1DArrayIdentifler('p1',lh);  
sCrit.add1DArrayIdentifler('p2',lh);  
sCrit.add1DArrayIdentifler('u',1);  
sCrit.add1DArrayIdentifler('y',lh);  
sCrit.add1DArrayIdentifler('ym',lh);  
sCrit.add1DArrayIdentifler('e2',lh);
```

3.6 Conclusions

Dans ce chapitre, nous avons présenté une méthode d'optimisation globale par intervalles pour la résolution du problème optimal d'estimation IMHSE (2.10). Cette étude a été faite en utilisant un algorithme d'optimisation existant. Nous avons apporté des modifications pour prendre en compte une sortie bornée, dans le cadre du problème d'estimation sur un horizon de temps fini.

Dans un premier temps, nous avons étudié les caractéristiques propres d'une boîte noire qui optimisent globalement un critère. Ensuite, nous avons exposé l'algorithme d'optimisation globale en détail, ainsi que sa complexité informatique.

Nous avons aussi présenté un outil développé en JAVA et couplé au logiciel Matlab pour la description d'un système dynamique. Des exemples illustratifs ont été proposés, pour mettre en évidence l'avantage de cet outil.

Cet outil, ainsi que les modifications apportées à l'algorithme d'optimisation globale le rendant plus performant, vont permettre la réduction d'un domaine admissible pour le problème d'estimation IMHSE, connaissant les contraintes liées au système estimé.

Notons que cette nouvelle formulation présentée vise à caractériser un ensemble de solutions sur un domaine admissible. Cet ensemble peut contenir tous les pavés, tels que l'image issue de ces sous pavés respecte une tolérance voulue. Nous ne cherchons donc pas à minimiser infiniment l'écart entre les bornes de l'image du critère, mais à rester sous un seuil de tolérance donné.

Chapitre 4

Observabilité numérique : Approche par intervalles

Nous allons présenter dans ce chapitre les concepts associés à l'étude de l'observabilité numérique par intervalles pour des systèmes non-linéaires. Nous avons développé ces concepts pour des systèmes dont les modèles sont décrits en utilisant une représentation par intervalles de leurs états.

Notre approche peut être vue comme une extension des concepts classiques d'observabilité non-linéaire, appelée dorénavant ε -observabilité. Cette observabilité est basée sur une notion de voisinage d'un état avec une précision epsilon donnée. Pour cela, nous allons définir les concepts de ε -observabilité et ε -indistinguabilité.

Pour étudier l'observabilité par intervalles, nous proposons la notion de trajectoire indiscernable par intervalle. Celle-ci prend en compte l'ensemble solution du problème optimal (2.10) de la méthode d'estimation IMHSE. Cette méthode d'estimation est à la base de la construction de la notion d'observabilité qui sera présentée. Elle est plus proche des méthodes numériques que des méthodes de description formelle [Boillereaux, 1996], [Boillereaux et Flaus, 2000], [Valdés-González et Flaus, 2002a].

Nous présenterons tout d'abord la notion de ε -indistinguabilité au sens des intervalles. Puis nous présentons l'interprétation graphique de cette proposition. Nous examinons dans une deuxième partie la notion de ε -observabilité et les différents types de cas qui peuvent être rencontrés.

4.1 États indistinguables

Soient \mathbb{V} et \mathbb{B} des sous-ensembles de \mathbb{IR}^n , $\mathbb{B} \subset \mathbb{V}$ (cf. figure 4.1). \mathbb{B} est l'ensemble de tous les états solution (vecteurs intervalle), du problème optimal global sur un horizon (2.10). Cet ensemble solution est découpé avec une précision donnée epsilon. Pour simplifier, et sans perte de généralité, les coupes seront considérées régulières sur le domaine admissible pour les figures suivantes.

- $x_i \in [x_i], x_i \in \mathbb{R}, [x_i] \subset \mathbb{B}, \mathbb{B} \subset \mathbb{V}, \mathbb{V} \subset \mathbb{IR}^n$
- $\text{Max}\{w([x_i])\} \leq \varepsilon$

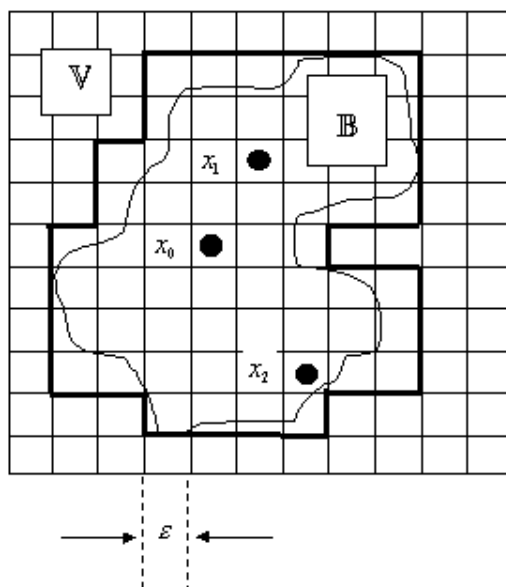


Figure 4.1 : Ensemble d'états intervalles en \mathbb{B} indistinguables

Proposition 1. (ε -indistinguabilité): Une paire de points x_1 et x_2 notée $I_\varepsilon(x_1, x_2)$, est dite indiscernable avec une précision ε fixe ($\varepsilon \in \mathbb{R}$), si la norme de l'écart de la trajectoire de sortie du système (en partant de ces points) est bornée par α ($\alpha \in \mathbb{R}$), quand une commande externe u est appliquée au système. En d'autres termes x_1 et x_2 sont ε -indistinguables quand x_1 et x_2 produisent la même sortie indiscernable par intervalles pour toute entrée admissible. Ainsi, pour un système du type (1.1) dont $(\alpha \wedge \varepsilon) \in \mathbb{R}$, $(x_1 \wedge x_2) \in \mathbb{R}^n$, on a :

$$I_\varepsilon(x_1, x_2) \Leftrightarrow \forall \varepsilon \geq 0, \exists \alpha \geq 0 / \|y(k/x_1) - y(k/x_2)\| < \alpha \quad (4.1)$$

Cette proposition est illustrée graphiquement dans la figure 4.2 pour un horizon de temps prédéfini.

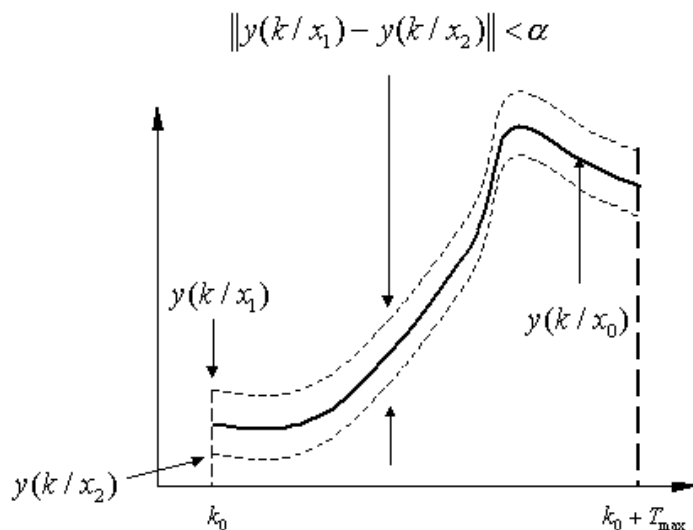


Figure 4.2 : α -voisinage sur la sortie du système

Proposition 2. (ε -observabilité) : Soient $\alpha, \varepsilon, \delta \in \mathbb{R}$ et $x_0 \in \mathbb{R}^n$. Le système (1.1) est dit observable par intervalles, quand la proposition suivante est vraie :

$$\forall \delta > \varepsilon, \exists \alpha \geq 0 / \|y(k/x_i) - y(k/x_0)\| < \alpha \Rightarrow \|x_i - x_0\| \leq \delta \quad (4.2)$$

Remarque 17: Un système est observable classiquement en \mathbb{R}^n [Hermann et Krener, 1977], quand il est dit 0-observable.

$$\|y(k/x_i) - y(k/x_0)\| \equiv 0 \Rightarrow x_i \equiv x_0 \quad (4.3)$$

La proposition sur l' ε -observabilité est illustrée dans les figures 4.3 et 4.4.

Remarque 18: Notons que dans certains cas un effort de calcul considérable peut être nécessaire pour distinguer des points sur un domaine admissible. Quand c'est le cas, ces points sont appelés "points discernables faiblement". En théorie, ce type de cas existe, mais d'un point de vue pratique (en particulier par intervalles), ces types d'états peuvent être considérés comme des points indiscernables pour une précision epsilon donnée.

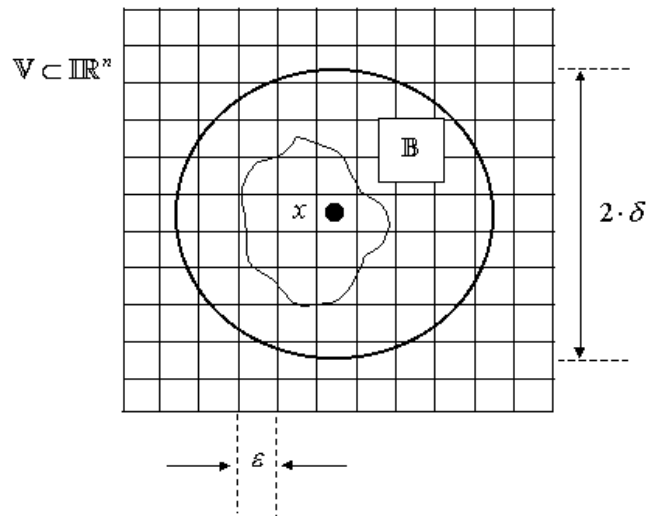


Figure 4.3 : δ -voisinage d'un état x sur un domaine admissible

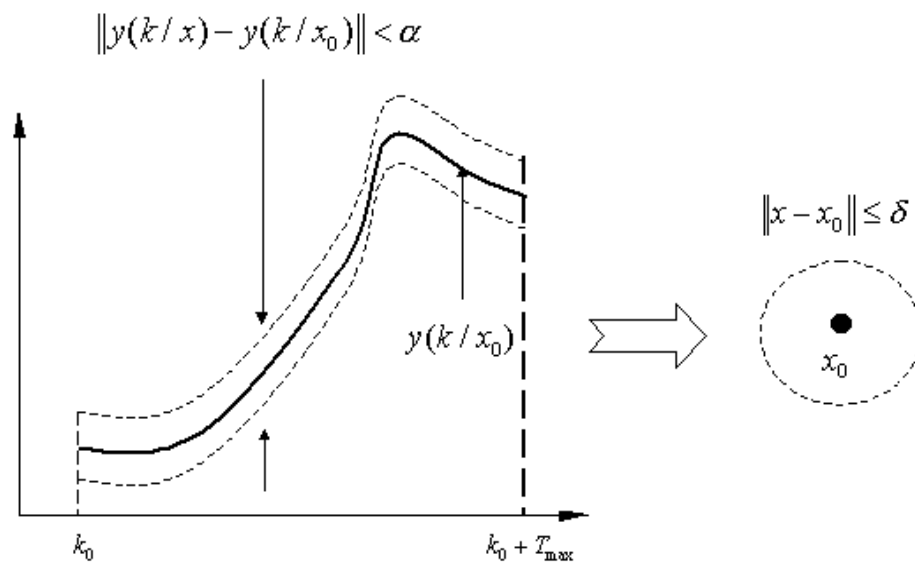


Figure 4.4 : Trajectoire de sortie α -bornée et un δ -voisinage d'un état x_0

4.2 Réduction de la précision ε autour de la référence

4.2.1 Systèmes observables

La propriété 2 sur l' ε -observabilité implique que, si ε devient de plus en plus petit, la trajectoire de sortie se contracte vers une plus petite trajectoire α -bornée, toujours contenue dans la zone de la trajectoire initiale. De plus, le voisinage autour x_0 se contracte aussi vers un δ -voisinage plus petit, tel que $\delta \geq \varepsilon$. En d'autres termes, si $\varepsilon_1 \gg \varepsilon_2 \Rightarrow \alpha_1 \gg \alpha_2 \wedge \delta_1 \gg \delta_2$. Graphiquement, nous avons :

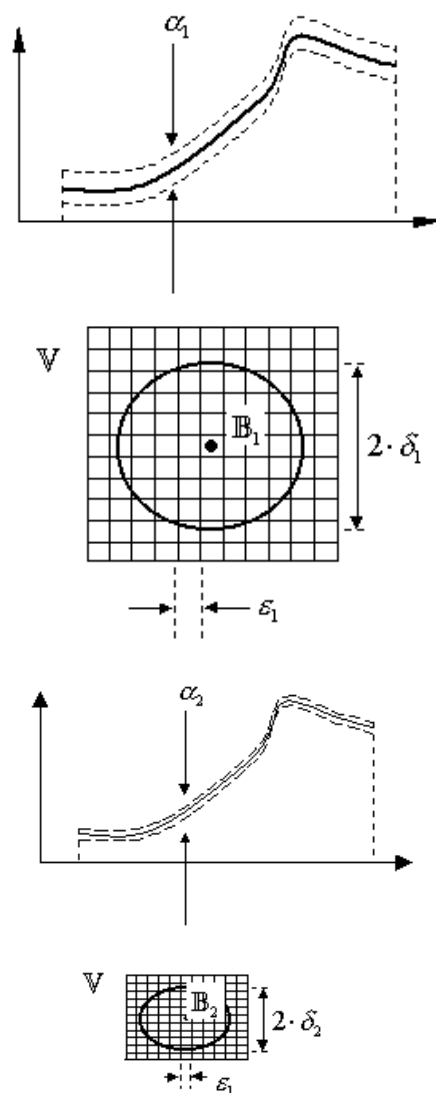


Figure 4.5 : Visualisation graphique de la propriété de ε -observabilité

4.2.2 Systèmes non observables : Cas 1

Soit un ensemble \mathbb{B} solution du problème (2.10), tel que \mathbb{B} contient deux sous-ensembles indiscernables différents. Le système représenté est un système non-observable (voir définition 1, section 2.6.1). En effet, il y a deux sous-ensembles différents sur le domaine admissible (dans le cas de nombres réels il y aura deux points), qui donnent la même trajectoire de sortie bornée indiscernable (proposition 1). Remarquons que la largeur de l'ensemble \mathbb{B}_2 est semblable à la largeur de l'ensemble \mathbb{B}_1 même si la précision d'optimisation pour le problème (2.10) a été réduite radicalement. En d'autres termes :

Si $\varepsilon_1 \gg \varepsilon_2 \Rightarrow \alpha_1 \gg \alpha_2 \wedge \delta_1 \approx \delta_2$. Cela peut être aussi représenté graphiquement comme dans la figure 4.6.

4.2.3 Systèmes non observables : Cas 2

Ce cas représente un système non-observable avec un nombre infini de solutions pour le problème optimal (2.10), c'est-à-dire un système avec un sous-ensemble \mathbb{B} contenant beaucoup d'états indiscernables différents. Comme dans le cas 1, la largeur de l'ensemble \mathbb{B}_2 est semblable à la largeur de l'ensemble \mathbb{B}_1 (même ordre de magnitude). Ceci même si la précision d'optimisation pour le problème (2.10) a été réduite radicalement. Nous avons ici aussi une relation du type :

Si $\varepsilon_1 \gg \varepsilon_2 \Rightarrow \alpha_1 \gg \alpha_2 \wedge \delta_1 \approx \delta_2$. Voir figure 4.7.

4.3 Indicateur de qualité ou indice d'observabilité

Dans un système dit ε -observable, l'élément principal de ce formalisme est de savoir s'il existe une amplitude suffisante du lien entre les états pour les distinguer, ceci afin d'autoriser une estimation des états par intervalles.

Nous cherchons donc à proposer un indicateur, appelé ici *indice d'observabilité numérique*, et qui est noté I_{NO} . La caractéristique principale de cet indicateur est de fournir la largeur du voisinage d'états (sous-ensemble \mathbb{B}) sur le domaine admissible \mathbb{V} . En d'autres termes, on peut utiliser cet indicateur pour faire des estimations avec prudence. Cet indicateur est calculé pour chaque variable d'état δ -borné, obtenus pour une marge α sur la trajectoire de sortie (précision de la mesure).

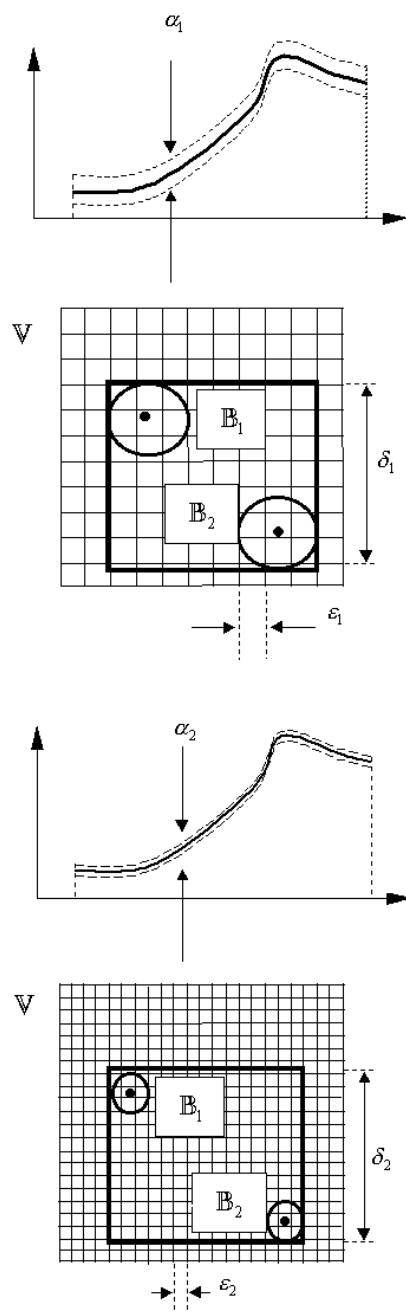


Figure 4.6 : Visualisation graphique d'un système non observable : cas 1

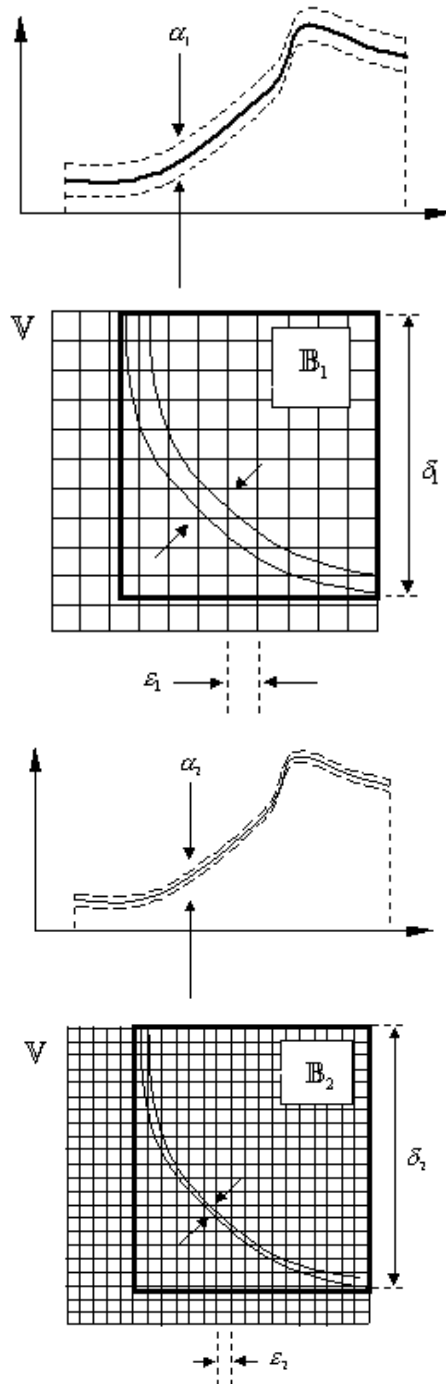


Figure 4.7 : Visualisation graphique d'un système non observable : cas 2

En d'autres termes et d'après (4.2) :

$$\forall \delta_i > \varepsilon, \exists \alpha \geq 0 / \|y(k/x_i) - y(k/x_0)\| < \alpha \Rightarrow \|x - x_0\| \leq \delta_i \quad (4.4)$$

Dans ce contexte, les propositions suivantes peuvent être données.

Proposition 3. (Indice d'observabilité numérique) : L'indicateur est défini pour chaque variable d'état du vecteur d'état estimé $[x]$ en début de l'horizon en considérant (4.4). Cet indice est calculé pour la largeur δ_i obtenu pour une marge α de la trajectoire de sortie. C'est-à-dire, qu'il indique s'il existe une amplitude suffisante du lien entre les états pour permettre une estimation

$$\pi = \frac{\alpha}{\max(\delta_i)} = \frac{\alpha}{w([x])} \quad (4.5)$$

Proposition 4. (Facteur d'amplitude) : Il est défini comme le rapport entre la largeur de la variable d'état examiné (du vecteur d'état estimé $[x]$) et l'amplitude du domaine admissible prédéfini \mathbb{V}_i pour chaque variable d'état. Le facteur d'amplitude normalisé est donné par :

$$\psi_i = \frac{\delta_i}{w(\mathbb{V}_i)} \quad (4.6)$$

Proposition 5. (Facteur de forme) : Ce facteur est défini comme un indicateur volumétrique du sous-ensemble \mathbb{B} . Il indique si toutes les variables d'état sont discernables de la même façon.

$$\Gamma = \frac{\prod_{i=1}^n \delta_i}{\delta_{\max}^n} = \frac{\prod_{i=1}^n \delta_i}{(w(\mathbb{B}))^n} \quad (4.7)$$

Une valeur de Γ qui tend vers l'unité indique que les différentes variables d'état qui composent le vecteur d'état sont du même ordre de magnitude, donc discernables à une même amplitude près (pour la précision ε donné). Si Γ tend vers zéro, cela indique qu'il y a au moins une variable d'état qui est moins distinguable que les autres. Pour les connaître il faut regarder le facteur d'amplitude (4.6).

Proposition 6. (Indice numérique de discernabilité) : Cet indice est défini comme une fonction du facteur d'amplitude. Il indique la proportion de discernabilité des états par rapport au plus petit espace de recherche \mathbb{V} qui puisse être défini comme domaine admissible.

$$I_{ND} = \prod_{i=1}^n (1 - \Psi_i) = \prod_{i=1}^n \left(1 - \frac{\delta_i}{w(\mathbb{V}_i)} \right) \quad (4.8)$$

Pour illustrer (4.4) et les indicateurs donnés, étudions l'exemple numérique suivant :

Exemple 4.1 : Soit un problème d'estimation d'état lié au problème (2.10). Supposons que le système (1.1) soit bidimensionnel (2D), c'est à dire composé uniquement de deux états x_1 et x_2 à estimer.

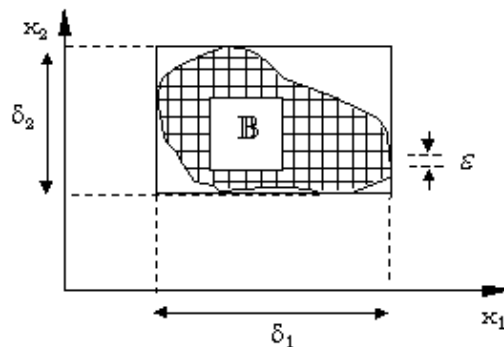


Figure 4.8 : Sous-ensemble indiscernable \mathbb{B}

Supposons aussi que nous avons les valeurs numériques suivantes pour le domaine admissible, et pour le sous-ensemble \mathbb{B} solution de (2.10).

1. $\mathbb{V} = [0, 10] \times [5, 20]$
2. $\mathbb{B} = [5, 6.5] \times [14.3, 14.4]$
3. La marge de la trajectoire de sortie obtenu par \mathbb{B} est égale à $\alpha = 5\%$.

Pour ces valeurs de \mathbb{V} et \mathbb{B} , et en considérant (4.4)-(4.8), nous pouvons obtenir les valeurs suivantes, qui nous permettent d'obtenir l'indicateur I_{ND} et d'observabilité numérique :

1. De \mathbb{B} nous pouvons en déduire que : $\delta_1 = 1.5$ et $\delta_2 = 0.1$
 Sous ces conditions l'observabilité défini par (4.5) sera de :

$$\pi = \frac{5}{1.5} = 3.33$$

Cet indicateur nous informe que plus grande est la valeur π , plus le système est observable.

2. De (4.6) le facteur d'amplitude est donné par : $\psi_1 = 1.5/10 = 0.15$ et
 $\psi_2 = 0.1/15 = 6.66 \cdot 10^{-3}$
4. De (4.7) le facteur de forme est $\Gamma = \frac{1.5 \times 0.1}{1.5^2} = 0.066$. il y a donc une variable qui est moins distinguable que les autres (Remarquons que comme $\psi_1 \gg \psi_2$, cette variable est ψ_1).
5. Enfin, l'indicateur donné en (4.8) prend une valeur égale à $I_{ND} = (1 - 0.15) \cdot (1 - 6.6 \cdot 10^{-3}) = 0.84$, ce qui indique que les états sont discernables à 84% par rapport à \mathbb{V} .

4.3.1 Propriétés de l'indicateur de l'indicateur de discernabilité

Nous démontrons les propriétés suivantes à propos de (4.8) :

- I_{ND} est compris dans $[0,1]$, c'est à dire : $0 \leq I_{ND} \leq 1$
- Si $I_{ND} = 0$ alors la commande appliquée est une entrée singulière.
- Si le système est observable, alors I_{NO} converge vers la valeur maximale possible, c'est à dire l'unité.

Preuve de a) : Deux cas sont possibles

- *Le système est observable.*

Si le système est observable, alors la solution \mathbb{B} au problème global d'optimisation (2.10) est unique, et est petite par rapport à \mathbb{V} .

$w(\mathbb{B}) \rightarrow 0$, alors $\delta_i \rightarrow 0$

En d'autres termes et pour une précision d'optimisation ε donnée, $\mathbb{B} \subset \mathbb{V}$ et le facteur d'amplitude tend vers zéro.

Si $\delta_i \rightarrow 0$, alors $\psi(\mathbb{B}) \rightarrow 0$, enfin l'indice numérique $I_{ND} \rightarrow 1$

- *Le système n'est pas observable.*

Si le système n'est pas observable, alors la solution \mathbb{B} au problème global d'optimisation (2.10) n'est pas unique, et a un ordre de magnitude similaire à celle de l'espace de recherche \mathbb{V} (voir sections 4.2.2 et 4.2.3)

$$w(\mathbb{B}) \rightarrow w(\mathbb{V}), \text{ alors } \delta_i \rightarrow w(\mathbb{V})$$

En d'autres termes, pour une précision d'optimisation ε donnée, $\mathbb{B} \subseteq \mathbb{V}$ et le facteur d'amplitude tend vers l'unité.

$$\text{Si } \delta_i \rightarrow w(\mathbb{V}), \text{ alors } \psi(\mathbb{B}) \rightarrow 1 \text{ donc l'indice numérique } I_{ND} \rightarrow 0$$

Preuve de b) : Si $I_{ND} = 0$ alors $\psi(\mathbb{B}) \rightarrow 1$ pour une précision d'optimisation appropriée. Par conséquent l'ensemble \mathbb{B} est comparable en magnitude à \mathbb{V} . Pourtant la commande appliquée n'a pas permis de distinguer les états : elle est donc singulière.

Preuve de c) : Idem que le cas a) avec un système observable.

4.4 L'observabilité pas à pas avec IMHSE

Parmi les avantages de la méthode d'estimation présentée, nous pouvons mentionner que l'observabilité du système impliqué peut être analysée pas à pas. Ceci est possible en étudiant l'ensemble solution du problème global d'optimisation (2.10). Comme nous l'indiquons dans la remarque 6, la solution à ce problème est unique en début d'horizon, si et seulement si le système impliqué est observable. La figure 4.9 montre un système qui est observable pas à pas, pendant que le début d'horizon glisse.

Si le système n'est pas observable ou traverse une zone de non observabilité locale, pendant que le début d'horizon glisse, cette non-observabilité sera détectée comme illustré dans la figure 4.10. Autrement dit la taille de l'ensemble \mathbb{B} peut nous donner des informations sur la possibilité de distinguer différents états du système impliqué. Ainsi dans le cas d'un instant de non-observabilité locale, nous pourrions détecter cette non-observabilité depuis un programme superviseur, et fixer par défaut la valeur de l'estimation actuelle à celle de l'estimation précédente actualisée.

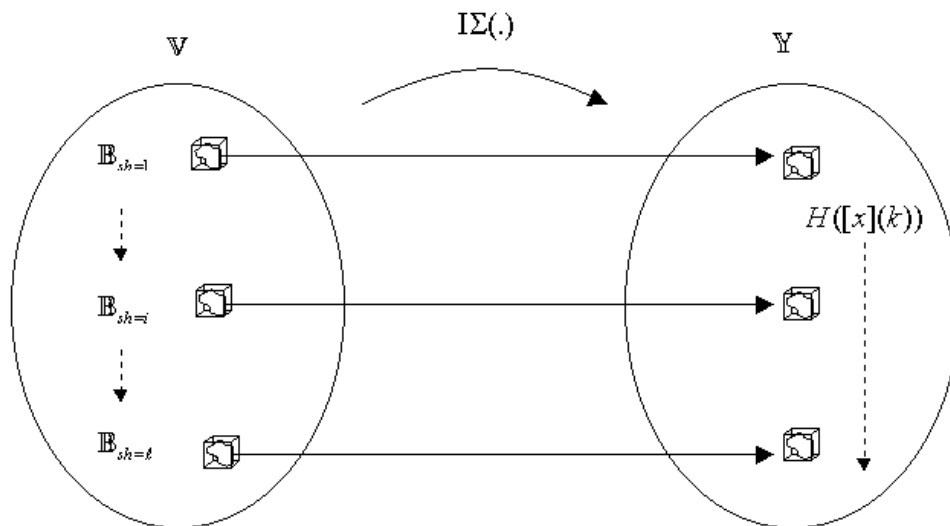


Figure 4.9 : Système observable sur tout le domaine admissible

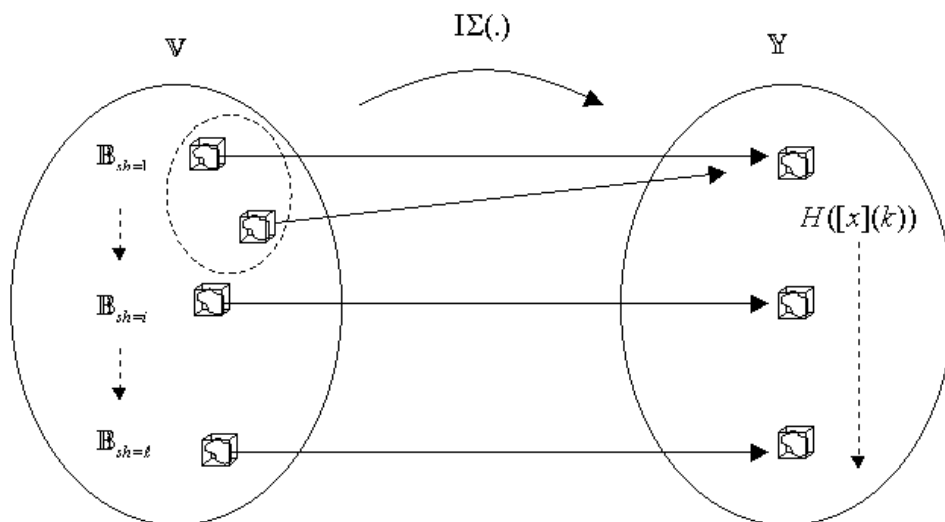


Figure 4.10 : Système non-observable localement sur le domaine admissible

4.5 Conclusions

Nous avons présenté dans ce chapitre une autre notion de l'observabilité et de la distinguabilité numérique, appelées ε -observabilité et ε -distinguabilité. Ces notions peuvent être vues comme des extensions de leurs représentations classiques, dont les états sont décrits en utilisant des techniques d'intervalles.

Cette approche considère un sous-ensemble qui contient la solution globale du problème d'estimation optimale IMHSE. Ces notions combinées avec la méthode d'estimation IMHSE sur des systèmes non-linéaires permettent de savoir s'il existe une amplitude suffisante pour faire une estimation des états du système.

La formulation proposée prend aussi en compte des indices de ε -observabilité calculés par la méthode IMHSE, qui nous donnent des informations sur l'observabilité pas à pas du système. En particulier quand les problèmes d'observabilité en ligne dépendent du point de fonctionnement et de l'excitation de l'entrée).

Des exemples graphiques en rapport avec des systèmes observables et non-observables ainsi qu'un exemple numérique illustratif ont été proposés pour mettre en évidence les idées présentées (d'autres exemples seront donnés au chapitre 6).

Chapitre 5

Détection de dysfonctionnements du modèle : Un observateur multi-modèles

Cette partie du travail sera consacrée à l'étude de la méthode IMHSE pour la détection de dysfonctionnements non prévus par le modèle interne de l'observateur.

L'estimation d'état est un processus dans lequel rien ne garantit que le modèle utilisé comme modèle interne de référence pour l'observateur, sera toujours valide ou compatible. D'un point de vue mathématique, nous pouvons faire quelques considérations pour choisir "*le meilleur modèle*" parmi un groupe de possibilités par rapport à une évolution observée (étape d'identification du procédé). Cependant la question de la validité du modèle persiste plusieurs fois après l'identification, particulièrement dans des procédés biotechnologiques. Cette problématique est illustrée pour trois modèles possibles dans la figure 5.1, dont le modèle choisi est donné dans la figure 5.2 [Gelmi, 1999], [Pérez-Correa et Agosin, 1999].

Techniquement, notre observateur IMHSE est couplé à une méthode globale d'optimisation par intervalles du type "Branch&Bound". Cette méthode permet à l'algorithme de base de faire une exploration de tout l'espace de recherche défini ou prédit. Elle trouvera alors toujours un ensemble solution qui minimisera globalement un critère prédéfini (tel que les images issues de cet ensemble restent à l'intérieure des bornes de tolérance τ données, c'est-à-dire tel que $V(.) \leq \tau$), si cet ensemble existe.

Comme il avait été présenté dans la section 2.3.4, une prédiction de l'espace de recherche peut-être utilisée pour chercher l'état en début de l'horizon pour faire la

prochaine estimation. Cependant, que faut-il faire s'il n'y a pas d'ensemble solution sur l'espace de recherche prédit ?

Remarque 19 : *Un modèle est incompatible avec une évolution observée, lorsque aucun minimum global n'existe (par rapport à tout le domaine admissible) dans la prédiction de l'espace de recherche (voir section 2.3.4). On peut détecter cet ensemble solution vide, par différents critères de l'algorithme d'optimisation global ensembliste, comme le fait que le gradient du critère ne s'annule jamais dans l'espace de recherche.*

Définition 5. *Dysfonctionnement : Le fait que l'ensemble $[\hat{\Theta}]_{sh}$ soit vide (étape 2 de l'algorithme présenté dans la section 2.3.2), c'est-à-dire qu'il n'y a pas de solution sur la prédiction $[\Omega]_{sh}$ (section 2.3.4), montre que le modèle est incompatible avec l'évolution observée du système sur l'horizon choisi. Il s'agit donc d'un dysfonctionnement du système.*

Remarque 20 : *Notons que la non-observabilité locale d'un système (Remarque 6), peut être vue aussi comme un dysfonctionnement du système, et donc traitée par commutation de modèles.*

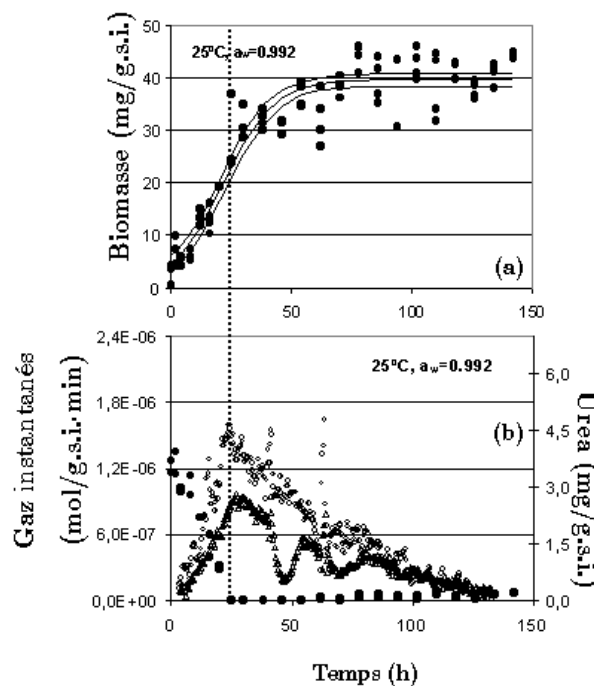


Figure 5.1 : a) Biomasse mesurée (trois possibilités) b) Gaz instantanés et Urea

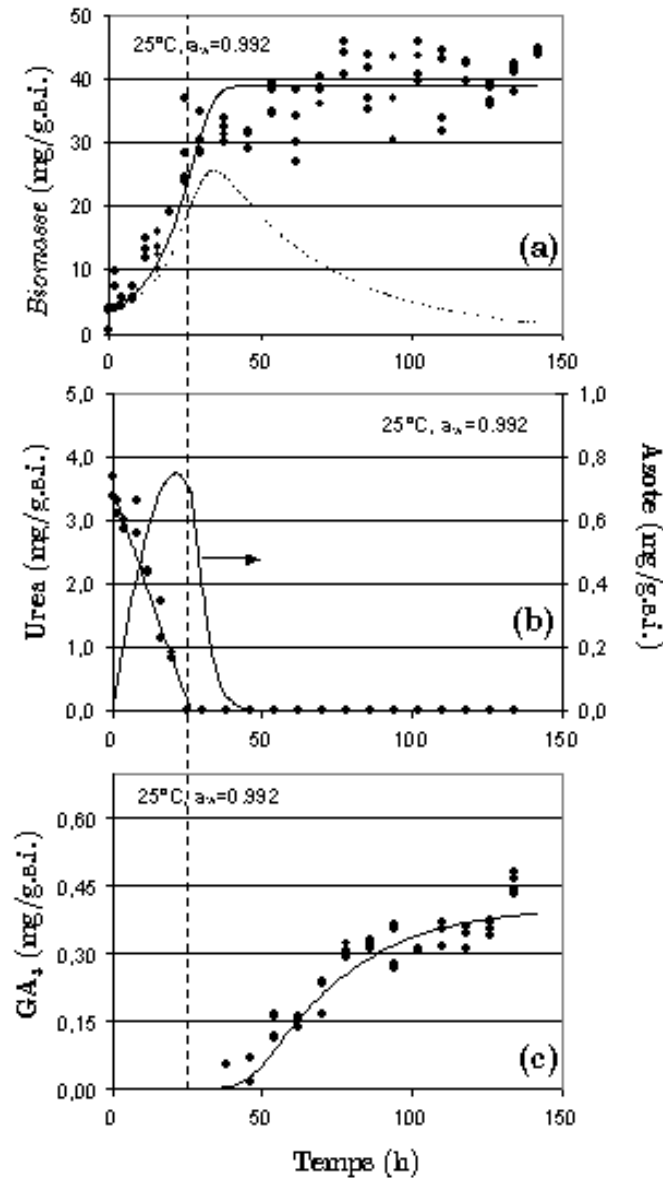


Figure 5.2 : Modèle choisi de la biomasse, Urea et GA_3

L'approche proposée dans ce chapitre correspond à l'estimateur IMHSE, qui utilise des mesures indirectes en ligne, des mesures hors ligne peu fréquentes, une prédiction pour l'espace de recherche (*section 2.3.4*), et une famille de modèles possibles pour le procédé. Cette famille de modèles sera appelée *modèles de dysfonctionnement prévus* [Valdés-González *et al*, 2002c], [Valdés-González *et al*, 2003].

L'objectif de la méthode IMHSE multi-modèles est, lors de l'estimation d'état, de sélectionner le ou les modèles capables d'expliquer le fonctionnement d'un système pendant un horizon de temps prédéfini parmi une famille de modèles.

Nous disposons donc d'une famille de modèles $\{\text{Modèles}\}$, dont le modèle complet est construit par agrégations de sous-modèles $\{\text{Modèle}-1, \text{Modèle}-2, \dots\}$. De plus, les modèles rajoutés symbolisent les différentes configurations, dans lesquelles peut se trouver l'état réel du système.

La commutation entre les modèles internes dans notre observateur (algorithme IMHSE présenté dans la section 2.3.2), est représenté par le diagramme de la figure 5.3. Nous verrons également que ce type de procédure peut être particulièrement intéressant lors qu'il s'agit de systèmes biologiques ou biotechnologiques.

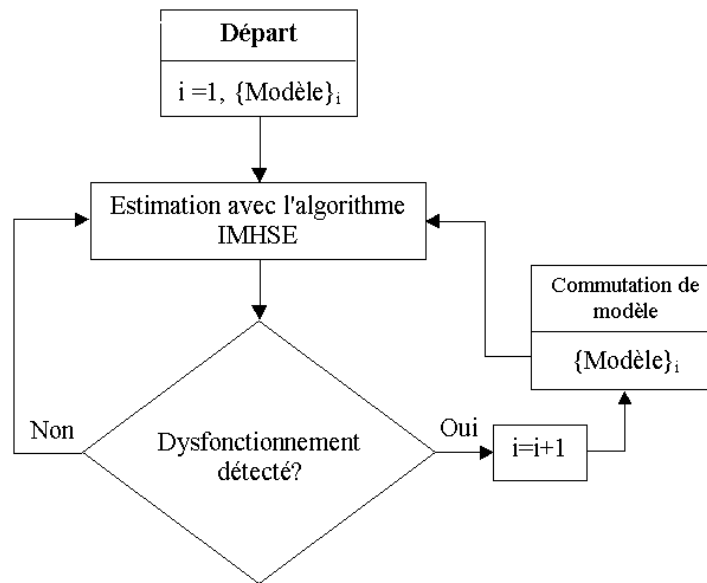


Figure 5.3 : Méthode IMHSE avec commutation de modèle interne

Le schéma ci-dessus montre de façon intuitive l'algorithme IMHSE quand le modèle interne du procédé utilisé est commuté, dû au fait d'un dysfonctionnement. Ce schéma est détaillé dans les algorithmes suivants.

5.1 IMHSE Multi-modèles

Pour implémenter cet observateur, nous allons redéfinir le système (1.1) en introduisant l'ensemble $\{\eta_i\}_1^\kappa$ qui représente les " κ " modèles disponibles pour un système dont le modèle de référence, tel que $\eta_i = \{f_i, h_i\}$. Cet ensemble nous conduit donc à transformer le système d'équations original (1.1) en :

$$\Sigma_i : \begin{cases} x(k+1) = f_i(x(k), u(k)) \\ y(k) = h_i(x(k)) \end{cases} \quad (i : 1 \rightarrow \kappa) \quad (5.1)$$

L'observateur utilisera cette famille pour chercher le modèle capable d'expliquer le fonctionnement du système sur l'horizon choisi. Pour cela et en utilisant les concepts de prédiction de l'espace de recherche données en 2.3.4, nous aurons plusieurs domaines prédits possibles $[\Omega]_{sh}$, comme il est montré dans la figure suivante :

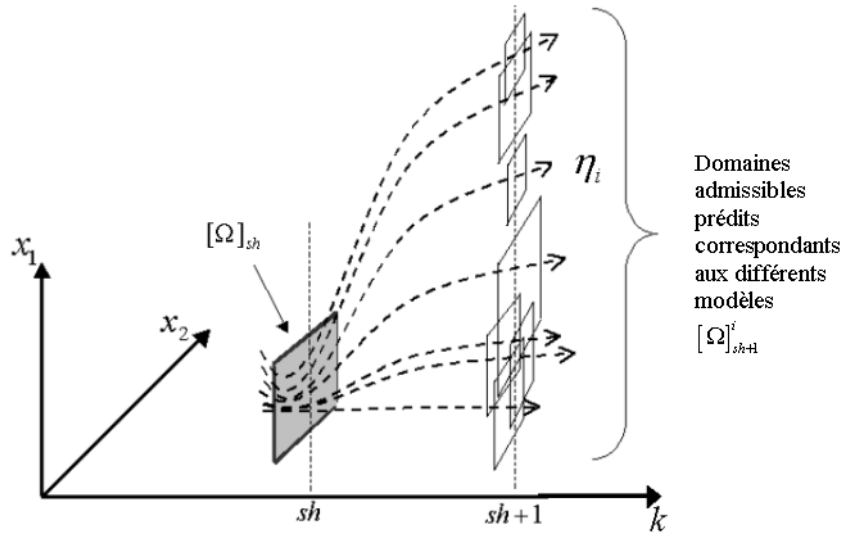


Figure 5.4 : Domaines admissibles prédits pour tout modèle de la famille

5.1.1 L'algorithme multi-modèles de base

L'algorithme de base suivante met en œuvre le principe de cette approche multi-modèles, avec $\ell = sh + lh - 1$ et T_k la durée de l'expérience:

Algorithme 4 : *IMHSE MULTI-MODELES(.)*

Entrée : $T_k, [\Omega], lh, B_\varepsilon, \Sigma(\cdot), \{[u]\}_{sh}^\ell, \{[y]\}_{sh}^\ell, \{\eta_i\}_1^\kappa$

Sortie : $[\hat{x}]_\ell$

Initialisation : $sh := 1, i := 1, [\Omega]_{sh} = [\Omega], flag := 0$

WHILE $sh \leq T_k$

 WHILE $flag := 0$

 1. Résoudre le problème global d'optimisation (2.10) par intervalles

$[\hat{\Theta}]_{sh} := \text{G-Optimisation}([\Omega]_{sh}, B_\varepsilon, V, \eta_i)$

```

2. IF  $[\hat{\Theta}]_{sh} \neq \{\emptyset\}$ 
    2.1 Régularisation ou filtrage
         $[\hat{x}]_{sh} := \beta_{sh}([\hat{\Theta}]_{sh}, [\hat{x}]_{sh/sh-1})$ 
    2.2 Calcul de la séquence d'état jusqu'à la fin de l'horizon
         $\{[\hat{x}]\}_{sh}^\ell := \text{I-Trajectoire}([\hat{x}]_{sh}, \Sigma(\cdot), sh, lh, \{[u]\}_{sh}^\ell, \eta_i)$ 
    2.3 Définition du domaine admissible de la prochaine estimation
         $[\Xi]_{sh} := \text{I-Trajectoire}([\hat{x}]_{sh}, \Sigma(\cdot), sh, sh+1, \{[u]\}_{sh}^{sh+1}, \eta_i)$ 
         $[\Omega]_{sh} := \text{Expansion}([\Xi]_{sh}, \delta)$ 
         $flag := 1$ 
    ELSE
         $i = i + 1$ 
         $flag := 0$ 
    END
3. Faire glisser l'horizon d'un pas pour la prochaine estimation:  $[\hat{x}]_{\ell+1}$ 
     $sh := sh + 1$ 
END
END

```

5.1.2 Remarques à propos de l'implémentation de base

- a) Pour accélérer l'approche nous commençons la recherche à chaque pas, avec le dernier modèle utilisé (modèle précédent).
- b) Il est possible d'interrompre l'algorithme et de ne pas balayer tous les modèles de la famille. Dès qu'une solution est trouvée, on pourra affirmer avoir trouvé un modèle explicatif sans garantir que d'autres comportements sont aussi possibles.
- c) S'il est possible de munir l'ensemble des modèles d'une distance, on pourra balayer la famille des modèles en partant du plus proche au plus éloigné du modèle initial (c'est-à-dire précédent) et s'arrêter. On aura trouvé le meilleur modèle au sens du plus proche pour la distance considérée.

Remarque 21 : *L'idée (5.1.2.c) précédente est possible en supposant que le système évolue lentement dans le temps. Ceci n'est pas une contrainte majeure parce que cette approche est conçue spécialement pour des applications comme*

des bioprocédés, dans lesquels cette hypothèse est vraie la plupart du temps [Pérez-Correa et Agosin, 1999], [Peña y Lillo et al, 2001].

5.1.3 Garantie sur tous les comportements prévus

La remarque (5.1.2.b), nous montre qu'il est aussi possible de trouver parmi la famille de modèles disponibles, d'autres modèles qui expliquent aussi le comportement du système (modèles faisables). Comme il y a peut-être plusieurs domaines admissibles prédits faisables (voir figure 5.4), le domaine admissible prédit $[\Omega]_{sh}$ ne sera pas forcément optimal, parmi les modèles de dysfonctionnement prévus. Pour remédier à ce problème, nous proposons d'améliorer la version précédente de l'algorithme en ajoutant de nouvelles étapes, à savoir :

- a) Recherche de l'ensemble $[\Omega]_{sh+1}^i$ issu de tous les modèles $\{\eta_i\}_1^\kappa$ compatibles avec l'évolution observée (minimum global contenu dans l'espace initial de recherche).

FOR $i = 1 \rightarrow \kappa$

$$[\Xi]_{sh+1}^i := \text{I-Trajectoire}([\hat{x}]_{sh}, \Sigma(\cdot), sh, sh+1, \{[u]\}_{sh}^{sh+1}, \eta_i)$$

$$[\Omega]_{sh+1}^i := \text{Expansion}([\Xi]_{sh+1}, \delta)$$

$$[\hat{\Theta}]_{sh+1}^i := \text{G-Optimisation}(\cdot)$$

$$\text{IF } [\hat{\Theta}]_{sh+1}^i \neq \{\phi\}$$

$[\Omega]_{sh+1}^i \rightarrow$ Garder cette boîte dans une liste Λ de domaines admissibles compatibles

ENDIF

END

- b) Union des domaines admissibles compatibles $[\Omega]_{sh+1}^i$, correspondant aux modèles trouvés dans l'étape précédente (5.1.3.a).

$$[\Omega]_{sh+1} = \bigcup_{\Lambda} [\Omega]_{sh+1}^i \quad (5.2)$$

Ces étapes a) et b) sont illustrées dans la figure suivante, qui complète les concepts représentés par les figures 2.4 et 5.4.

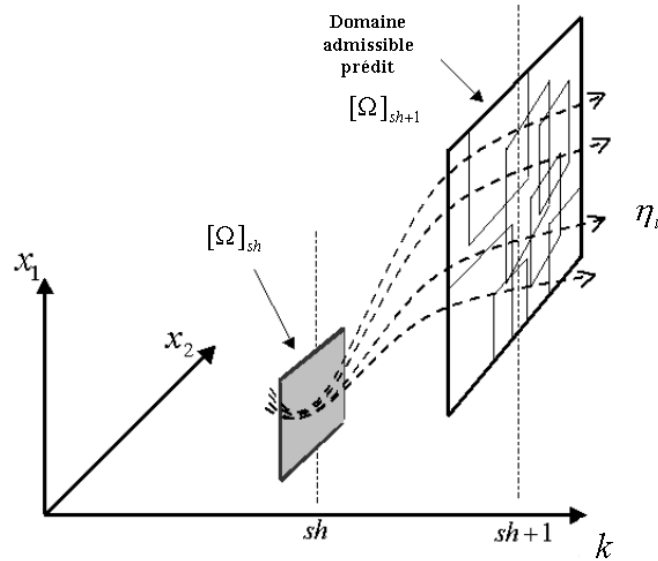


Figure 5.5 : Union de tous les domaines admissibles compatibles en une seule boîte

5.1.4 Sélection du meilleur modèle

Considérant les remarques 22 et 5.1.2.c, le modèle pour lequel on optera quand il y a plus d'un modèle qui peut expliquer l'évolution observée, sera celui qui est le plus proche du modèle précédemment utilisé, c'est-à-dire celui qui minimise l'expression de la distance $\|\eta_{sh} - \eta_{sh+1}\|_2$.

Remarque 22 : Notons qu'une fois choisi un modèle pour l'observateur, ce dernier reste valable pendant tout l'horizon du temps pour l'estimation en cours.

Exemple 5.1: Nous allons prendre un modèle du type (5.1) avec 10 modèles de dysfonctionnements prévus $\{\eta_i\}_1^{10}$. La différence entre ces modèles sera la valeur de deux de ses paramètres (δ_1, δ_2) , c'est à dire, $\eta_i = (\delta_{1,i}, \delta_{2,i})$ tel que $\delta_1 = \{0.1, 0.2\}$ et $\delta_2 = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. L'étape qui consiste à savoir si la prédiction du domaine admissible est compatible avec l'évolution observée, est effectuée en regardant si le gradient du critère peut s'annuler dans le domaine admissible prédit. Ceci est illustré dans le tableau I suivant :

Tableau 1 : Modèles compatibles avec l'évolution observée

M.D.P	η_1	η_2	η_3	η_4	η_5	η_6	η_7	η_8	η_9	η_{10}
$0 \in \nabla[V](\Omega_{sh+1}^i)?$	1	0	0	0	0	1	0	0	0	1

M.D.P. : Modèles de dysfonctionnement prévus.

Observation : La valeur binaire "1" est affichée dans le tableau I, si le gradient du critère s'annule dans l'espace de recherche (autrement dit si le modèle η_i est compatible avec l'évolution observée), "0" dans le cas contraire.

En considérant 5.1.4, nous cherchons le modèle le plus proche de celui utilisé précédemment (qui est supposé être η_2 dans notre exemple). Pour cela nous proposons de choisir le modèle dont les paramètres s'écartent le moins du modèle utilisé dans l'estimation antérieure.

Tableau 2 : Sélection de modèle

	$\delta_2 = 0.1$	$\delta_2 = 0.2$	$\delta_2 = 0.3$	$\delta_2 = 0.4$	$\delta_2 = 0.5$
$\delta_1 = 0.1$	η_1	η_2	η_3	η_4	η_5
$\delta_1 = 0.2$	η_6	η_7	η_8	η_9	η_{10}

Dans notre exemple, le critère pour choisir le meilleur modèle en considérant que les modèles compatibles avec l'évolution observée sont $i = \{1, 6, 10\}$, et que le modèle précédent est $j = 2$, sera le suivant :

$$D_{j,i} = \sqrt{(\delta_{1,i} - \delta_{1,j})^2 + (\delta_{2,i} - \delta_{2,j})^2} \tag{5.3}$$

Les résultats sont $D_{2,1} = 0.10$, $D_{2,6} = 0.14$, $D_{2,10} = 0.31$, donc le modèle qui sera choisi pour notre approche est η_1 , qui est le plus proche.

De manière générale, l'équation (5.3) peut être réécrite en considérant que i représente les modèles compatibles et que j représente le modèle précédemment utilisé. Nous avons alors :

$$D_{j,i} = \sqrt{\sum_i (\eta_{sh}^j - \eta_{sh}^i)^2} \quad (5.4)$$

5.1.5 L'algorithme IMHSE Multi-Modèles

L'algorithme suivant représente la version de l'approche IMHSE Multi-modèles (Algorithme 4), dans lequel nous avons intégré les étapes précédentes (recherche du domaine admissible et recherche du meilleur modèle parmi les modèles de dysfonctionnement prévus).

Algorithme 5 : IMHSE MULTI-MODELES(.)

Entrée : T_k , $[\Omega]$, lh , B_ε , $\Sigma(\cdot)$, $\{[u]\}_{sh}^\ell$, $\{[y]\}_{sh}^\ell$, $\{\eta_i\}_1^K$

Sortie : $[\hat{x}]_\ell$

Initialisation : $sh := 1$, $i := 1$, $[\Omega]_{sh} = [\Omega]$

WHILE $sh \leq T_k$

1. Résoudre le problème global d'optimisation (2.10) par intervalles

$$[\hat{\Theta}]_{sh} := \text{G-Optimisation}([\Omega]_{sh}, B_\varepsilon, V, \eta_i)$$

2. Régularisation ou filtre

$$[\hat{x}]_{sh} := \beta_{sh}([\hat{\Theta}]_{sh}, [\hat{x}]_{sh/sh-1})$$

3. Calcul de la séquence d'état jusqu'à la fin de l'horizon

$$\{[\hat{x}]\}_{sh}^\ell := \text{I-Trajectoire}([\hat{x}]_{sh}, \Sigma(\cdot), sh, lh, \{[u]\}_{sh}^\ell, \eta_i)$$

4. Prédiction du domaine admissible $[\Omega]_{sh+1}$ et du meilleur modèle

- 4.1 Recherche de l'ensemble de tous les modèles $\{\eta_i\}_1^K$ compatibles avec l'évolution observée (Section 5.1.3.a)

- 4.2 Union des domaines admissibles $[\Omega]_{sh+1}^i$ correspondant aux modèles trouvés ci-dessus (5.2).

- 4.3 Sélection du meilleur modèle au sens du plus proche (5.4).

5. Faire glisser l'horizon d'un pas pour la prochaine estimation : $[\hat{x}]_{\ell+1}$

$$sh := sh + 1$$

END

5.2 Conclusions

Une procédure pour détecter des dysfonctionnements du système basée sur un observateur non-linéaire ensembliste globalement convergent a été présentée. Cette approche utilise les avantages de la méthode IMHSE pour faire des estimations d'états sur des systèmes non linéaires. Elle utilise aussi le concept de la prédiction d'un espace de recherche pour les états du système afin de détecter des dysfonctionnements non prévus par le modèle interne de l'observateur.

Quand le dysfonctionnement a été détecté, la méthode utilisera une famille de modèles de dysfonctionnements prévus. Le choix d'un autre modèle de cette famille, permettra au fur et à mesure de se rapprocher du modèle le plus représentatif du comportement réel observé du système. Ceci est fait en commutant le modèle interne, parmi les modèles de dysfonctionnement prévus disponibles. L'algorithme développé permet de choisir le meilleur modèle, c'est-à-dire le plus proche de celui qui a été utilisé dans l'estimation antérieure.

Avec cette méthode, on exploite tous les types de mesures qui peuvent être disponibles, en ligne ou hors ligne. De plus, les problèmes d'observabilité locale du système peuvent être considérés comme un dysfonctionnement du modèle interne de l'observateur et donc traités par commutation de modèles.

Il faut signaler que pour ce type d'approche, plus la taille du vecteur d'état est grande, plus elle devrait être appliquée à des systèmes qui évoluent lentement dans le temps pour des raisons de temps de calcul. La taille du vecteur d'état reste une limite pour cette approche. Cependant, au niveau des bioprocédés la taille du vecteur d'état est normalement limitée.

Des exemples d'applications de cette méthode multi-modèles seront donnés au chapitre 6.

Chapitre 6

Applications et discussions

Dans ce chapitre, nous allons montrer plusieurs exemples d'application de la méthode d'estimation proposée dans cette thèse. Nous utiliserons pour cela des modèles de systèmes non-linéaires théoriques et d'autres modèles issus de systèmes réels. Nous appliquons la méthode IMHSE pour l'estimation des variables d'état sur différents bioprocédés, et nous montrerons que la méthode IMHSE et les algorithmes développés répondent à nos attentes en simulation. Nous traiterons plusieurs types de systèmes, en prenant soin de comparer les résultats obtenus avec ceux fournis par d'autres méthodes classiques d'estimation ou par des mesures expérimentales disponibles.

Le premier exemple est un modèle de fermentation générique. Il va nous permettre d'illustrer et de valider les capacités de la méthode d'estimation par intervalles proposée. Nous validerons aussi la technique de détection de systèmes qui ne sont pas observables, l'estimation avec des perturbations sur l'état et l'utilisation de filtrage y compris dans le cas d'un bruit additif en sortie.

La deuxième application correspond au même modèle que le système générique précédent, mais dont un des paramètres (le taux spécifique de croissance) est décrit de façon hybride. Cette application a pour but de mettre en valeur la méthode d'estimation couplée à un système hybride (section 2.5).

La troisième application correspond à la détection de systèmes qui ne sont pas observables ou qui sont faiblement observables, comme ceux indiqués au chapitre 4.

Dans le cadre de la quatrième application, le test correspond à un modèle d'état issus d'un bioprocédé réel de fermentation sur substrat solide ou SSC (Solid Substrate Culture). L'objectif est la croissance d'un champignon appelé *Gibberella Fujikuroi* et la production d'acide gibbérellique (GA_3) sur substrat solide en colonnes de Raimbault. Il s'agit donc de la production d'une hormone végétale, biologiquement active. Parmi les principales utilisations agricoles du GA_3 figurent le traitement des fruits (oranges, raisin...), ainsi que des légumes (pommes, artichaut...) et des céréales. Ce procédé est important car plus de 90% de l'acide gibbérellique utilisé dans le monde, et principalement en Californie, Chili et Israël, est destiné à l'obtention de raisin de table sans pépin.

Le cinquième test est une application de la méthode d'estimation avec de multiples modèles, mais cette fois-ci sur un modèle non-linéaire de Van-Der-Pol à paramètres perturbés.

Enfin, le sixième test est une application de la méthode IMHSE multi-modèle pour l'obtention des mesures indirectes de la proportion d'eau dans un bioréacteur. Il s'agit d'un bioréacteur à échelle pilote de 200-Kg avec agitation périodique. La supervision de ce bioréacteur, et du pourcentage d'eau de la fermentation est très importante pour la croissance du champignon *Gibberella Fujikuroi* cultivé.

Pour les deux derniers bioprocédés, nous utilisons des données expérimentales. Ceci a été possible grâce aux données et expérimentations réalisées au laboratoire de génie chimique de l'Université Catholique du Chili [Gelmi, 1999], [Peña y Lillo, 1999], [Pérez-Correa et Agosin, 1999] et [Peña y Lillo *et al*, 2001].

6.1 Cas 1 : Application à un procédé batch générique

Dans une première étape nous cherchons à valider la méthode IMHSE permettant d'estimer les variables d'état non mesurables à partir de données simulées. Le modèle du procédé permettant d'exprimer la relation entre les mesures indirectes et les variables à estimer est présenté en [Cazzador et Lubenova, 1995] et [Boillereaux, 1996]. Il s'agit d'un modèle simple et générique de fermentation, qui va nous permettre de valider notre observateur sur un exemple unique en appliquant pas à pas la méthode proposée. Afin d'évaluer les performances de l'observateur, nous ferons en même temps une comparaison avec le filtre de Kalman étendu (FKE, voir annexe D) et une version locale de la méthode à horizon fuyant appelé MHSE-BFGS (nommé ainsi pour la technique quasi-newtonienne d'optimisation utilisée, voir annexe C).

Ce bioprocédé fonctionne en mode batch. Le modèle dynamique discret non-linéaire qui décrit ce genre de bioprocédés est le suivant, où Δt est la période d'échantillonnage :

$$\begin{aligned} X_{k+1} &= X_k \cdot (1 + \Delta t \cdot \mu_k) \\ S_{k+1} &= S_k - \Delta t \cdot \frac{\mu_k}{\gamma} \cdot X_k \end{aligned} \quad (6.1)$$

Dans le modèle (6.1), X_k représente la concentration en biomasse en (g/l) dans le réacteur, S_k représente la concentration en substrat, aussi en (g/l). Le terme μ_k (ou $\mu_k(S_k)$) est le taux spécifique de croissance de la biomasse en (h^{-1}) et sera caractérisé ici par le modèle de Monod :

$$\mu_k = \mu_{\max} \cdot \frac{S_k}{k_s + S_k} \quad (6.2)$$

Le modèle de la sortie correspond en général aux capteurs disponibles, et donc il varie d'un procédé à un autre. Ici ce modèle est lié à la respiration de micro-organismes, dont le modèle suivant permet de mesurer la vitesse de respiration (a est généralement plus grand que b).

$$y_k = (a \cdot \mu_k + b) \cdot X_k \quad (6.3)$$

6.1.1 Conditions générales du test

a) Les paramètres du modèle sont les suivants :

Tableau 3 : Paramètres du modèle

	Nom / Description	Valeur
μ_{\max}	Taux spécifique de croissance maximale	0.4 (h^{-1})
k_s	Constant	5 (g/l)
γ	Constant	0.8
a	Constant	0.1
b	Constant	0.01

b) Conditions initiales du système (6.1)-(6.3) en simulation :

- $X_k(0) = 0.15$ (g/l) et $S_k(0) = 4.9$ (g/l)
- Bruit additif en sortie : Bruit gaussien de 3%

c) Conditions initiales pour la méthode IMHSE :

- Etats initiaux : inconnus
- Le domaine admissible pour la biomasse X_k et le substrat S_k sont :

$$[X_k] = [0 \quad 4.5] \text{ et } [S_k] = [0 \quad 5.5]$$

Alors le domaine admissible du problème d'estimation est donné par :

$$[\Omega] = [X_k] \times [S_k].$$

- Les mesures sur la sortie simulée (5.3) sont disponibles régulièrement toutes les 6 minutes ($\Delta t = 6$ minutes).
- L'expérience a une durée de 30 heures.
- La longueur de l'horizon est fixée à 2 heures : $lh = 2$ heures.
- Paramètre de pondération du filtre : $\alpha = 0.3$ (voir équation (2.13))
- Précision d'optimisation $B_\varepsilon = 0.001$
- La borne du critère d'optimisation est fixée à 5%

6.1.2 Résultats

- a) *Pas de perturbation* : La sortie utilisée non bruitée, représentée mathématiquement par (6.3), est illustrée dans la figure 6.1, tandis que les résultats des estimations sur la biomasse, le substrat, le taux spécifique de croissance et l'observabilité numérique sont présentés dans les figures 6.2 à 6.5 respectivement [Valdés-González et Flaus, 2001a].

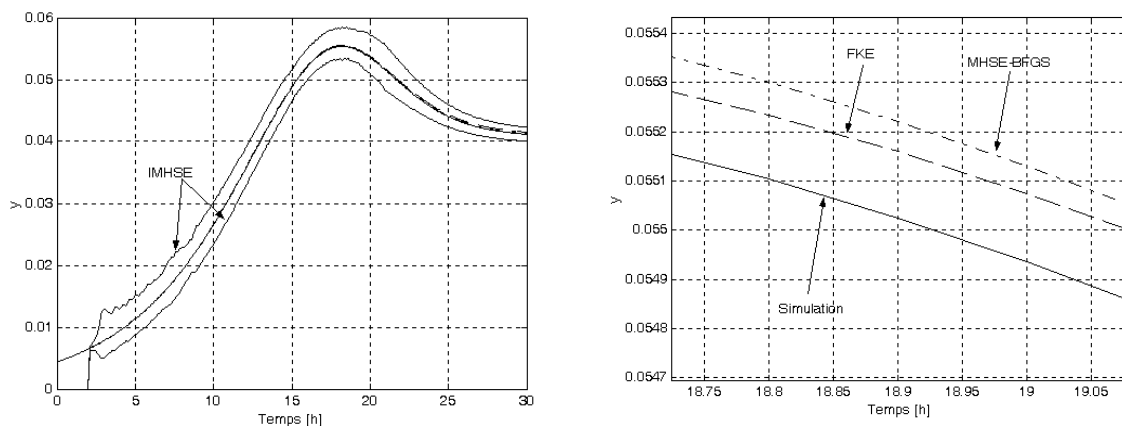


Figure 6.1 : Sortie non bruitée

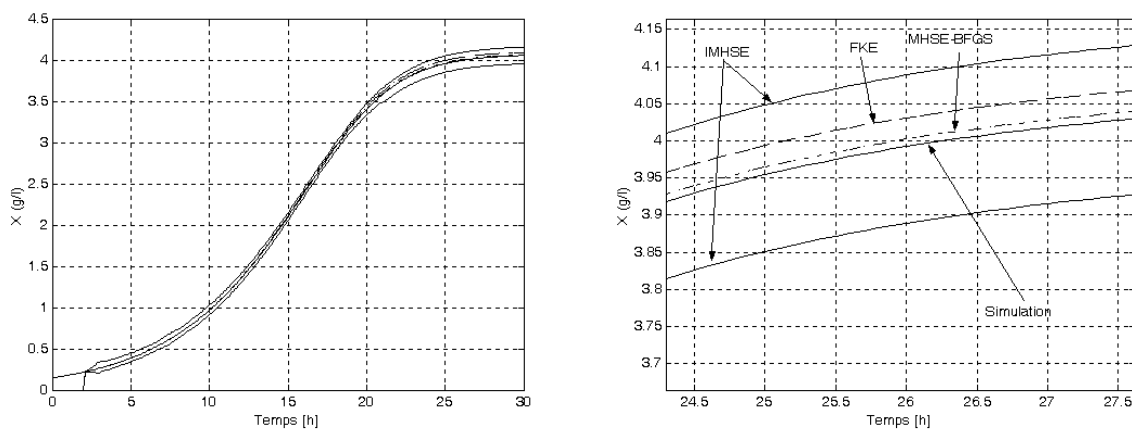


Figure 6.2 : Biomasse simulée (ligne continue) et estimations

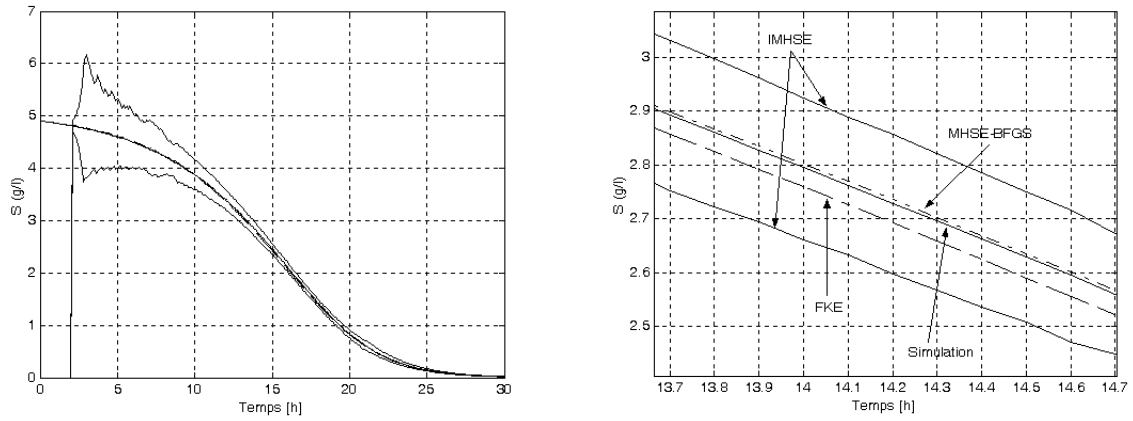


Figure 6.3 : Substrat simulé (ligne continue) et estimations

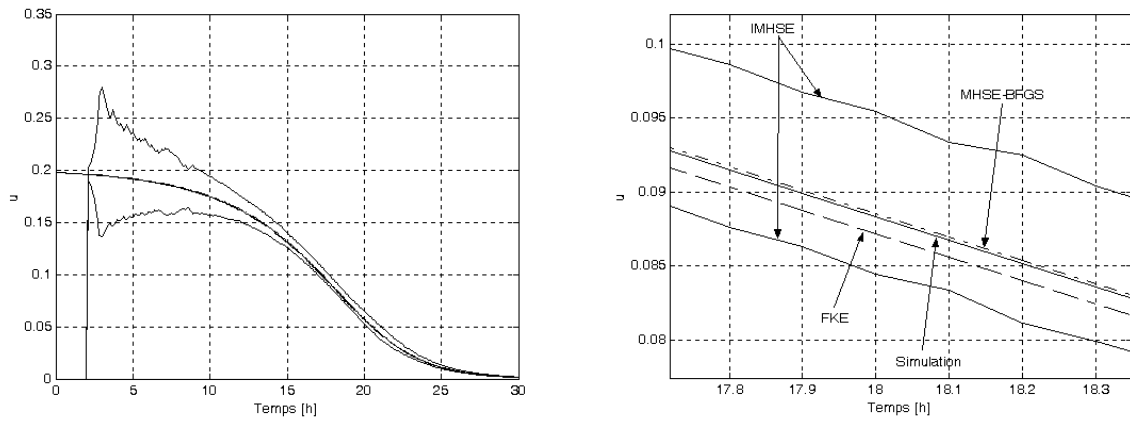


Figure 6.4 : Taux spécifique de croissance simulé (ligne continue) et estimations

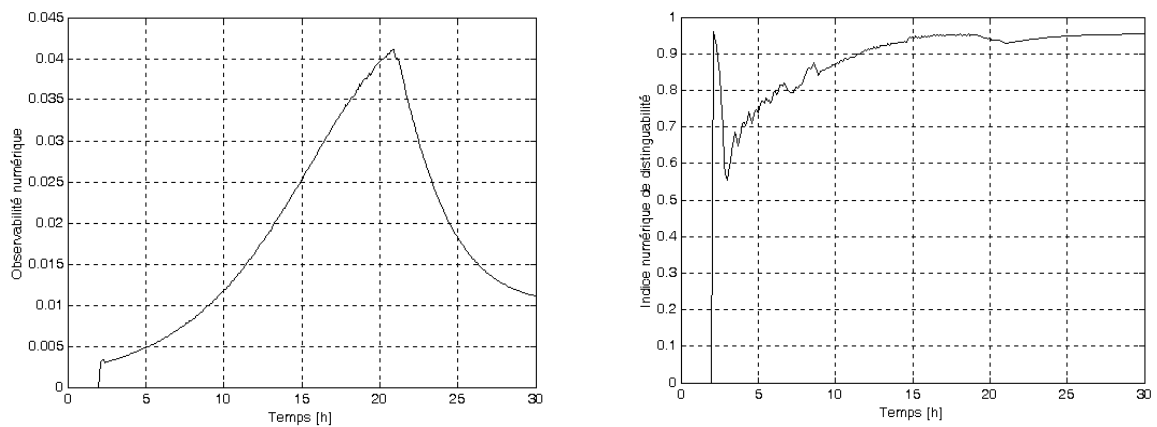


Figure 6.5 : Indicateurs d'observabilité numérique et de discernabilité

6.1.2.1 Analyse de résultats et conclusion cas a)

Les estimations montrent une excellente cohérence entre les valeurs simulées et les résultats des estimations par intervalles de la biomasse, le substrat et le taux spécifique de croissance. Le temps de calcul est approximativement de 30 secondes pour chaque estimation, ce qui confirme les bonnes performances de la méthode IMHSE proposée. Remarquons que les intervalles contiennent la solution au problème d'estimation optimal à la fin de l'horizon. De plus, les techniques classiques d'estimation telles que le filtre de Kalman étendu et la méthode locale IMHSE-BFGS donnent aussi de bons résultats sous les conditions imposées.

D'autre part, les indicateurs d'observabilité numérique et de discernabilité, π et I_{ND} respectivement, présentés dans la section 4.3, équations (4.5) et (4.8) respectivement, nous informent d'une part de l'observabilité pas à pas et de la distinguabilité des états du système. Grâce à l'indicateur π , nous constatons que la zone d'observabilité optimale se situe aux alentours du point d'inflexion des courbes qui désignent les trajectoires de la biomasse et du substrat, c'est à dire vers $k=20$ heures. Nous constatons aussi que les zones où l'effort nécessaire pour distinguer les états est le plus grand, sont au commencement et à la fin de l'expérience. Grâce à l'indicateur I_{ND} , nous constatons que les états sont discernables sur le domaine admissible approximativement à 90% après $k=12$ heures, pour les conditions imposées et la sortie bornée obtenue.

b) *Avec bruit additif en sortie de 3%* : La sortie bruitée est donnée dans la figure 6.6. Les résultats des estimations sur la biomasse, le substrat, le taux spécifique de croissance et l'observabilité numérique associée sont présentés dans les figures 6.7 à 6.10 respectivement. Nous utilisons un filtrage linéaire identique à celui présenté en (2.13).

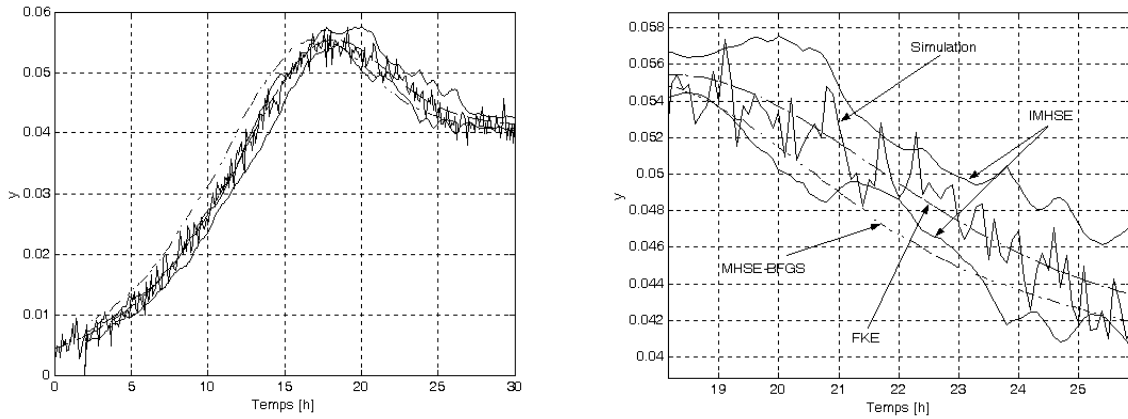


Figure 6.6 : Sortie bruitée

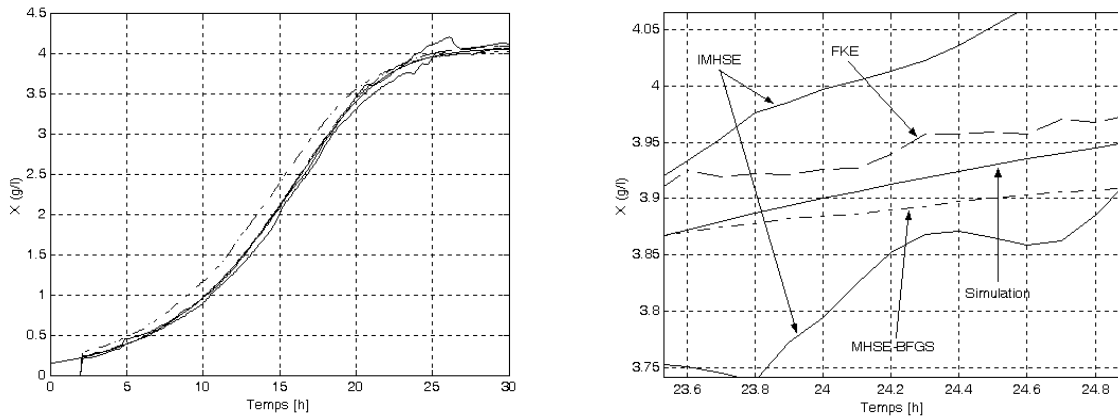


Figure 6.7 : Biomasse simulée (ligne continue) et estimations

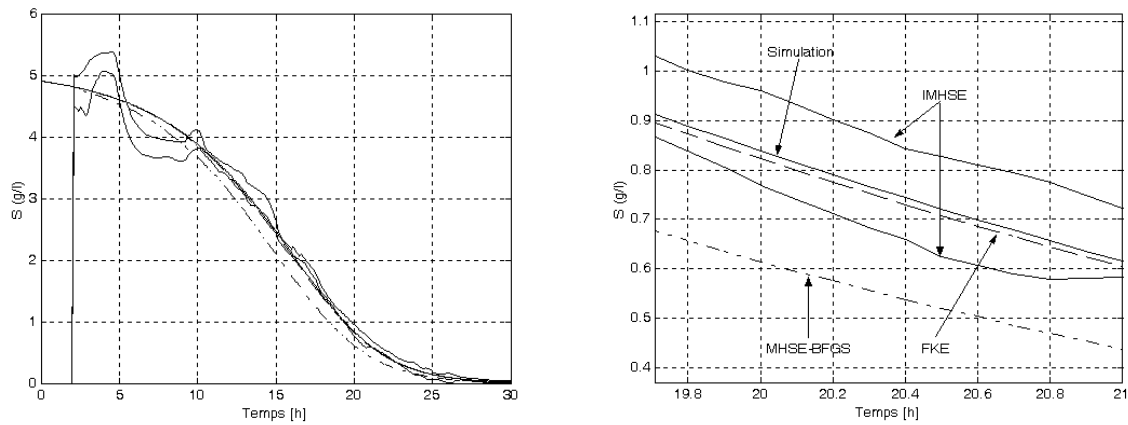


Figure 6.8 : Substrat simulé (ligne continue) et estimations

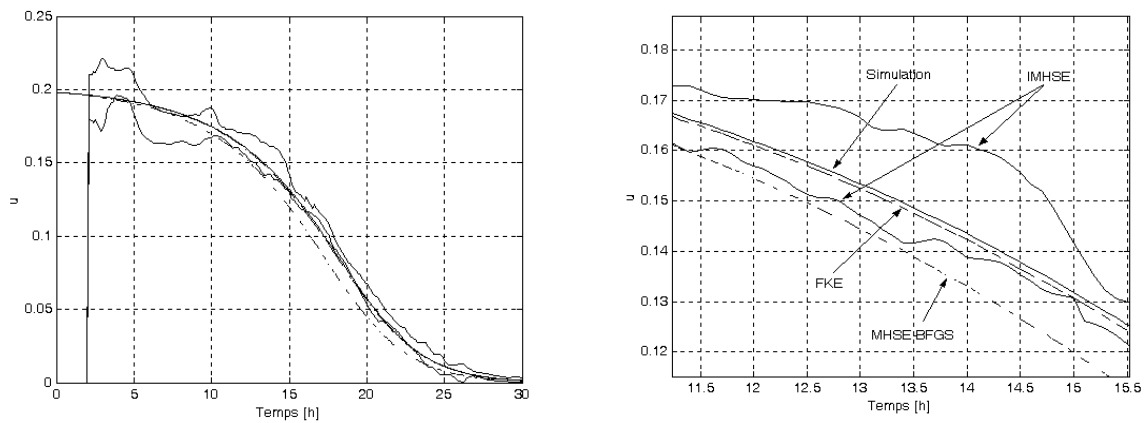


Figure 6.9 : Taux spécifique de croissance simulé (ligne continue) et estimations

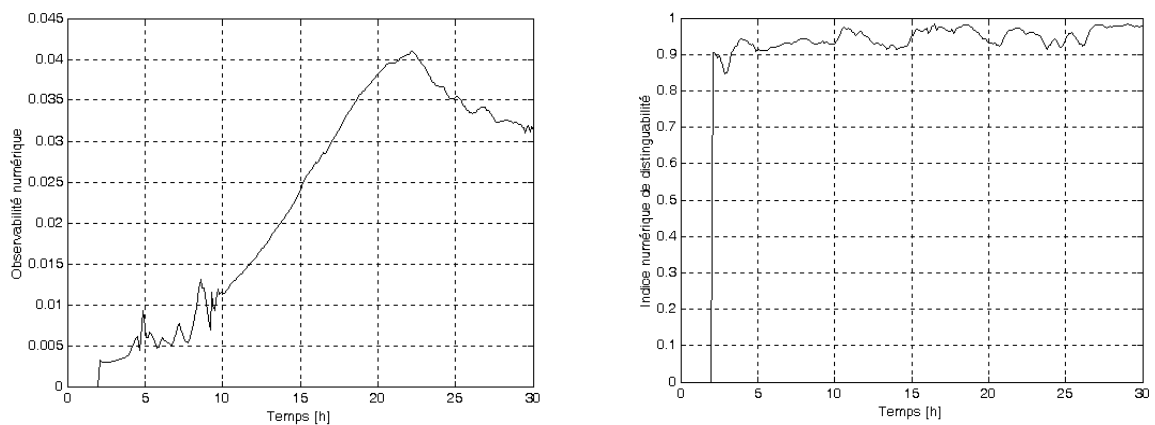


Figure 6.10 : Indicateurs d'observabilité numérique et de discernabilité

6.1.2.2 Analyse de résultats et conclusion cas b)

La bruit additif de 3% ajouté à la sortie qui entraîne des variations sur la biomasse et le substrat estimé est bien compensé et filtré par la méthode. Le filtrage permet un lissage des courbes et les variations sont donc moins brutales. Les résultats d'estimation globaux trouvés nous donnent de bonnes estimations des variables d'état. Le temps de calcul reste de l'ordre de 30 secondes pour faire chaque estimation. Notons encore que les techniques classiques d'estimation tel que le filtre de Kalman et la méthode locale IMHSE-BFGS donnent des bons résultats sous les conditions imposées. Remarquons aussi que la variable d'état qui est la moins observable est le substrat, surtout pendant les premières heures de l'expérience.

L'indicateur d'observabilité, nous montre encore le besoin d'un effort informatique plus grand aux extrêmes des trajectoires, pour discerner les états du domaine admissible. Nous constatons aussi que la zone d'observabilité optimale se maintient aux alentours de $k=20$ heures.

En même temps l'indicateur de discernabilité nous informe que pour les conditions imposées et la sortie bornée obtenue, les états sont discernables et que le système est plus facilement observable après $k=5$ heures. Les états sont discernables approximativement à plus de 90% sur le domaine admissible.

- c) *Avec perturbations sur l'état* : Considérons que suite à une intervention externe sur le réacteur à $k=10$ heures, le substrat disponible est perturbé à 50% de sa valeur (50% du substrat est perdu de façon instantanée). L'influence de cette perturbation sur l'état du système et la performance de notre observateur sont montrées dans les figures suivantes :

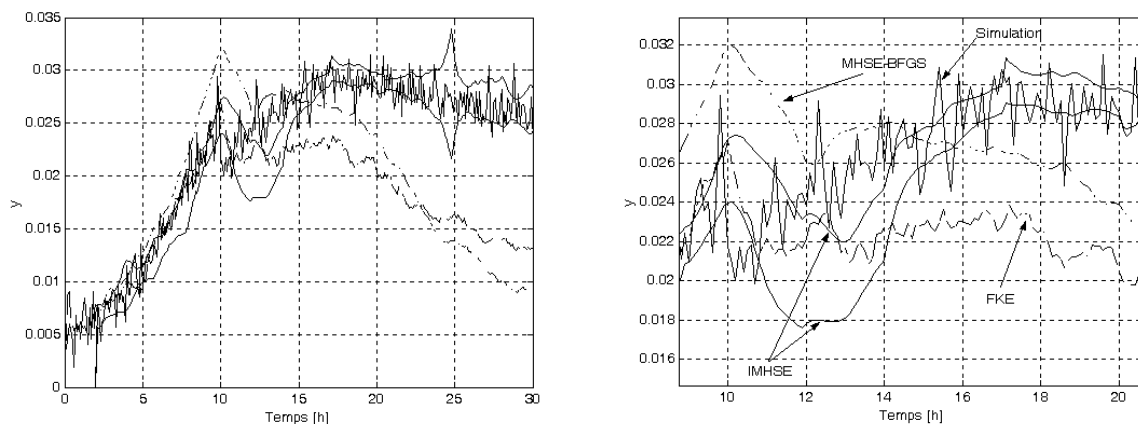


Figure 6.11 : Sortie bruitée considérée

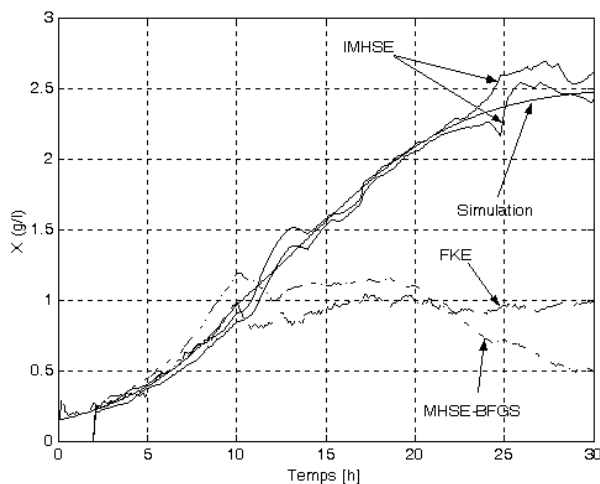


Figure 6.12 : Biomasse simulée (ligne continue) et estimations

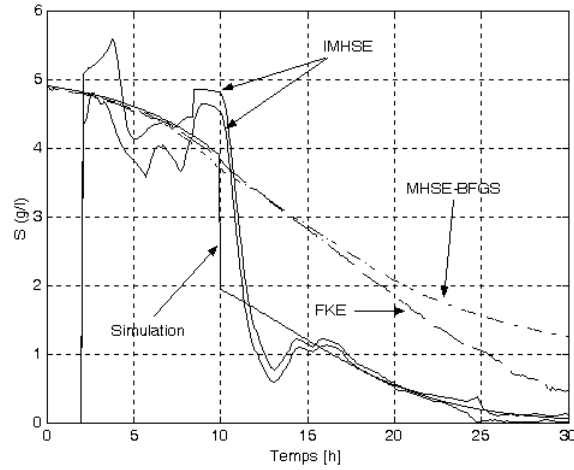


Figure 6.13 : Substrat simulé (ligne continue) et estimations

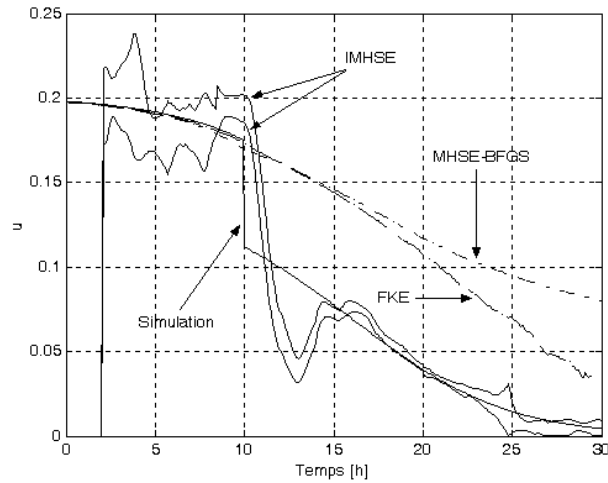


Figure 6.14 : Taux spécifique de croissance simulé (ligne continue) et estimations

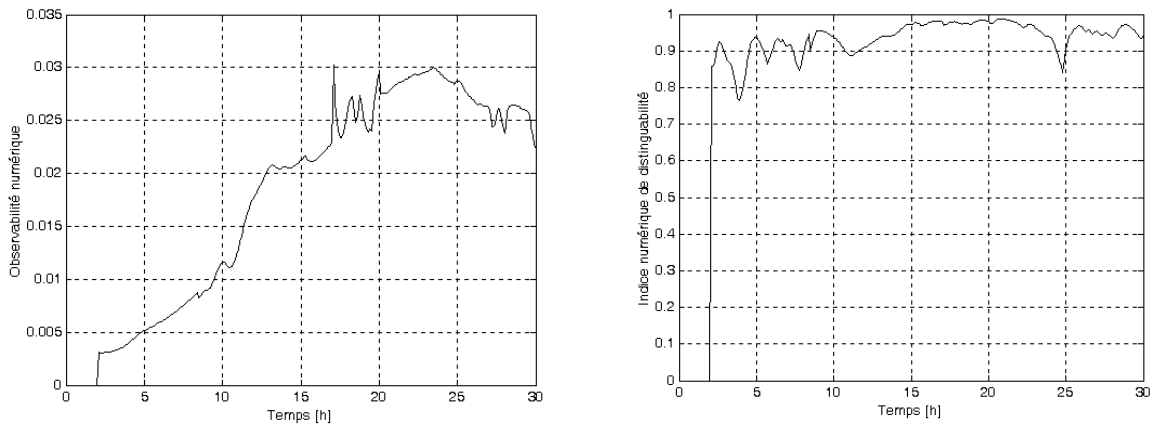


Figure 6.15 : Indicateurs d'observabilité numérique et de discernabilité

6.1.2.3 Analyse de résultats et conclusion cas c)

Vis à vis d'une perturbation sur l'état, la méthode IMHSE tient ses promesses. La convergence globale de l'algorithme d'estimation proposé permet de déterminer sur le domaine admissible le véritable état du système face à une perturbation. La divergence des méthodes classiques est due d'une part, dans le cas MHSE-BFGS, à l'impossibilité de la méthode d'estimation locale utilisée à trouver le minimum global. Ce minimum a été "déplacé par la perturbation", et la prédiction faite en partant de l'état avant la perturbation ne peut pas en tenir compte, ce qui montre les problèmes de convergence de la méthode locale d'estimation MHSE-BFGS. D'autre part, dans le cas du filtre de Kalman étendu face à une perturbation, le filtre est construit sur un bon modèle mais le procédé est maintenant erroné (par rapport au modèle) dû à la perturbation. Il apprend le mauvais état "trop bien" quand il traite les mesures : l'estimation ne présente aucun problème lorsque les bruits sur les mesures sont faibles. Dans ces conditions le filtre FKE est capable d'estimer l'état de façon très précise. La matrice de covariance devient très petite et par conséquent le gain devient très faible lui aussi, et les observations finissent par avoir peu d'effet sur l'état. Cependant quand le modèle dynamique du système n'est pas exactement celui utilisé dans la synthèse du filtre, l'état et son estimation divergent.

L'indicateur d'observabilité, nous indique qu'il a été nécessaire encore de faire un effort plus grand dans le début et la fin de l'expérience pour faire l'estimation de l'état. L'indicateur I_{ND} nous informe que les états ont quand même été bien discernés et donc que la méthode IMHSE a bien tenu compte de la perturbation sur l'état.

- d) *Avec perturbations sur un paramètre* : Nous pouvons maintenant prendre en compte le cas d'une perturbation sur un paramètre du modèle. Une perturbation classique sur un bioprocédé, est la diminution du paramètre μ_{\max} . Ce terme est sensible aux variations de pH, de température, etc. La perturbation sur ce paramètre est illustrée comme une diminution de 40% pendant 6 heures, entre les instants $k=14$ et $k=20$ heures. L'influence de cette perturbation sur un paramètre du système et ses estimations correspondantes (faites avec un bruit additif de 3%) peuvent être vues dans les figures 6.11 à 6.15.

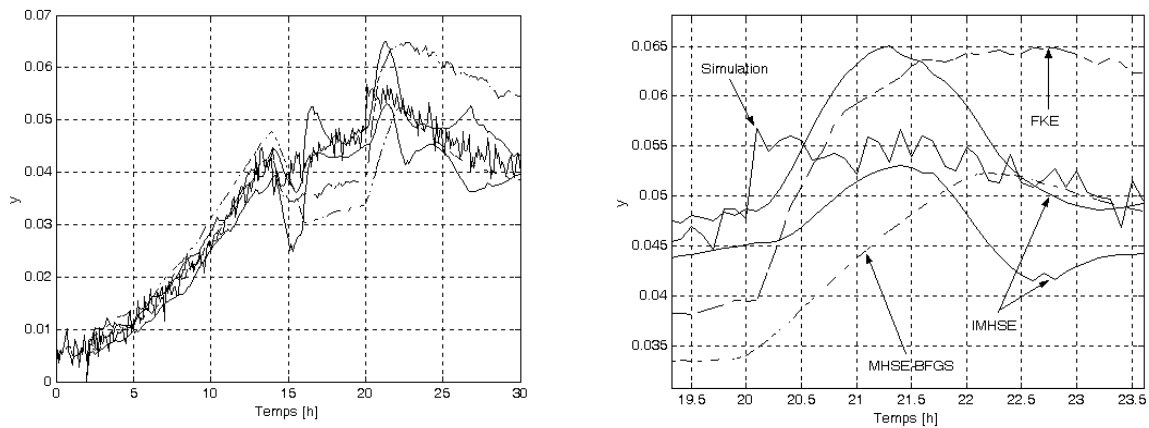


Figure 6.16 : Sortie bruitée considérée

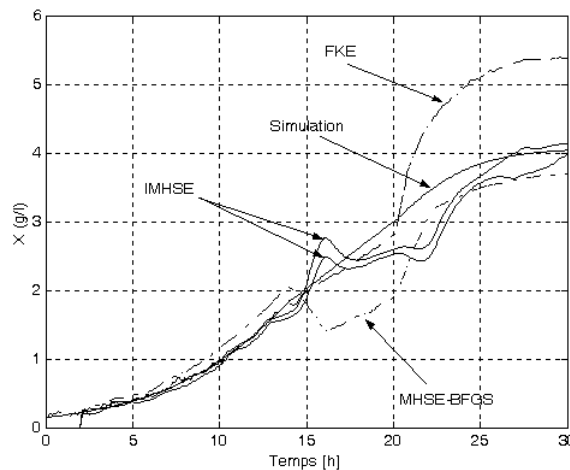


Figure 6.17 : Biomasse simulée (ligne continue) et estimations

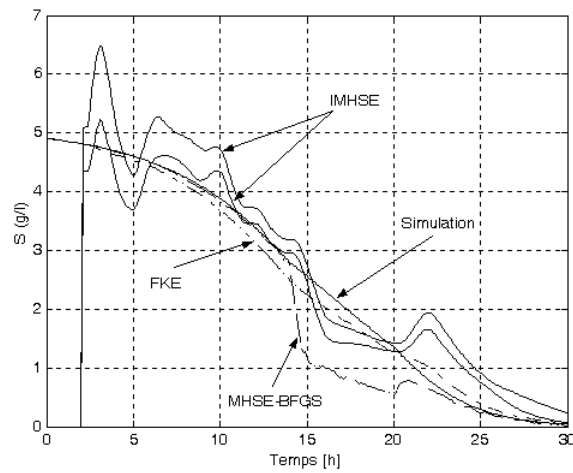


Figure 6.18 : Substrat simulé (ligne continue) et estimations

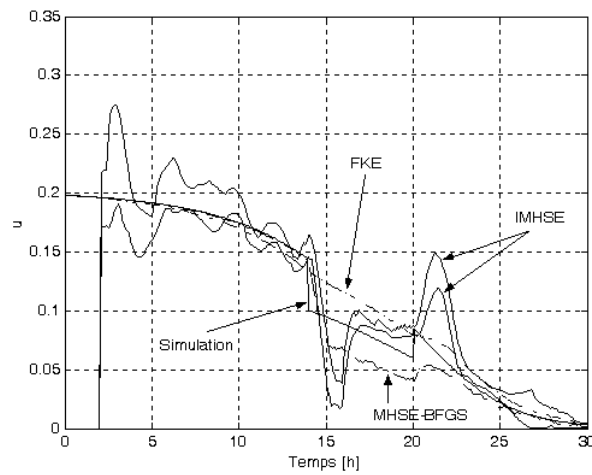


Figure 6.19 : Taux spécifique de croissance simulé (ligne continue) et estimations

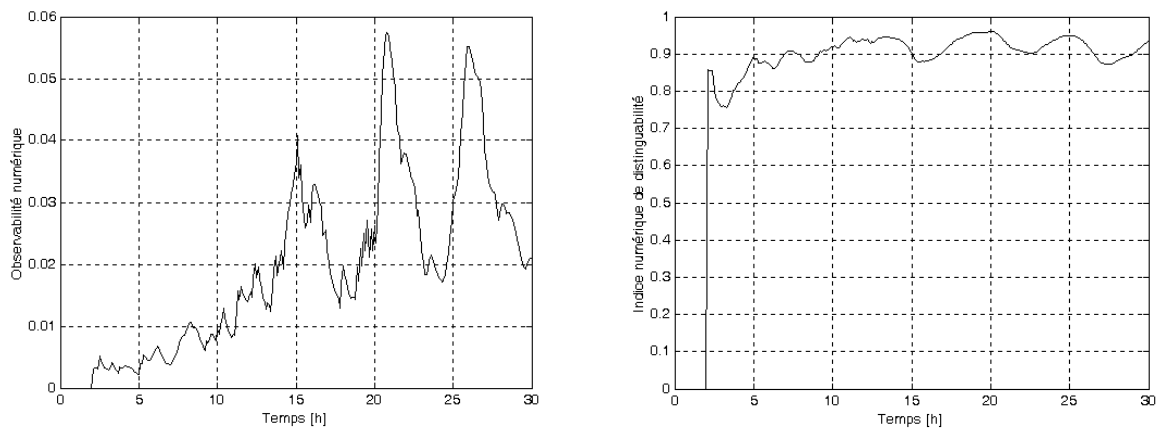


Figure 6.20 : Indicateurs d'observabilité numérique et de discernabilité

6.1.2.4 *Analyse de résultats et conclusion cas d)*

Les estimations montrent une performance satisfaisante des approches basées sur l'optimisation, c'est à dire de l'approche IMHSE et MHSE-BFGS. Cependant le FKE n'arrive pas de détecter la perturbation sur le paramètre. En général le FKE apprend le mauvais état "trop bien" quand il traite les mesures. Notons aussi que la méthode IMHSE détecte et suit bien les vrais trajectoires du système. En général la perturbation a des conséquences sur la fonction objectif, produisant une augmentation de sa valeur. La méthode d'optimisation globale cherche alors les états qui minimisent globalement le critère sur l'espace de recherche défini, ce qui permet en définitive de compenser la perturbation.

Nous voyons grâce aux indicateurs d'observabilité numérique et de discernabilité, qu'un effort informatique important a été réalisé, mais que les états ont été bien discernés : la méthode IMHSE a bien tenu compte de la perturbation sur le paramètre.

- e) *Avec des mesures hors ligne* : Supposons maintenant que nous disposons de mesures du substrat (l'état le moins observable). Ces mesures sont faites hors ligne et disponibles à des intervalles de deux heures. Supposons que suite à une intervention externe sur le réacteur à $k=15$ heures, le substrat est diminué de 50% de sa valeur de façon instantanée. L'influence de cette perturbation sur l'état du système et de ses estimations correspondantes, qui considèrent les mesures hors ligne sont illustrées dans les figures 6.21 à 6.26. Les méthodes MHSE-BFGS et FKE ne seront pas utilisées cette fois-ci, compte tenu des mauvaises performances montrées dans le cas précédent.

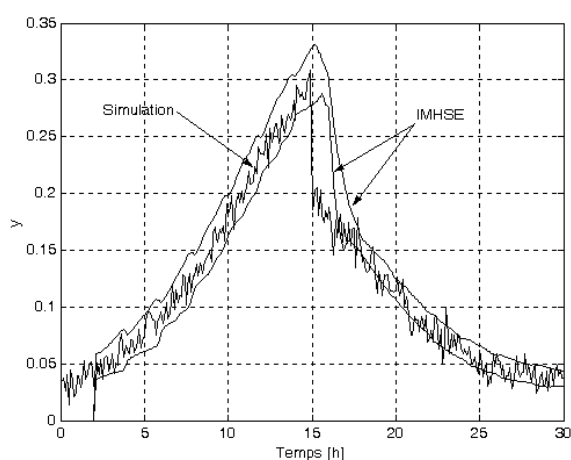


Figure 6.21 : Sortie bruitée considérée

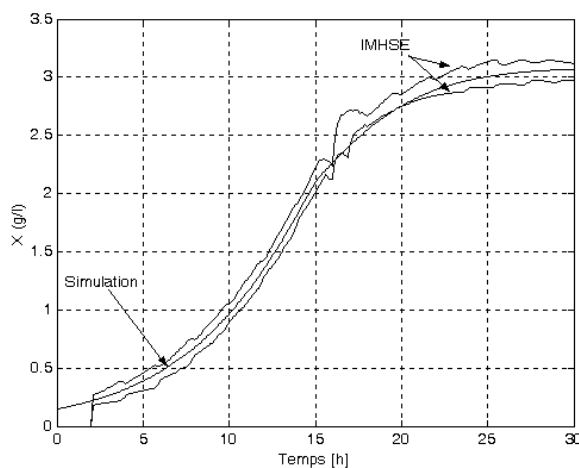


Figure 6.22 : Biomasse simulée (ligne continue) et estimations

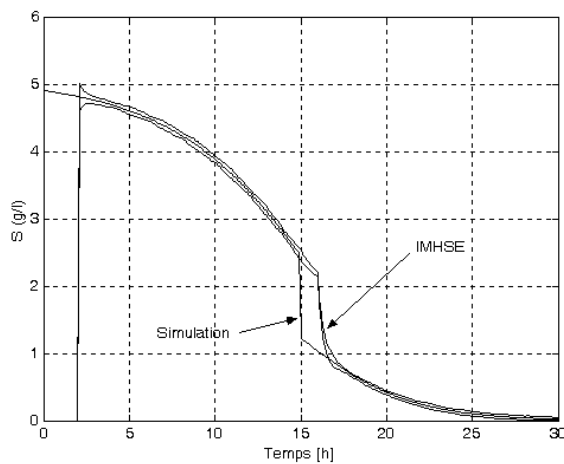


Figure 6.23 : Substrat simulé (ligne continue) et estimations

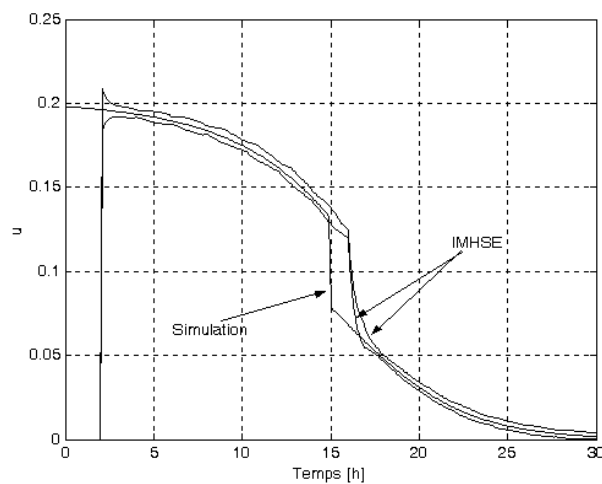


Figure 6.24 : Taux spécifique de croissance simulé (ligne continue) et estimations

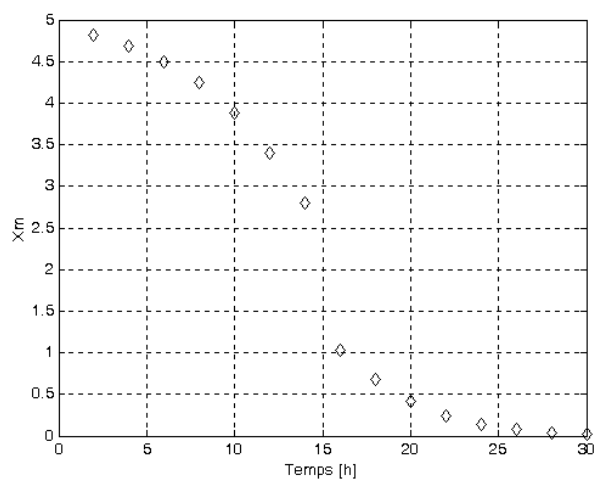


Figure 6.25 : Mesures hors ligne disponibles du substrat

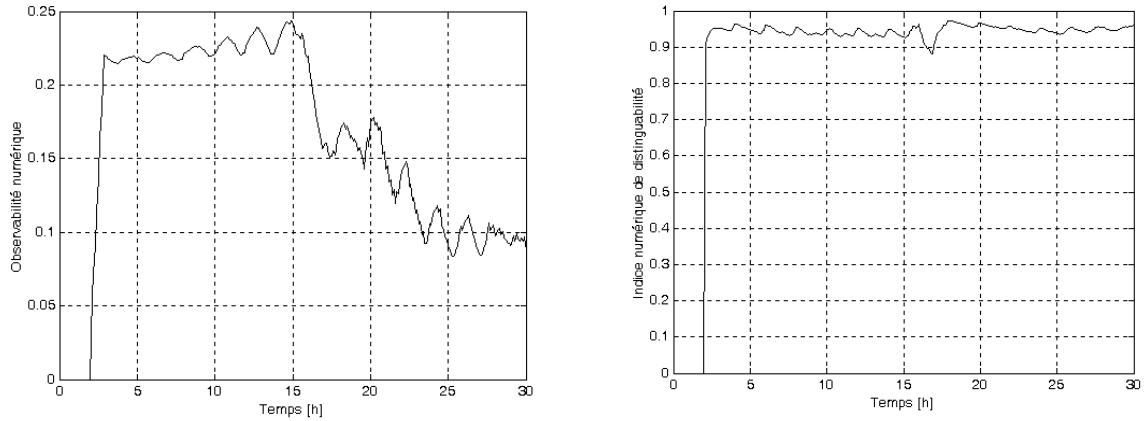


Figure 6.26 : Indicateurs d'observabilité numérique et de discernabilité

6.1.2.5 Analyse de résultats et conclusion cas e)

Nous avons vu dans le cas b), c) et d) que le substrat était l'état le moins observable. Cependant, quand nous considérons les mesures hors ligne de cette variable d'état, l'observateur peut caractériser sans problèmes la trajectoire de l'état. Ceci malgré la présence du bruit additif sur la sortie et d'une forte perturbation sur l'état. La méthode IMHSE tient donc aussi ses promesses et se montre performante vis à vis de ce type de problèmes, alors que la méthode locale MHSE-BFGS et le FKE ne savent pas le résoudre. Le temps de réponse se montre adéquat et l'inclusion des mesures hors lignes disponibles dans le critère d'optimisation rend visiblement plus performante la méthode proposée.

Ici les indicateurs d'observabilité π et I_{ND} nous donnent des informations très intéressantes. Le premier nous indique que le système est partout plus simple à observer, par rapport à l'effort informatique nécessaire dans les cas précédents. Le deuxième indicateur nous confirme que les états ont été bien distingués. Cela confirme que l'utilisation des informations contenues dans les mesures faites hors ligne sont d'une grande importance pour la méthode d'estimation IMHSE, au moment de distinguer les états. Les deux indicateurs nous confirment aussi que la méthode IMHSE a tenu plus facilement compte de la perturbation sur l'état.

- f) *Avec des mesures hors ligne* : Supposons aussi dans ce cas que nous disposons de mesures du substrat. Ces mesures sont faites hors ligne et disponibles à des intervalles de deux heures. Supposons aussi que suite à une intervention externe sur le réacteur à $k=15$ heures, la biomasse meurt et sa valeur tombe instantanément à 60% de sa valeur. L'influence de cette perturbation sur l'état du système et des estimations correspondantes, en considérant les mesures hors ligne sont montrées dans les figures 6.27 à 6.32.

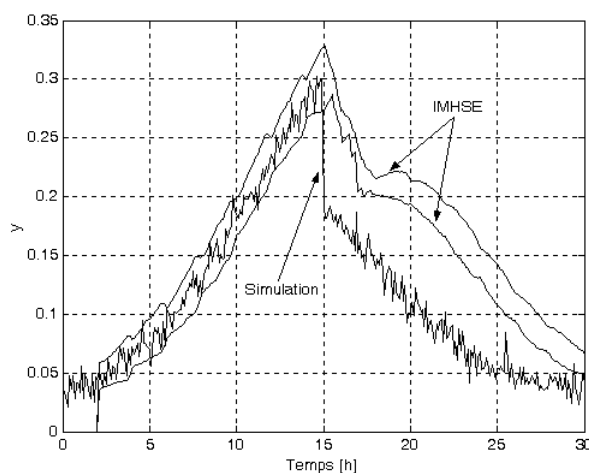


Figure 6.27 : Sortie bruitée considérée

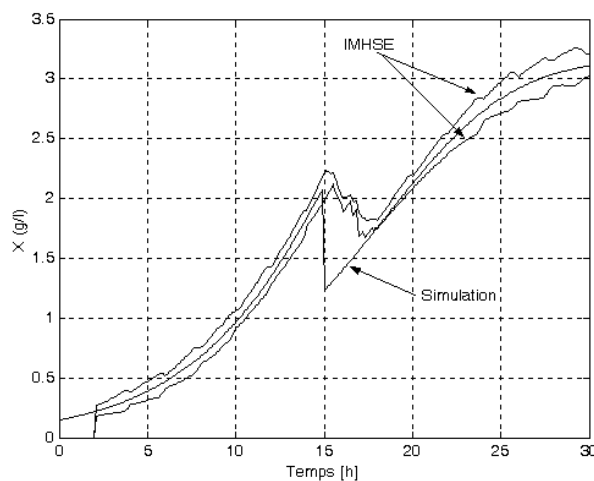


Figure 6.28 : Biomasse simulée (ligne continue) et estimations

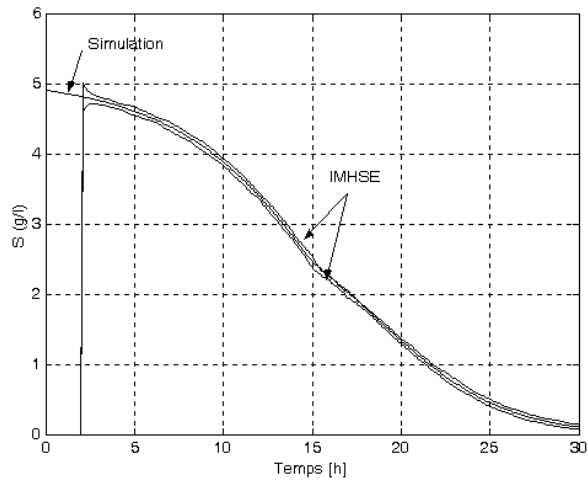


Figure 6.29 : Substrat simulé (ligne continue) et estimations

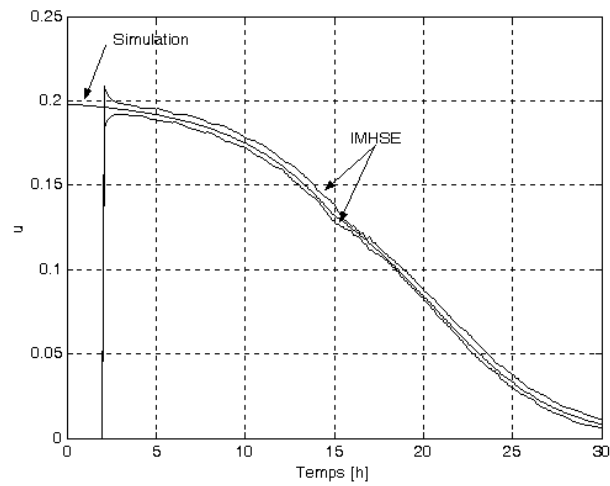


Figure 6.30 : Taux spécifique de croissance simulé (ligne continue) et estimations

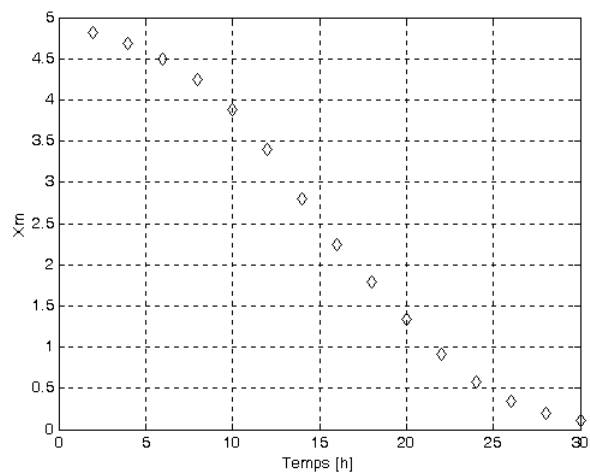


Figure 6.31 : Mesures hors ligne disponibles du substrat

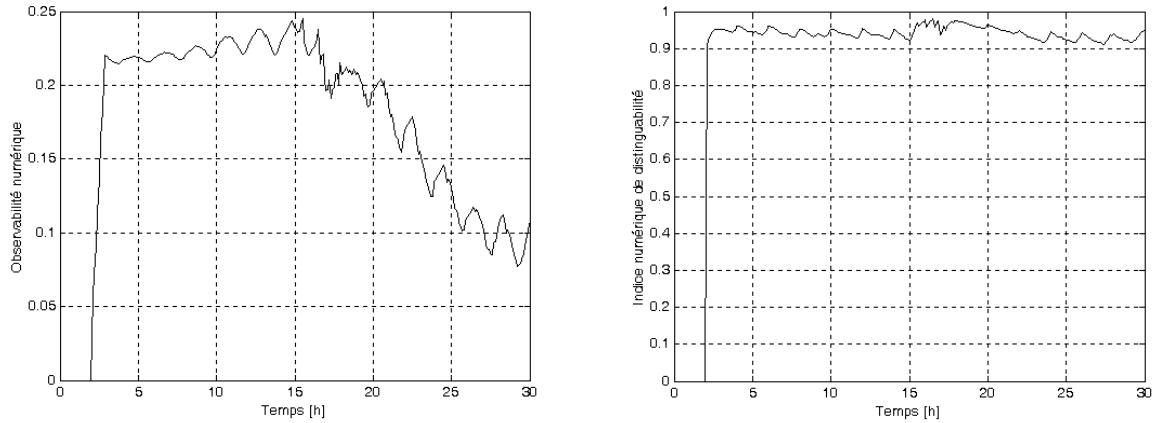


Figure 6.32 : Indicateurs d'observabilité numérique et de discernabilité

6.1.2.6 Analyse de résultats et conclusion cas f)

Cette application nous montre que la bonne performance illustrée par la méthode IMHSE dans le cas e) reste la même quand l'état perturbé est la biomasse. Le temps de réponse à la perturbation est aussi adéquat. Les estimations montrent une excellente cohérence entre les valeurs simulées et les résultats des estimations par intervalles des états en présence de bruit et des perturbations.

Comme dans le cas précédant, les indicateurs d'observabilité et de discernabilité nous apportent la confirmation d'une très bonne performance de la méthode IMHSE. Le premier confirme que le système est partout plus simple à observer, par rapport à l'effort informatique nécessaire pour les cas où les mesures hors ligne n'étaient pas considérées. Le deuxième indicateur nous confirme que les états ont été aussi cette fois-ci bien distingués. Les deux indicateurs nous confirment de la même façon qu'auparavant, que la méthode IMHSE a tenu plus facilement compte de la perturbation sur l'état biomasse.

6.1.2.7 Analyse générale des résultats et conclusion cas 1

Lors de cette première série de test, la méthode IMHSE a pu être validée avec un procédé simple de fermentation générique en simulation. Les comparaisons faites avec la méthode locale MHSE-BFGS et le FKE permettent d'extraire un certain nombre de conclusions, et de mettre en évidence les avantages de la méthode IMHSE.

Les performances obtenues avec la méthode à horizon glissant par intervalles sont meilleures qu'avec les méthodes locales utilisées en termes d'erreurs, en particulier lors de perturbations non prévues par le modèle interne de l'observateur. Nous pourrions

donc utiliser la méthode sur une plus vaste classe de procédés (ce que nous ferons dans les prochaines applications).

Nous pouvons remarquer que notre méthode nous permet de caractériser les "zones" dans lesquelles le système est moins observable, ce qui explique les résultats moins bons en début de l'expérience. Ainsi, la méthode IMHSE nous permet de fournir des estimations de l'état du système, mais aussi de fournir une explication pertinente lors de résultats moins performants.

Il faut noter en général, que la méthode MHSE-BFGS et le FKE peuvent obtenir meilleurs résultats en début de l'expérience, car leur initialisation est faite dans le voisinage des valeurs réelles.

Remarquons aussi l'avantage présenté par la méthode au niveau du calibrage, dont le seul réglage nécessaire reste la longueur de l'horizon. Ce choix est fonction du temps de réponse du système, ce qui n'est pas le cas du FKE. Au contraire, le FKE a besoin de réglages tels que Q et R, appelées matrices de covariance. De plus, lier leur réglage à un sens physique n'est pas évident, ce qui explique que le FKE soit délicat à régler.

Un tableau qui résume les temps de calcul réalisés pour chaque cas étudié précédemment est montré dans le tableau suivant, pour un PC Pentium II 450 Mhz, 128 Mb RAM :

Tableau 4 : Temps de calcul des cas considérés

Cas / Temps		Cas / Temps	
a)	< 4.0 heures	e)	< 40 minutes
b)	< 5.0 heures	f)	< 40 minutes
c)	< 5.0 heures		
d)	< 5.0 heures		

Le tableau 4 montre que l'utilisation de mesures hors ligne, souvent disponibles dans les bioprocédés, simplifie énormément le problème d'estimation. De plus, l'utilisation de cette information diminue de façon substantielle les temps de calcul à plus de 86%.

6.2 Cas 2 : Application à un bioprocédé générique décrit par un modèle hybride

Cet exemple d'application considère le même procédé de fermentation générique donnée dans la section 6.1, et dont le modèle es présenté par (6.1)-(6.3). De manière classique, le taux spécifique de croissance de la biomasse, μ_k ou $\mu_k(S_k)$, a une évolution qui est en général difficile à caractériser. Les difficultés de la représentation de ce paramètre ont amené à proposer différents types de lois pour modéliser son comportement en fonction des procédés (Monod, Haldane,...).

Nous allons proposer une approche hybride à trois places, pour modéliser la relation existante entre ce paramètre et le substrat. L'objectif que nous poursuivons est de faire l'estimation avec la méthode IMHSE dont le modèle interne de l'observateur sera en partie hybride. Nous présenterons ensuite les résultats des trajectoires suivies par ce système évoluant suivant l'inclusion de chaque place.

6.2.1 Modèle du système

On suppose pour cet exemple que le modèle discret non-linéaire qui décrit le bioprocédé dans la section 6.1 reste invariant, ainsi que le modèle de sortie, et donc la biomasse et le substrat. Ils sont décrit par (6.1).

Nous allons utiliser un modèle hybride, pour représenter la dépendance entre le taux spécifique de croissance de la biomasse et le substrat. Nous cherchons à simplifier la représentation utilisée, en particulier pour des paramètres généralement difficiles à identifier comme le taux spécifique de croissance de la biomasse.

La loi qui caractérise son évolution est représentée par une fonction affine par morceaux, voir figure 6.33. Trois places l_i détermineront ces segments comme suit :

$$\begin{aligned}
 l_1 : 0 \leq S_k \leq 2 &\Rightarrow & \mu_k &= 0 \\
 l_2 : 2 < S_k \leq 9 &\Rightarrow & \mu_k &= 0.04 \cdot S_k - 0.08 \\
 l_3 : S_k > 9 &\Rightarrow & \mu_k &= 1.5 \cdot 10^{-3} \cdot S_k + 0.28
 \end{aligned} \tag{6.4}$$

Ce modèle est aussi donné du point de vue des conditions de transition (arcs) entre les trois places (l_1 , l_2 et l_3), comme l'ensemble de la figure 6.34.

Nous travaillerons alors avec un modèle commuté, dont les états du système sont représentés par des intervalles. Cependant il faut remarquer que l'état de l'automate hybride sous ces conditions, n'est pas donné à chaque instant simplement comme une paire $(l_i, [x])$. La principale difficulté provient du fait que différentes places peuvent être requises en même temps par un même état intervalle. Cette sollicitation multiple est illustrée dans la figure 6.35, où les états (biomasse et substrat) sont concernés par trois places du modèle hybride de façon simultanée.

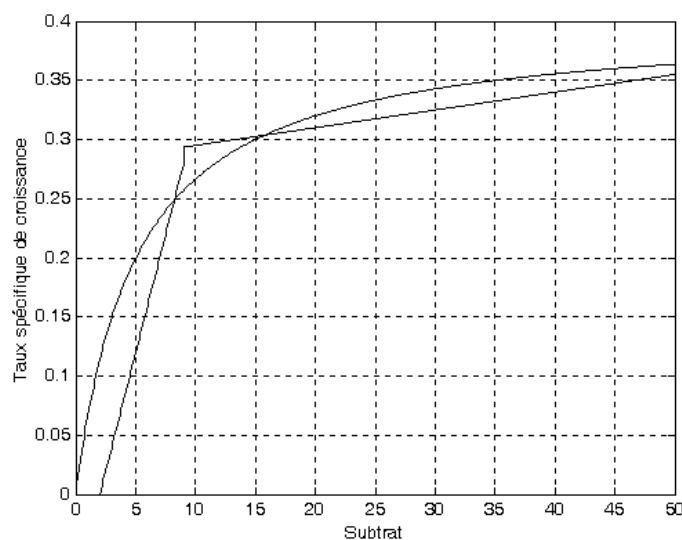


Figure 6.33 : Loi hybride entre le taux spécifique de croissance et le substrat

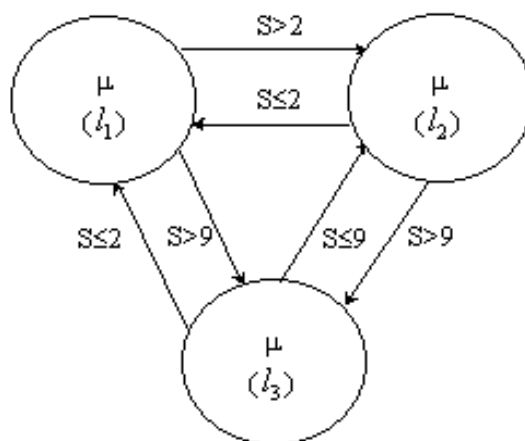


Figure 6.34 : Conditions de transitions du modèle hybride

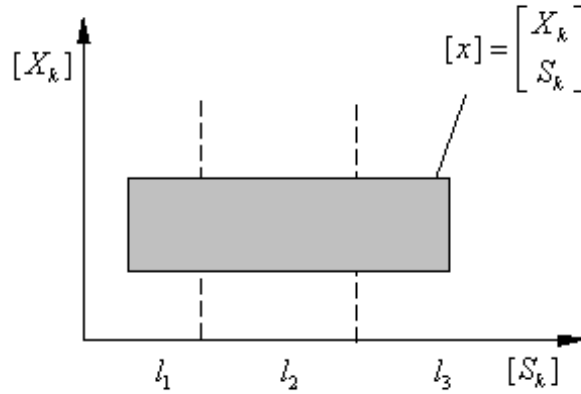


Figure 6.35 : État intervalle dont plusieurs zones de commutation sont concernées

Quand plus d'une place est concernée pour l'état intervalle, la méthode doit opérer plusieurs conditions de transition en même temps. Ce qui implique que tous les cas possibles doivent être analysés.

Cette analyse est faite par l'union de tous les états possibles de l'automate qui sont concernés. Dans ce cas, le modèle (2.19), qui considère les trois possibles places peut être réécrit comme suit :

$$\Sigma_i : \begin{cases} x(k+1) = \bigcup_{i=1}^3 f_i(x(k), u(k)) \\ y(k) = h(x(k)) \end{cases} \quad (6.5)$$

6.2.2 Résultats

Les conditions générales du test pour l'application de IMHSE restent identiques à celles de la section 6.1.1 du cas 1. Il faut remarquer qu'avec cette application nous ne cherchons pas à étudier le système hybride décrit, sous des conditions de bruit et des perturbations. Cependant nous cherchons à établir une procédure d'évaluation d'un modèle du type (6.5). Ceci parce que dans l'application 4 de ce chapitre, nous travaillerons avec un modèle issu d'un bioprocédé réel, dont l'état nommé azote intermédiaire, a été modélisé de façon hybride.

- a) *Pas de perturbations* : Les résultats des estimations sur la biomasse, le substrat et le taux spécifique de croissance sont présentés dans les figures 6.36 à 6.38 respectivement [Valdés-González et Flaus, 2001b].

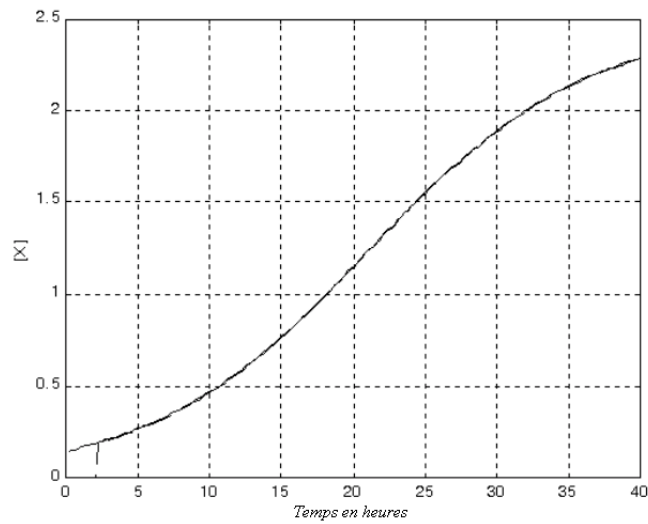


Figure 6.36 : Biomasse simulée (ligne continue) et estimée (lignes intervalles)

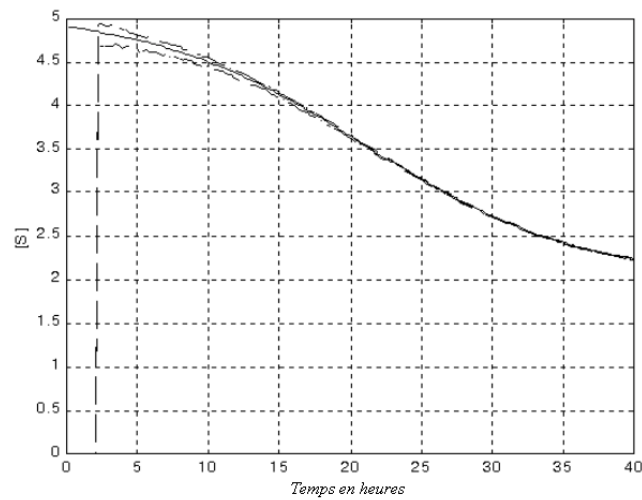


Figure 6.37 : Substrat simulé (ligne continue) et estimé (lignes intervalles)

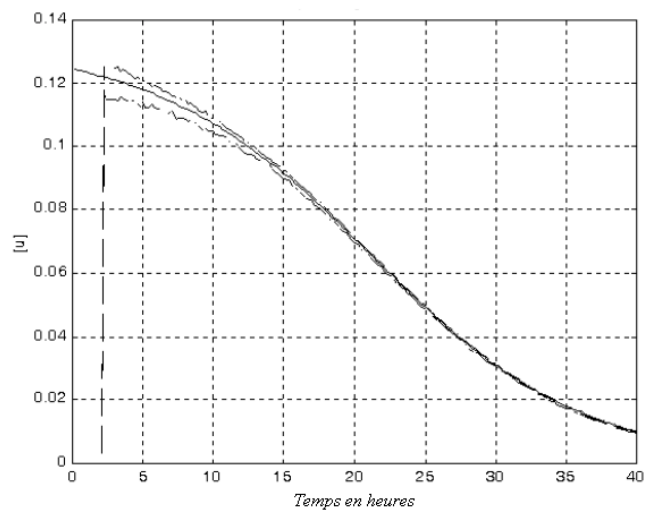


Figure 6.38 : Taux spécifique de croissance simulé (ligne continue)
et estimé (lignes intervalles)

6.2.2.1 *Analyse de résultats et conclusion cas 2*

Les estimations montrent une excellente cohérence entre les valeurs simulées et les résultats des estimations par intervalles de la biomasse, le substrat et le taux spécifique de croissance. Ceci nous permet de conclure que la méthode d'estimation IMHSE peut accueillir facilement des modèles hybrides. La procédure d'évaluation d'un état à intervalles dont les différentes places sont susceptibles d'être évaluées simultanément a été réalisée de façon satisfaisante. Remarquons que les intervalles contiennent de façon garantie la solution globale du problème d'estimation optimal en début d'horizon quand le modèle impliqué est décrit de façon hybride.

6.3 Cas 3 : Application à la détection de systèmes non-observables ou faiblement observables

Dans cet exemple d'application nous allons nous concentrer sur les capacités de la méthode IMHSE pour détecter des systèmes qui ne sont pas observables (voir chapitre 4), ou faiblement observables (remarque 18).

Il s'agit en effet d'étudier l'ensemble solution $[\hat{\Theta}]_{sh}$ du problème global d'optimisation (2.10), en début d'horizon pour une précision d'optimisation donnée [Valdés-González et Flaus, 2002a].

Considérons une fois de plus le système proposé pour le cas 1, les équations (6.1)-(6.3), sous les mêmes conditions de simulation.

- a) *Système observable* : Le système décrit dans le cas 1 (6.1)-(6.3), dont nous rappelons les équations :

$$\begin{aligned} X_{k+1} &= X_k \cdot (1 + \Delta t \cdot \mu_k) & \mu_k &= \mu_{\max} \cdot \frac{S_k}{k_s + S_k} & y_k &= (a \cdot \mu_k + b) \cdot X_k \\ S_{k+1} &= S_k - \Delta t \cdot \frac{\mu_k}{\gamma} \cdot X_k \end{aligned}$$

est un système à première vue observable. La figure suivante nous montre l'ensemble $[\hat{\Theta}]_{sh}$ sous les conditions décrites et pour une précision d'optimisation $B_\varepsilon = 10^{-4}$.

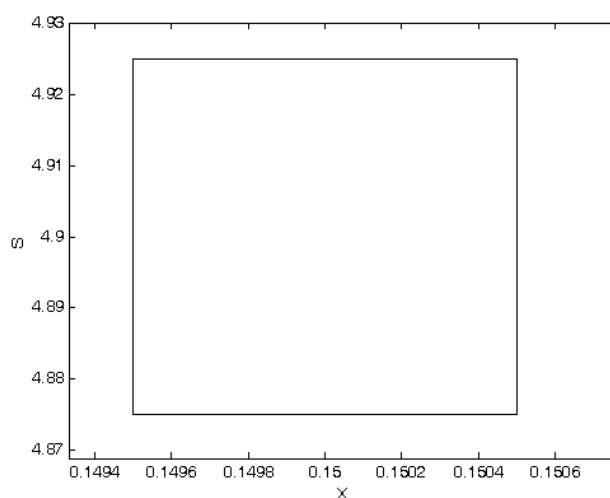


Figure 6.39 : Ensemble $[\hat{\Theta}]_{sh}$ (biomasse et substrat)

En d'autres termes, l'ensemble solution est égal à :

$$[\hat{\Theta}]_{sh} = \begin{bmatrix} [0.149, 0.150] \\ [4.875, 4.925] \end{bmatrix} \text{ sur un domaine admissible égal à } [\Omega] = \begin{bmatrix} [0, 4.5] \\ [0, 5.5] \end{bmatrix}$$

b) *Système non observable (une infinité de solutions)* : Considérons tout de même le système proposé pour le cas 1, équations (6.1)-(6.3), sous les mêmes conditions de simulation. Cette fois ci, la dépendance entre le taux spécifique de croissance de la biomasse et le substrat ; l'équation (6.2), sera définie selon la loi hybride suivante :

$$\begin{aligned} &IF \quad S_k < 2.5 \\ &\quad \mu_k = 0.064 \cdot S_k \\ &ELSE \\ &\quad \mu_k = 0.16 \\ &ENDIF \end{aligned} \tag{6.6}$$

Avec cette formulation, nous cherchons à rendre non observable, le système impliqué. La figure 6.40 nous montre l'ensemble $[\hat{\Theta}]_{sh}$ sous les conditions décrites et pour une précision d'optimisation $B_\varepsilon = 0.01$, dont l'état substrat ne peut pas être discerné.

Remarquons qu'avec la formulation hybride (6.6), nous pouvons toujours tenter d'utiliser un FKE ou la méthode MHSE-BFGS, mais ceux-ci donneront des résultats erronés parce que le système n'est pas observable (voir figure 6.41).

Le problème d'observabilité est dû au fait qu'il y a un grand nombre d'états indiscernables différents comme l'illustre la figure 6.40 (le substrat est indistinguable). De plus, comme FKE et la méthode MHSE-BFGS sont initialisés dans le voisinage des valeurs réelles de simulations, les résultats suivants sont possibles.

Cependant, pour les estimations montrées dans la figure 6.41, une perturbation sur l'état (biomasse) va montrer immédiatement que le système n'est pas observable (pas de convergence), et donc que les estimations vont diverger comme le montre la figure 6.42.

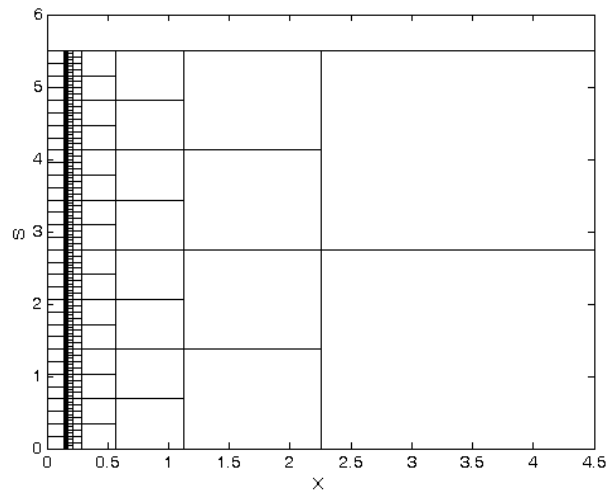


Figure 6.40 : Ensemble $[\hat{\Theta}]_{sh}$ d'états indiscernables déterminés en début d'horizon

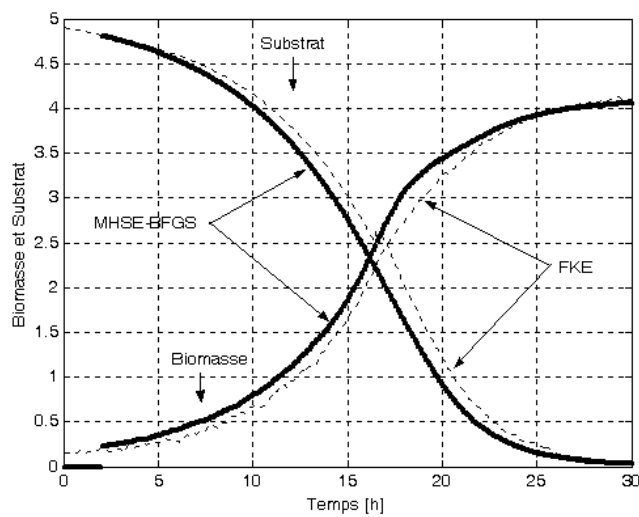


Figure 6.41 : Concentration de biomasse et substrat estimés par FKE (lignes segmentés) et par MHSE-BFGS (grosses lignes continues).

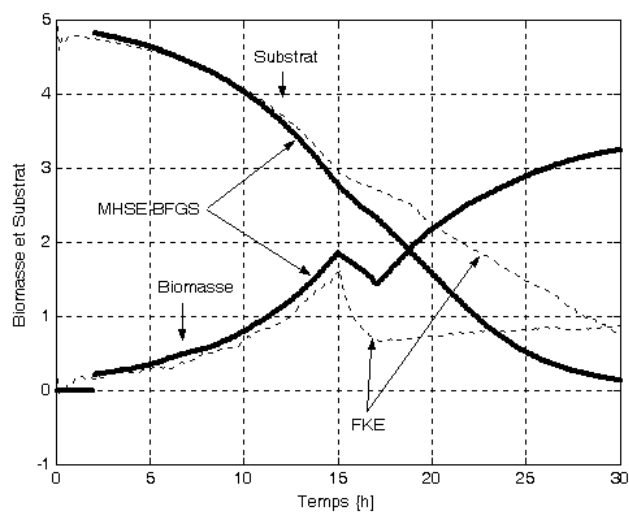


Figure 6.42 : Divergence des estimations face à une perturbation dans la biomasse

c) *Système non observable (deux solutions différentes) :*

Cet exemple d'application considère un système non linéaire discret à deux états définis comme suit :

$$\begin{aligned} x_1(k+1) &= x_1(k) + \Delta t \cdot x_2(k) \\ x_2(k+1) &= x_2(k) - 0.2 \cdot \Delta t \cdot x_1(k) - \Delta t \cdot x_2(k) \cdot (x_1^2(k) - 1) \end{aligned} \quad (6.7)$$

$$y_1 = x_1(k) \cdot x_2(k) \quad (6.8)$$

- Conditions du test :

La longueur d'horizon : 2 heures ; Période d'échantillonnage: 0.1 h ; Précision d'optimisation : $B_\varepsilon = 0.01$; *Etat initial pour la simulation* : $x_0 = [0.5, 0.3]^T$.

Domaine admissible :

$$[\Omega] = \begin{bmatrix} -3 & 3 \\ -4 & 4 \end{bmatrix}$$

La figure 6.43 nous montre l'ensemble $[\hat{\Theta}]_{sh}$, qui contient deux états indiscernables différents. Ceci nous amène à conclure que le système n'est pas observable.

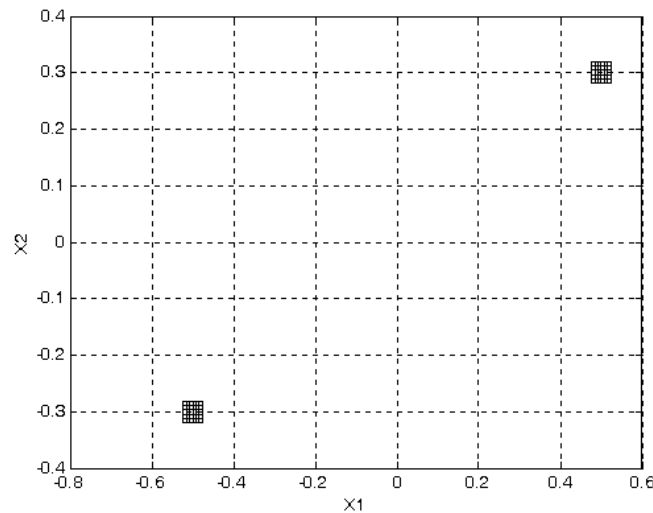


Figure 6.43 : Ensemble $[\hat{\Theta}]_{sh}$ d'états indiscernables en début d'horizon

6.3.1 Effort nécessaire pour distinguer certains états

Bien que cette méthode détecte sans soucis les problèmes d'observabilité d'un système, il existe certains cas où il est très difficile distinguer certains états [Othman, 1992], [Hermann et Krener, 1977]. Dans ce type de cas nous pourrions être tentés de classer le système comme non-observable, alors qu'il l'est bien en réalité (c'est pour ça que nous avons défini l' ϵ -observabilité). Pour illustrer ce cas, considérons encore le système (6.1)-(6.3) du cas 1. Nous avons classé ce système comme un système observable (voir la section 6.3.a et les figures 6.1-6.4. Comme il avait déjà été expliqué pour ce système, le substrat S_k est très difficile à distinguer (particulièrement quand le substrat a une valeur élevée). Les résultats suivants montrent le problème de distinction d'état, pour différentes précisions d'optimisation dans la méthode IMHSE.

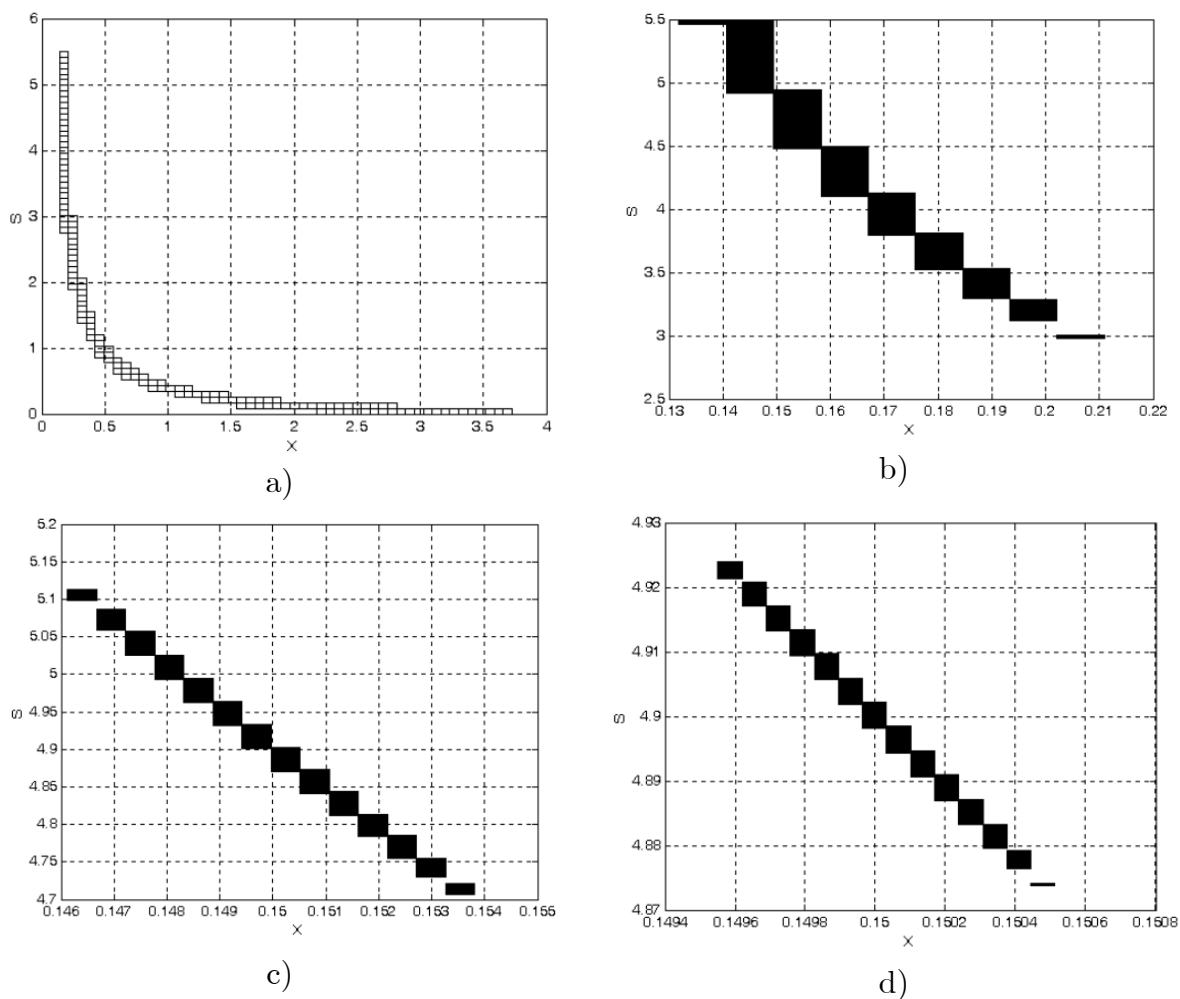


Figure 6.44 : Ensemble $[\hat{\Theta}]_{sh}$ d'états indiscernables en début d'horizon ($k=0$ hrs) pour une précision de a) $B_\epsilon = 0.1$, b) $B_\epsilon = 0.01$, c) $B_\epsilon = 0.001$, d) $B_\epsilon = 0.0001$

L'effort informatique est montré dans le tableau suivant, pour un PC Pentium II 450 Mhz, 128 Mb RAM:

Tableau 5 : Temps de calcul pour le problème (2.10)

	Précision / Temps	Valeur obtenue de $[\hat{\Theta}]_{sh}$
10^{-1}	19.1 secondes	$[\hat{\Theta}]_{sh} = \begin{bmatrix} [0, 3.7] \\ [0, 5.5] \end{bmatrix}$
10^{-2}	3.4 minutes	$[\hat{\Theta}]_{sh} = \begin{bmatrix} [0.13, 0.21] \\ [3.0, 5.5] \end{bmatrix}$
10^{-3}	12.2 minutes	$[\hat{\Theta}]_{sh} = \begin{bmatrix} [0.146, 0.154] \\ [4.7, 5.12] \end{bmatrix}$
10^{-4}	19.8 minutes	$[\hat{\Theta}]_{sh} = \begin{bmatrix} [0.14955, 0.1505] \\ [4.874, 4.924] \end{bmatrix}$

6.3.1.1 Analyse de résultats et conclusion cas 3

Cette technique d'estimation permet de déterminer si un système est observable pour les conditions imposées, et en particulier pour une précision d'optimisation déterminée et pour une borne de tolérance imposée au critère.

L'étude de l' ε -observabilité en début de l'horizon nous permet de classer un système comme observable, même si ses états sont difficiles à discerner. Remarquons aussi que dans certains cas, il est très difficile de distinguer certains états, et donc un effort informatique important doit être effectué pour réaliser ce travail.

Ceci rend notre observateur très performant et lui donne une grande fiabilité, en nous permettant d'utiliser les estimations avec toute la prudence nécessaire. De plus, cette caractéristique peut s'avérer très utile lorsque le modèle ou des parties du modèle sont mal connus. Cette caractéristique a une influence non négligeable. En effet, rappelons qu'il est difficile de vérifier que pour toute entrée, le système sera partout localement observable. La détection de l' ε -observabilité pas à pas est faite en regardant l'ensemble $[\hat{\Theta}]_{sh}$. Cet ensemble est la solution au problème global d'optimisation sur un horizon de temps du problème d'estimation IMHSE.

6.4 Cas 4 : Application à un modèle de fermentation sur substrat solide

Cette étude montre l'application de la méthode IMHSE utilisée sur un procédé de production d'Acide Gibbéréllique (SSC - Solid Substrate Cultivation) [Valdés-González *et al.*, 2002b]. Il s'agit d'une hormone végétale produite par le champignon filamenteux *Gibberella Fujikuroi*. Le champignon se développe grâce à un procédé de fermentation sur substrat solide dans un support inerte. Un modèle simple qui décrit l'évolution des principales variables du procédé batch est présenté en [Gelmi, 1999] et [Pérez-Correa et Agosin, 1999]. Ce modèle phénoménologique présente six variables d'états :

- X : Biomasse vivante
- X_m : Biomasse mesurée
- U : Urea
- NI : Azote intermédiaire
- CO_2 : Gaz produit
- O_2 : Gaz consommé

Seules les deux dernières variables peuvent être mesurées directement en ligne par intégration. Les équations qui décrivent le modèle non linéaire discret sont les suivantes :

$$X_{k+1} = X_k \cdot (1 + \Delta t \cdot (\mu_k - k_d)) \quad (6.9)$$

$$Xm_{k+1} = Xm_k + \Delta t \cdot \mu_k \cdot X_k \quad (6.10)$$

$$U_{k+1} = U_k - K \cdot \Delta t \quad (6.11)$$

$$NI_{k+1} = NI_k + \Delta t \cdot 0.47 \cdot K - \Delta t \cdot \mu_k \cdot \frac{X_k}{Y_{X/NI}} \quad (6.12)$$

$$CO_{2(k+1)} = CO_{2(k)} + \Delta t \cdot \mu_k \cdot \frac{X_k}{Y_{X/CO_2}} + \Delta t \cdot m_{CO_2} \cdot X_k \quad (6.13)$$

$$O_{2(k+1)} = O_{2(k)} + \Delta t \cdot \mu_k \cdot \frac{X_k}{Y_{X/O_2}} + \Delta t \cdot m_{O_2} \cdot X_k \quad (6.14)$$

Cependant si $U_k < 0$, alors (6.12) n'est plus valide et nous devons commuter l'équation de l'azote intermédiaire par (6.15), de plus $U_k = 0$. Il s'agit donc d'un automate hybride avec deux places comme il est montré dans la figure 6.45, où sont résumées aussi les conditions de transition.

$$NI_{k+1} = NI_k - \Delta t \cdot \mu_k \cdot \frac{X_k}{Y_{X/NI}} \quad (6.15)$$

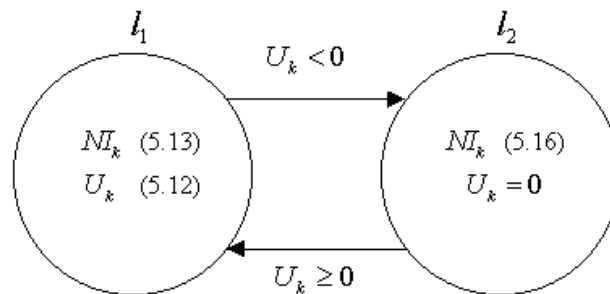


Figure 6.45 : Conditions de transitions du modèle hybride pour NI_k

Où μ_k correspond au taux spécifique de croissance de la biomasse, et sa dépendance à l'azote intermédiaire est modélisé par une loi de Monod :

$$\mu_k = \mu_m \cdot \frac{NI_k}{k_n + NI_k} \quad (6.16)$$

6.4.1 Conditions générales du test

Les paramètres du modèle sont identifiés d'après des expériences ou des considérations expérimentales spécifiques [Gelmi, 1999]. Leurs valeurs peuvent être trouvées dans le tableau 6, sous des conditions contrôlées de température et d'activité de l'eau ($T=25^\circ\text{C}$, $A_w=0,992$).

Tableau 6 : Paramètres du modèle

	Nom /	Description	Valeur
μ_m		Taux spécifique de croissance maximale	0.21 (h^{-1})
K		Constante de dégradation de la Urea	$1.34 \cdot 10^{-4}$ (h^{-1})
k_n		Constante	$78.46 \cdot 10^{-5}$
k_d		Taux de mortalité	$26.60 \cdot 10^{-3}$ (h^{-1})
m_{CO_2}		Coefficient de conservation du CO_2	0.13
m_{O_2}		Coefficient de conservation du O_2	$54.40 \cdot 10^{-3}$
$Y_{X/NI}$		Coefficient de production	21
Y_{X/CO_2}		Coefficient de production	1.15
Y_{X/O_2}		Coefficient de production	2.60

Remarque 23: Dans ce travail, la méthode utilisée suppose que le modèle non-linéaire est connu exactement, c'est à dire qu'il ne considère pas les incertitudes du modèle. Cependant, l'arithmétique par intervalles nous permet de définir (si besoin), les paramètres d'un modèle comme un intervalle. Ceux-ci pourront donner un autre type de représentation des incertitudes du modèle réel.

Par conséquent, les intervalles qui contiennent les vraies valeurs de l'état estimé seront élargis (largeurs indiscernables). De cette façon les valeurs expérimentales mesurées pourront rester à l'intérieur de l'état intervalle estimé. Cependant il est possible que les valeurs expérimentales mesurées soient à l'extérieur des intervalles, parce que dans la pratique, par exemple dans l'estimation sur des bioprocédés, la différence entre les estimations et les vraies valeurs des variables d'états sont dues aux erreurs du modèle, mais aussi à la structure du modèle qui est souvent une approximation brutale de la réalité.

Le modèle non-linéaire discret du bioprocédé présenté a été simulé comme décrit dans les équations (6.9)-(6.16), pour générer les sorties du système (la production de CO_{2k} et la consommation O_{2k}). Ces sorties seront les mesures d'entrée de notre

observateur. Le reste des variables de l'état a été simulé pour comparer la performance de l'observateur par rapport aux états estimés.

L'expérience dure 60 heures, et pour appliquer la méthode IMHSE, les mesures sont disponibles tout au long de l'horizon de temps considéré. Ce dernier est de $lh=2$ heures. Le nombre de mesures sur l'horizon du temps considéré est 21 avec une précision d'optimisation de $B_e = 0.01$.

6.4.2 Résultats

Les résultats (simulation hors ligne) peuvent être vus dans les figures 6.46-6.50, où les estimations par des intervalles de X_k , X_{mk} , NI_k , et μ_k sont illustrés. Une application sur ce procédé, avec MHSE couplée à un algorithme d'optimisation heuristique, appelé MHSE-SA est montré dans l'annexe E. Notons que comme la variable X_{mk} est mesurée expérimentalement au niveau du laboratoire, elle est aussi simulée sur intégration depuis l'estimation de X_k (disponible seulement après $k=2$ heures). Les estimations par intervalles des variables et leurs valeurs expérimentales correspondantes sont graphiquement représentés ensemble seulement à des fins de comparaison (voir la remarque 23).

Par ailleurs, U_k est simulé comme une variable interne pour notre observateur, parce que cette variable est nécessaire pour faire les transitions du modèle hybride (commutation du modèle NI_k).

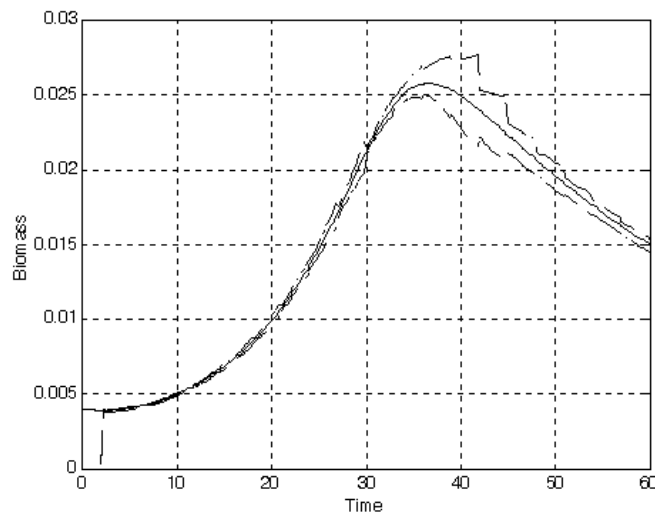


Figure 6.46 : Biomasse simulée (ligne continue) et estimée (lignes intervalles)

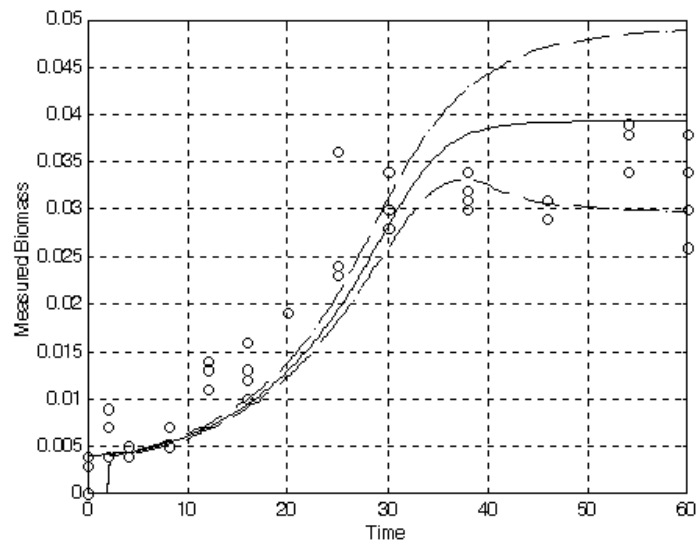


Figure 6.47 : Biomasse mesurée simulée (ligne continue), estimée-intégrée (lignes intervalles) et (o) réelle

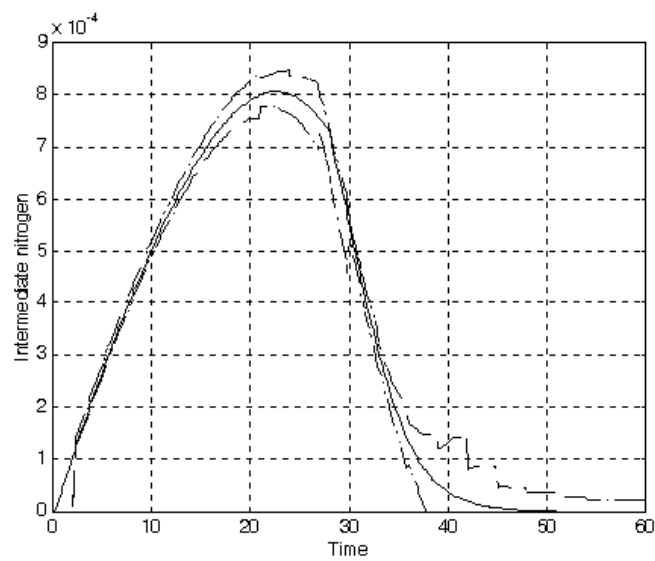


Figure 6.48 : Azote intermédiaire simulé (ligne continue) et estimé (lignes intervalles)

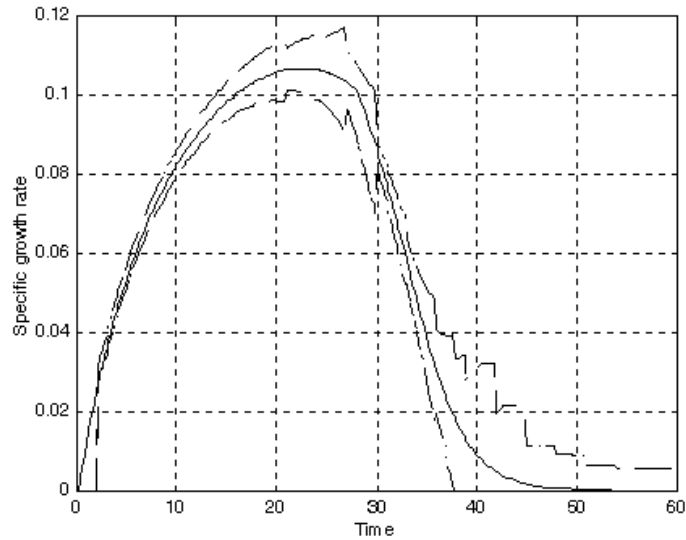


Figure 6.49 : Taux spécifique de croissance simulé (ligne continue) et estimé (lignes intervalles)

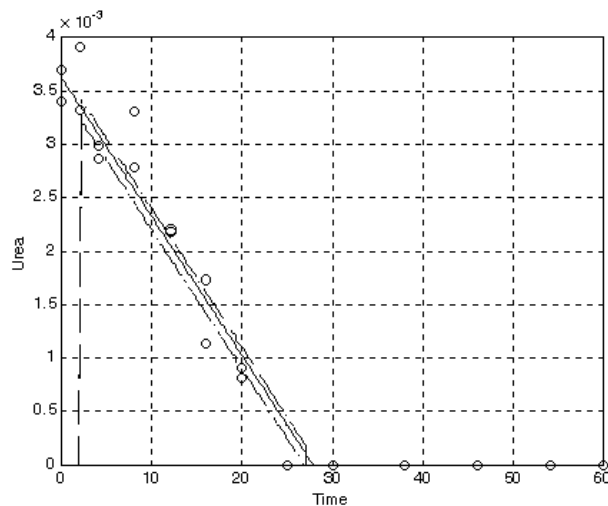


Figure 6.50 : Urea simulé (ligne continue) et (o) mesuré

6.4.2.1 Analyse de résultats et conclusion cas 4

Il y a une bonne cohésion entre les états simulés X_k , NI_k et μ_k , leurs valeurs estimées et les mesures expérimentales disponibles. Ce qui confirme les bonnes performances de la méthode IMHSE proposée, sur un modèle de bioprocédé issu d'un système réel. La comparaison des estimations avec des résultats expérimentaux montre la pertinence des intervalles indiscernables obtenus. Ces mesures hors-ligne permettent une comparaison éloquent sur ce procédé de fermentation sur substrat solide, difficile à modéliser et à estimer.

6.5 Cas 5 : Détection de dysfonctionnement par observateur ensembliste multi-modèles

Nous appliquerons la méthode IMHSE pour détecter les dysfonctionnements non prévus par les modèles internes de l'observateur, représentés par des variations dynamiques de ses paramètres. Ceci est pris en compte en utilisant un ensemble de plusieurs modèles différents qui commutent pour reconstruire les états du système. Pour appliquer notre méthode nous utiliserons un modèle de Van Der Pol.

6.5.1 Description du système

Considérons comme système le modèle de VAN-DER-POL suivant :

$$\begin{aligned}\dot{x}_1 &= x_2 \cdot (1 + \mathcal{G} \cdot \xi_1) \\ \dot{x}_2 &= \left(-0.2 \cdot (x_1^2 - 1) \cdot x_2 - x_1\right) \cdot (1 + \mathcal{G} \cdot \xi_2) \\ y &= x_1\end{aligned}\tag{6.17}$$

Le modèle (6.17) prend en compte les paramètres fixes ξ_1 et ξ_2 , et la variable aléatoire $\mathcal{G} \in [-1, 1]$. Ces paramètres et cette variable vont permettre de modifier la sortie y qui est l'entrée pour notre observateur IMHSE.

Le modèle interne pour faire les estimations des états x_1 et x_2 sera le système discret suivant :

$$\begin{aligned}x_1(k+1) &= x_1(k) + \Delta t \cdot \delta_1 \cdot x_2(k) \\ x_2(k+1) &= x_2(k) - 0.2 \cdot \Delta t \cdot \delta_2 \cdot x_1(k) - \Delta t \cdot \delta_2 \cdot x_2(k) \cdot (x_1^2(k) - 1) \\ y(k) &= x_1(k)\end{aligned}\tag{6.18}$$

Où les paramètres δ_1 et δ_2 ont été mis en place pour tenir compte des variations dynamiques des paramètres du modèle (6.17), c'est à dire des dysfonctionnements du système qui devront être détectés par la méthode IMHSE multi-modèles.

6.5.2 Conditions générales du test

Les paramètres δ_1 et δ_2 du modèle (6.18) appartiennent à l'intervalle $\delta_i \in [0.5 \ 1.5]$ avec $i \in \{1,2\}$, échantillonnés avec $\Delta\delta_i = 0.1$. Nous appellerons famille de Van-Der-Pol cette famille de modèles de dysfonctionnements prévus. Ces modèles de dysfonctionnement sont montrés pour quelques paramètres dans la figure 6.51.

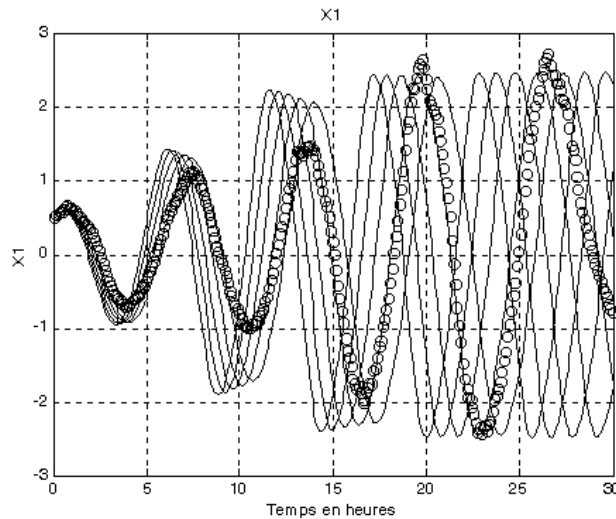


Figure 6.51 : Une famille de Van-Der-Pol prévue

Remarquons que sur la figure 6.51, la courbe formée par des "o" noirs, est la sortie du système dont l'état doit être estimé, mais dont le modèle exact n'est pas connu.

a) Considérons que le modèle du procédé (6.17) n'est pas perturbé, c'est à dire que les paramètres $\xi_1 = 0$ et $\xi_2 = 0$ sont fixes. Dans ces conditions, il y aura un seul modèle (6.18) possible pour le système, c'est à dire quand $\delta_1 = 1$ et $\delta_2 = 1$. Dans les figures 6.52 et 6.53 sont donnés les résultats d'estimation bornée de la méthode IMHSE multi-modèles pour 30 heures d'évolution, appliqué aux systèmes (6.17) et (6.18) sous les conditions décrites, avec une longueur d'horizon d'une demi-heure, $\Delta t = 6$ minutes, un domaine admissible défini par $[\Omega] = [-3,3] \times [-3,3]$ et une précision d'optimisation de $B_\varepsilon = 0.01$.

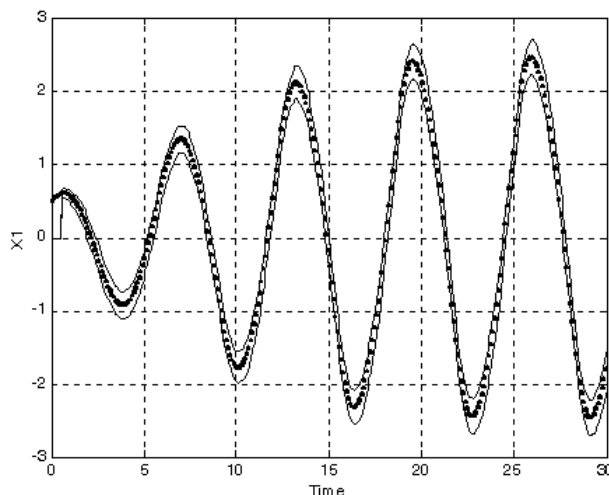


Figure 6.52 : Estimation bornée de l'état x_1

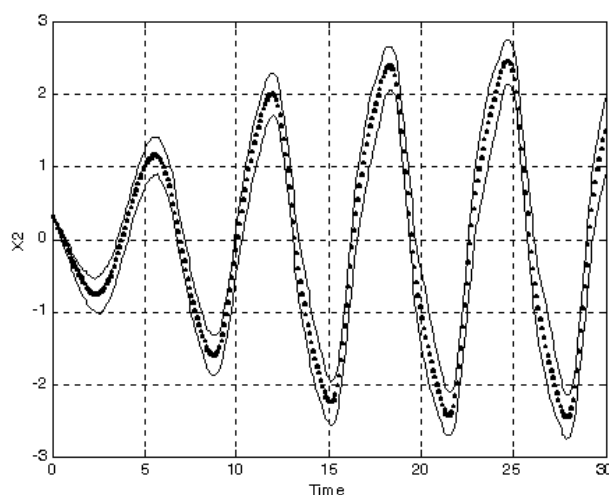


Figure 6.53 : Estimation bornée de l'état x_2

b) Considérons maintenant que le modèle du système (6.17) présente des variations dynamiques dans ces paramètres fixes $\xi_1 = 5\%$ et $\xi_2 = 10\%$ sur les états. Supposons que la famille de modèles de dysfonctionnements considérée est définie comme suit :

$$\{\delta_1, \delta_2\}_i = \begin{Bmatrix} 0.5 & 0.5 \\ 0.5 & 0.6 \\ \vdots & \vdots \\ 1.4 & 0.9 \\ \vdots & \vdots \\ 1.5 & 1.5 \end{Bmatrix}$$

Les figures 6.54 et 6.55 montrent les résultats d'estimation pour 30 heures d'expérience, appliqué aux systèmes (6.17) et (6.18) sous les mêmes conditions décrites précédemment.

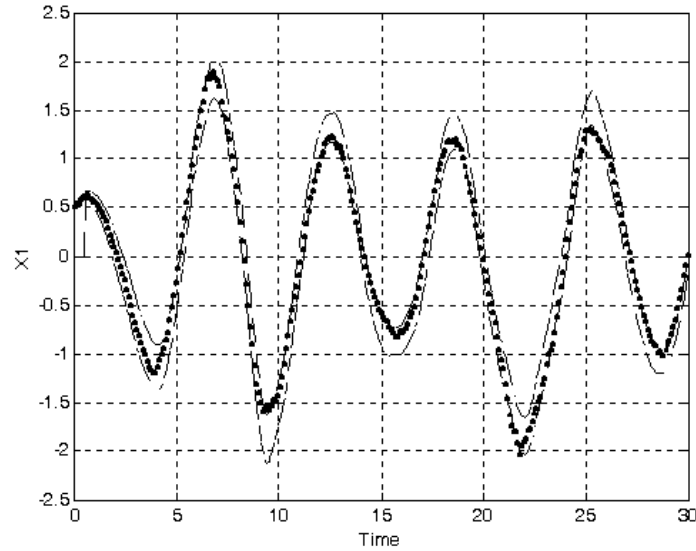


Figure 6.54 : Estimation bornée de l'état x_1

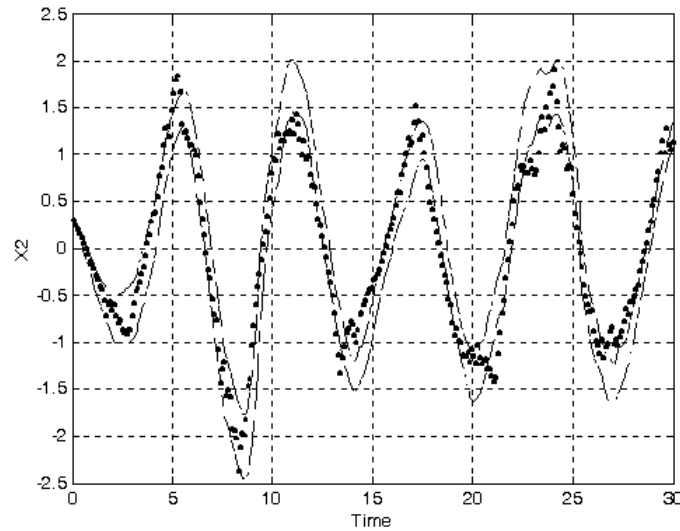


Figure 6.55 : Estimation bornée de l'état x_2

c) Appliquons la méthode IMHSE multi-modèle au même système, mais cette fois-ci les variations dynamiques de ces paramètres sont de $\xi_1 = 10\%$ et $\xi_2 = 20\%$. Les résultats sous les mêmes conditions décrites précédemment, sont montrés par les figures 6.56 et 6.57.

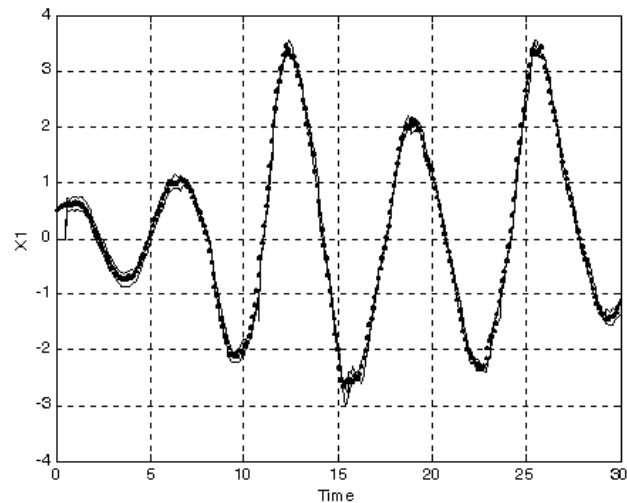


Figure 6.56 : Estimation bornée de l'état x_1

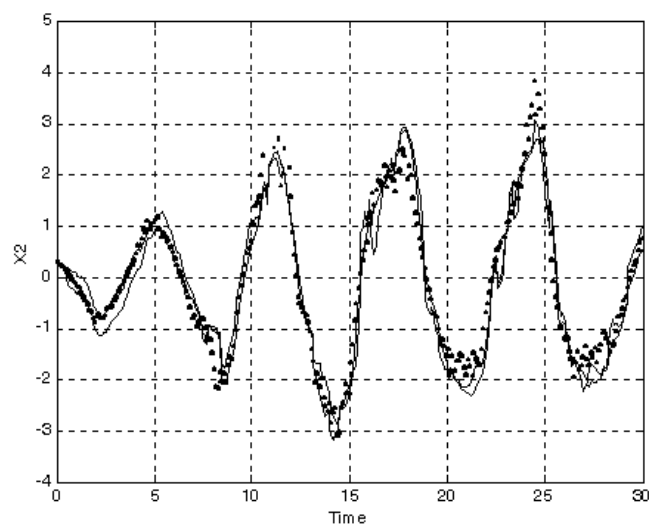


Figure 6.57 : Estimation bornée de l'état x_2

6.5.2.1 Analyse des résultats et conclusion cas 5

Nous avons présenté l'application de la méthode ensembliste IMHSE pour détecter les dysfonctionnements non prévus par le modèle nominal. Cet observateur utilise de multiples modèles. Cette approche utilise les avantages de la méthode IMHSE pour faire des estimations dans des systèmes non-linéaires. Elle utilise aussi les concepts liés à la prédiction d'un espace de recherche pour détecter les dysfonctionnements non prévus par le modèle interne de l'observateur. Cet observateur s'appuie sur une famille de modèles de dysfonctionnements. Les résultats d'estimation sur un modèle non-linéaire de Van-Der-Pol montrent dans les cas a), b) et c) les performances de la technique employée quand le modèle exact n'est pas connu.

6.6 Cas 6 : Application à un modèle de fermentation

Bioréacteur à échelle pilote

Maintenant, nous allons présenter un système non-linéaire issu d'un bioprocédé réel. Le but est l'estimation de l'humidité d'un réacteur. Pour vérifier les performances de l'approche, nous disposons d'un modèle dynamique et d'une base de données expérimentales [Peña y Lillo, 1999].

Nous cherchons à estimer l'état du système par notre approche IMHSE multi-modèles présenté au chapitre 5 [Valdés-González *et al*, 2002c].

Pour appliquer notre méthode, le modèle existant [Peña y Lillo *et al*, 2001] (qui n'est pas un bon modèle) sera modifié. Ceci sera caractérisé par deux paramètres variants dans le temps, très difficiles à identifier, mais nécessaires pour le contrôle du procédé.

6.6.1 Principe de fonctionnement

Cette étude présente un bio-réacteur de type SSC (Solid Substrate Cultivation) à agitation périodique de 200-kg de capacité.

Un modèle simplifié qui décrit l'évolution des variables principales est rapporté en [Peña y Lillo *et al*, 2001]. Ce modèle phénoménologique, qui est basé sur le bilan de masse et d'énergie, utilise deux variables d'état : La matière sèche (M_s) et la quantité d'eau (X_w). Le modèle en temps discret est montré dans les équations suivantes :

$$Ms_{k+1} = Ms_k - \Delta t_1 \cdot k_g \cdot CPR_k \quad (6.19)$$

$$Xw_{k+1} = Xw_k + \Delta t_1 \cdot \left(R_w + \frac{F_w}{Ms_k} - \frac{W}{Ms_k} - \frac{W_p}{Ms_k} - \frac{Xw_k}{Ms_k} \cdot Ms_{k+1} \right) \quad (6.20)$$

Où Δt_1 est la période d'échantillonnage, CPR représente le taux de production de CO_2 . R_w est la production de l'eau. Elle est considérée comme proportionnelle à CPR.

$$R_w = \frac{k_{w1} \cdot CPR_k}{Ms_k} \quad (6.21)$$

W est le taux de l'évaporation. De plus, de l'eau fraîche est périodiquement ajoutée à travers le sommet du réacteur (F_w), et un peu d'eau tombe goutte à goutte du fond (W_p). Ces variables (CPR , F_w et W_p) sont des données expérimentales, disponibles tout au long de l'expérience. Les valeurs des paramètres fixes du modèle sont présentées dans le tableau 6.

Tableau 6 : Paramètres fixes

Paramètres	Valeur calibrée
k_g	0.73
kw_1	0.41
W_p	0.50

- k_g : Facteur de conversion entre la consommation de la masse sèche et la production de CO_2 .
- kw_1 : facteur de relation entre le taux de production du CO_2 et la production d'eau.

6.6.2 Conditions générales du test

Pour la méthode IMHSE multi-modèles, des mesures hors ligne de la quantité d'eau sont disponibles tout au long de l'expérience, faites au niveau du laboratoire à intervalles réguliers de $\Delta t_2 = 3$ heures.

La sortie du modèle du bioréacteur est obtenue en utilisant des mesures indirectes de CO_2 comme suit, affectées par un bruit additif de 2% :

$$y_k = Ms_k + \omega_k \quad (6.22)$$

L'état initial du modèle est connu : $Ms_0 = 100.45$ kg et $Xw_0 = 1.1739$ kg_{H_2O} / kg_{Ms} .

L'objectif de cette application est l'estimation en ligne de la quantité d'eau X_w dans le bioréacteur. En effet, jusqu'à maintenant les mesures de cette variable ont été faites hors ligne. Plusieurs fois, d'importants investissements ont été perdus quand les mesures n'ont pas été faites à temps pour que la biomasse ne périsse pas. Le modèle

(6.19-6.20) qui est rapporté en [Peña y Lillo *et al*, 2001] n'est pas un très bon modèle du procédé.

Vu les valeurs expérimentales disponibles, nous avons déterminé l'existence de deux paramètres δ_1 et δ_2 qui changent avec chaque nouvelle expérience et chaque nouvelle condition initiale. L'équation (6.20) peut être donc réécrite comme suit :

$$Xw_{k+1} = Xw_k + \Delta t_1 \cdot \left(R_w + \frac{\delta_1 \cdot F_w - W - W_p - Xw_k \cdot Ms_{k+1}}{Ms_k} - \delta_2 \cdot Xw_k \right) \quad (6.23)$$

Il est à noter que le paramètre δ_1 renforce l'influence de l'eau fraîche ajoutée, alors que δ_2 modélise la descente du premier ordre nécessaire pour savoir quand ajouter de l'eau au réacteur. Dans une future étape de contrôle, ceci pourra être fait par exemple en utilisant un seuil inférieur pour X_w .

Considérons une famille de modèles prévus générés par δ_1 et δ_2 . Ces derniers seront échantillonnés dans les intervalles suivant $\delta_1 = [0.6, 2.8]$ et $\delta_2 = [0, 0.02]$. Pour notre application cet échantillonnage est montré dans le tableau 7.

Tableau 7 : Famille de modèles de dysfonctionnements prévus

η_i	1	2	...	7	8	...	51	52	53	54	55
δ_1	0.6	0.6	...	0.8	0.8	...	2.8	2.8	2.8	2.8	2.8
δ_2	0.0	0.04	...	0.0	0.004	...	0.004	0.008	0.012	0.016	0.020

L'expérience a une durée de 100 heures, les mesures en ligne du taux de CO_2 et hors ligne de (Xw) sur le système sont disponibles respectivement à des intervalles Δt_1 et Δt_2 . La longueur d'horizon est définie à $lh = 4$ heures.

La précision d'optimisation utilisée est $B_\varepsilon = 0.01$ et le domaine admissible pour le problème d'estimation est $[\Omega] = [X_w] \times [M_s]$ tel que $[\Omega] = [0, 4] \times [0, 120]$.

6.6.3 Résultats

La figures 6.58 et 6.59 montrent les résultats de l'estimation par intervalles, des deux états du système M_s et X_w respectivement. Notons que la ligne continue représente le modèle incomplet rapporté en [Peña y Lillo *et al*, 2001] et qui correspond aux équations (6.19)-(6.22). Les estimations par intervalles correspondent

au modèle (6.23), dont les paramètres δ_1 et δ_2 ont été inclus avec un critère d'optimisation du type (2.16) qui considère les mesures hors ligne de X_w .

Dans la figure 6.58, nous pouvons remarquer aussi que les seules mesures expérimentales possibles sont faites sur la matière sèche ; Ces mesures correspondent au début et à la fin de l'expérience. Dans le cas de la figure 6.59, les mesures hors ligne de X_w sont aussi identifiées avec le symbole " \diamond ".

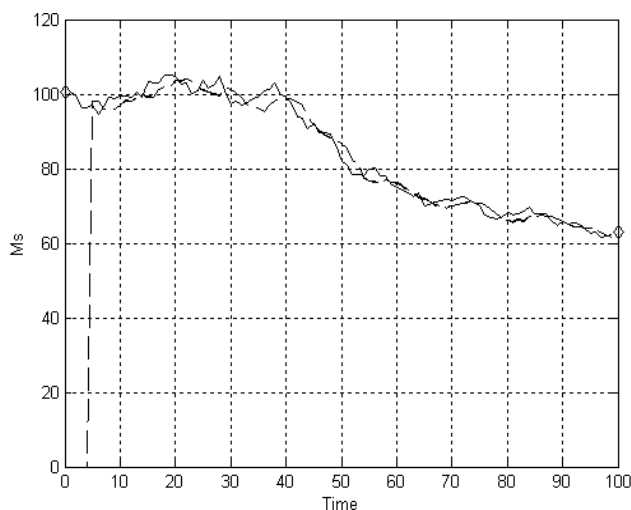


Figure 6.58 : La matière sèche (M_s) simulée et estimée

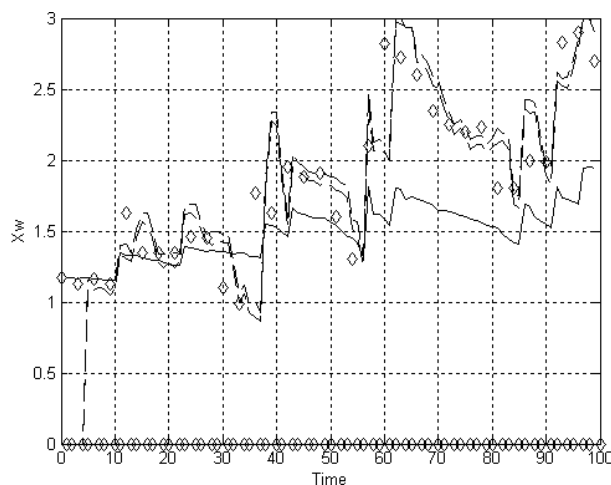


Figure 6.59 : Estimations de la quantité d'eau (X_w) simulée et estimée

Les différents modèles utilisés pendant l'estimation (modèles commutés) sont illustrés dans la figure 6.60. L'évolution dans le temps des paramètres δ_1 et δ_2 est donnée dans les figures 6.61 et 6.62 respectivement. Les données expérimentales de CPR , F_w et W sont données dans les figures 6.63, 6.64 et 6.65 respectivement.

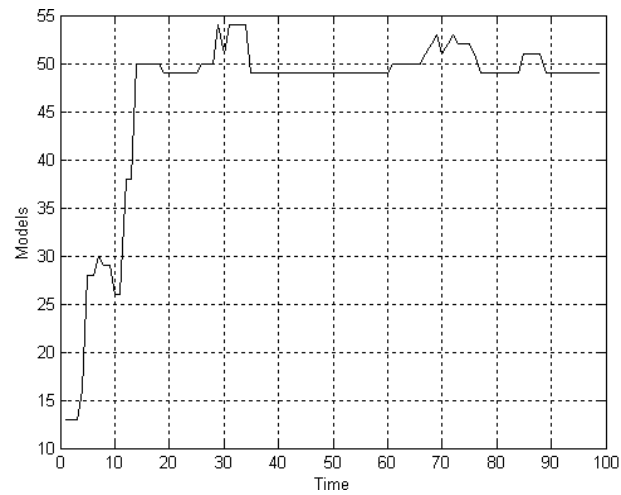


Figure 6.60 : Modèles de dysfonctionnement

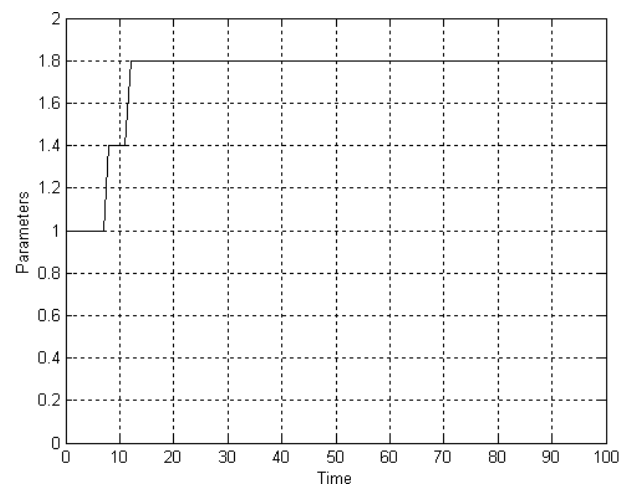


Figure 6.61 : Variations du paramètre δ_1 .

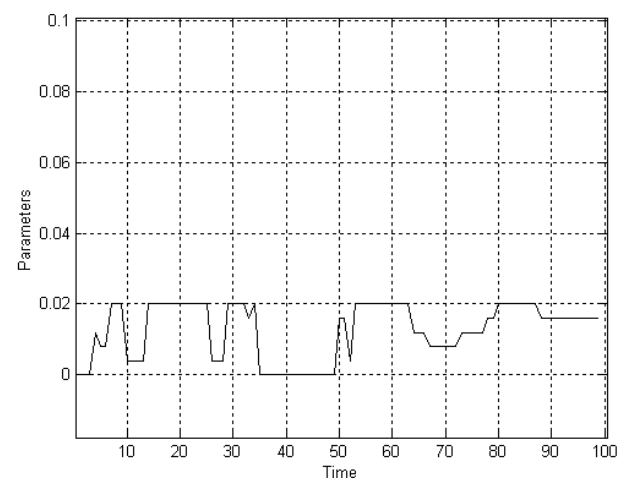


Figure 6.62 : Variations du paramètre δ_2 .

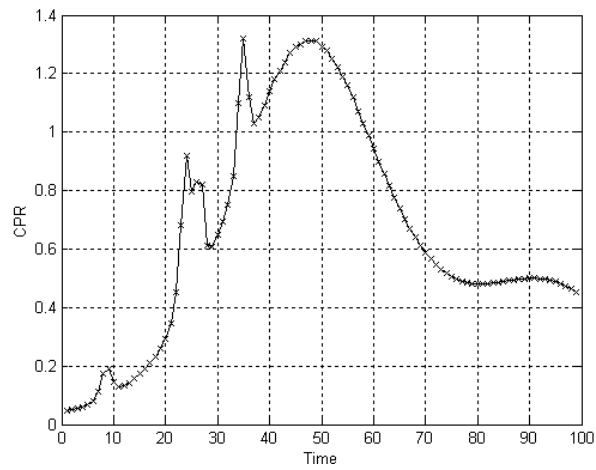


Figure 6.63 : Taux expérimental de CO_2

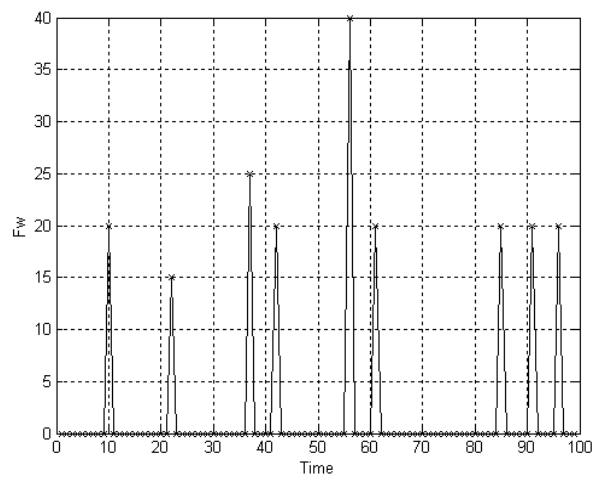


Figure 6.64 : Apports d'eau fraîche (F_w)

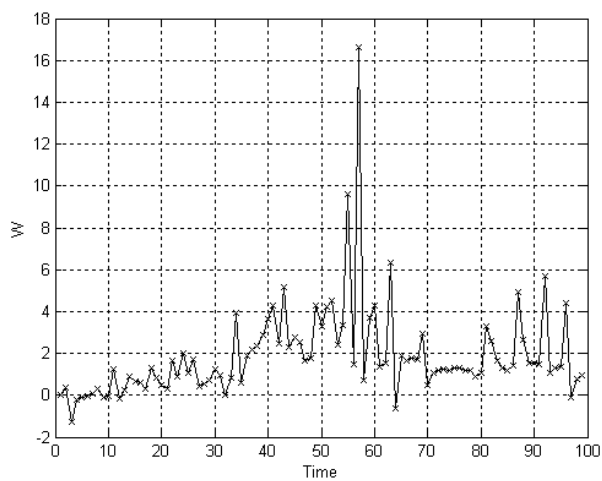


Figure 6.65 : Evaporation pendant la fermentation (W).

6.6.3.1 *Analyse de résultats et conclusion cas 6*

La détection des dysfonctionnements du système dans le cadre d'un observateur non-linéaire ensembliste globalement convergent a été présentée. Cette application a utilisé un modèle et des données expérimentales d'un bioprocédé issu d'un système réel. Cette approche utilise les avantages de la méthode IMHSE pour détecter des dysfonctionnements non prévus par le modèle interne de l'observateur.

Quand le dysfonctionnement est détecté, la méthode utilise une famille de modèles de dysfonctionnements prévus. L'algorithme présenté au chapitre 5, permet de choisir parmi les différents modèles "disponibles", celui qui donne une meilleure approximation au comportement réel du système. C'est à dire le meilleur modèle au sens de plus proche de celui utilisé dans l'estimation antérieure.

Il faut noter qu'avec la méthode proposée, dont nous disposons d'une famille de modèles de dysfonctionnements prévus, l'estimation d'état et de paramètres converge. C'est à dire, nous trouverons toujours un jeu de paramètres valides ; l'estimation paramétrique ne pourra pas diverger.

De plus, avec cette méthode, nous exploitons les différents types mesures qui peuvent être disponibles, que ce soit les mesures en ligne ou hors ligne.

Les résultats obtenus montrent une bonne performance de la technique appliquée, sur un mauvais modèle d'un procédé biotechnologique rapporté dans la littérature. Ce modèle a été modifié pour appliquer notre approche.

La méthode ensembliste utilisée permet de se rapprocher du modèle le plus représentatif du comportement réel du système au fur et à mesure du temps. Ceci est fait en commutant le modèle interne parmi les modèles disponibles.

Chapitre 7

Conclusions générales et perspectives

Dans ce travail, nous avons présenté une nouvelle approche pour l'estimation d'état dans des systèmes non-linéaires complexes. Cette approche a été mise en œuvre dans le cadre d'un algorithme numérique d'estimation d'état appelé IMHSE. Elle a été implémentée avec le langage MATLAB, couplé à un outil développé en JAVA pour les manipulations nécessaires sur les ensembles.

Nous nous étions fixés comme objectif principal de développer une méthode d'estimation d'état qui soit globalement convergente et bien adaptée au contexte industriel. Dans ce cadre, l'algorithme d'estimation proposé utilise un algorithme d'estimation non-linéaire à horizon fuyant, associé à une méthode globale d'optimisation par intervalles. Cette méthode d'estimation possède les caractéristiques suivantes :

- Globalement convergente
- Performante pour une large classe de systèmes
- Implémentation aisée
- Utilisation et obtention d'ensembles garantis
- Capacité de détection de non-observabilités locales d'un système, sur un domaine admissible
- Capacité de détection de dysfonctionnements non prévus par le modèle interne de l'observateur
- Capacité de réaction aux dysfonctionnements non prévus par le modèle, en utilisant de multiples modèles décrivant le comportement observé du système

L'algorithme d'estimation IMHSE est un algorithme efficace permettant de faire des estimations d'états pour un système quelconque en utilisant une représentation ensembliste. Ainsi, il a été utilisé ici pour faire des estimations sur des systèmes biotechnologiques réels.

Cet algorithme d'estimation est basé sur l'interaction de différents algorithmes ensemblistes. Ces algorithmes sont capables de manipuler des ensembles pour résoudre des tâches spécifiques, comme par exemple l'algorithme d'optimisation globale, de manipulation de modèles par intervalles, d'obtention de trajectoires ensemblistes, etc. Les résultats sont ensuite analysés et traités afin de générer l'estimation de l'état en fin d'horizon.

Notre principale contribution a été de formaliser un algorithme d'estimation non-linéaire par intervalles, applicable à une grande variété de systèmes. Notre algorithme bénéficie des avantages de la méthode à horizon fuyant, qui est indépendante des structures du modèle observé. Nos principales attentes étaient :

- a) Du point de vue de l'observateur ensembliste
 - _ Mise au point du couplage d'un observateur robuste de type MHSE avec une technique globale d'optimisation non-linéaire par intervalles.
 - _ Capacité de recalage en ligne à partir de mesures hors-ligne.
 - _ Fournir des indications en ligne à propos de l'observabilité du système, de façon à utiliser les estimations avec prudence.
- b) Du point de vue du modèle
 - _ Que la méthode IMHSE soit capable de travailler avec des modèles issus de représentations différentes.
 - _ Pouvoir utiliser (si nécessaire) plusieurs modèles d'un même procédé.
- c) Du point de vue du capteur logiciel
 - _ Simple à mettre en oeuvre, tout en nécessitant un minimum de réglages.
- d) Du point de vue de l'expérimentation
 - _ Nous cherchons la performance de l'estimation dans des procédés réels biotechnologiques.

En ce qui concerne les contraintes décrites en a), nous avons mis au point une méthode d'estimation d'état non-linéaire à horizon fuyant basée sur une méthode d'optimisation globale par intervalles, et nous avons vu qu'elle a des performances très intéressantes dans un bon nombre de cas. L'application de cet algorithme à des

procédés réels a montré ses performances. En particulier, citons l'application à un bioprocédé batch générique, dans lequel le filtre de Kalman étendu et une version locale de la méthode appelée MHSE-BFGS divergent. Cette divergence vient du fait que le système n'était pas observable, car le taux spécifique de croissance a été modélisé intentionnellement par une loi hybride afin de rendre le système non-observable. Evoquons aussi la même application avec des perturbations sur l'état (très réalistes pour des bioprocédés), dans laquelle la méthode IMHSE est clairement supérieure au filtre de Kalman et à la version locale de MHSE-BFGS. En effet, la méthode IMHSE n'a pas besoin de réglages, et converge globalement sur tout le domaine admissible du problème d'estimation.

Les différents types d'information que notre méthode d'estimation peut traiter font de cette approche un outil avantageux pour l'estimation d'état. L'utilisation de mesures hors ligne pour recalibrer l'observateur permettent une approche d'estimation qui tienne compte des trajectoires réelles du système, même si ces mesures ont été obtenues avec du retard. En ce qui concerne le bioréacteur à échelle pilote, la prise en compte des mesures hors ligne de la quantité d'eau, a permis d'améliorer remarquablement les estimations faites par rapport au modèle spécialement développé pour le procédé, même si celui-ci n'est pas un très bon modèle.

La considération de mesures hors ligne fait aussi partie de nos contributions originales. Elle repose sur une modification du critère du problème global d'optimisation, dans le problème d'estimation optimal IMHSE. Cette méthode peut maintenant accueillir toutes les mesures hors ligne, partielles ou totales des variables d'état. Il est important de remarquer, que l'utilisation de ce type d'information diminue les problèmes d'observabilité locale, parce qu'elles donnent des informations nouvelles sur les vraies positions des trajectoires du problème.

D'un autre côté, il est bien connu que les bioprocédés sont en général problématiques, car ils n'ont pas de point de fonctionnement et présentent fréquemment des problèmes d'observabilité locale. La méthode présentée nous permet cependant de détecter ces problèmes d'observabilité locale. Ainsi elle ne diverge pas en faisant des estimations erronées, comme cela arrive avec le filtre de Kalman ou d'autres techniques locales d'estimation d'état. Nous avons proposé des indicateurs d'observabilité pour faire des estimations pas à pas, et pour tenir compte des problèmes d'observabilité locale. Ainsi dans le cas d'un instant de non-observabilité ponctuelle ou locale, nous pourrions le détecter, par exemple, depuis un programme superviseur, en fixant par défaut la valeur de l'estimation actuelle à celle de

l'estimation antérieure actualisée. Ceci rend notre observateur très performant et lui donne une grande fiabilité, en nous permettant d'utiliser les estimations avec toute la prudence nécessaire. Cette caractéristique a une influence non négligeable. En effet, rappelons qu'il est difficile de vérifier que, pour toute entrée, le système sera partout localement observable.

Les contraintes présentées en b), imposent à notre observateur de pouvoir être utilisé dans une large classe de systèmes. A cet effet, nous avons montré que notre approche permet d'utiliser même des modèles hybrides. Dans ce cas, les résultats ont été tout à fait adéquats car pour fonctionner, la méthode a besoin seulement d'un modèle dynamique interne sous forme de représentation d'état et d'un modèle pour la sortie. Bien que le modèle ne soit pas toujours bien identifié, notre observateur admet facilement une représentation interne du procédé comme dans le cas de la quatrième application. Celle-ci a été réalisée en simulation sur le modèle d'un système dynamique de fermentation sur substrat solide. Dans ce système, le modèle de l'azote intermédiaire est difficilement identifiable. Il a donc été conçu de façon hybride. Dans le cas de la deuxième application, la dépendance du substrat de la taux spécifique de croissance a été modélisée de la même façon.

Nous savons que certains paramètres d'un modèle dynamique sont difficiles à identifier. Dans ce contexte, nous avons contribué à résoudre ce problème, par l'observation d'états multi-modèles. Ainsi, notre approche ensembliste permet de détecter facilement quand un modèle (voire certains paramètres) est incompatible avec l'évolution observée des variables d'état sur un horizon de temps donné. C'est un dysfonctionnement de modèle interne, qui est détecté lorsque aucun minimum global n'existe dans une prédiction de l'espace initial de recherche.

L'algorithme d'estimation avec de multiples modèles proposé possède un ensemble de modules simples pour traiter efficacement la détection des dysfonctionnements non prévus par le modèle interne de l'observateur. L'approche IMHSE multi-modèles qui a été proposée utilise des mesures indirectes en ligne, des mesures hors ligne peu fréquentes, une prédiction pour l'espace de recherche, ainsi qu'une famille de modèles possibles pour le procédé. Cette famille de modèles est appelée modèle de dysfonctionnements prévus.

En ce qui concerne la question de l'observabilité locale d'un système, elle peut être vue comme un dysfonctionnement du modèle interne, et donc traitée par commutation de modèles.

La facilité de la mise en œuvre de notre méthode d'estimation, est la contrainte donnée en c) du point de vue capteur logiciel. A ce propos, notre méthode a montré sa facilité d'implantation dans différents types de systèmes. En effet, la méthode d'estimation à horizon fuyant possède un seul paramètre à régler, la longueur de l'horizon.

Enfin, nous avons illustré notre méthode IMHSE pour différentes applications. Ces exemples ont permis de valider les capacités de notre approche en simulation, et de montrer les performances de cet outil d'estimation dans différents cas théoriques. Nous avons également validé l'outil (en simulation) sur un bioprocédé industriel réel de fermentation sur substrat solide. Là encore, la méthode IMHSE a tenu ses promesses. Ce qui vérifie la contrainte formulée en d).

Il faut souligner que pour ce type d'approche, plus la taille du vecteur d'état est grande, plus elle devrait être utilisée pour des systèmes qui évoluent lentement dans le temps. Dans le cas des bioprocédés, la taille du vecteur d'état est généralement limitée.

Comme résumé des contributions originales de notre travail, nous pouvons mentionner :

- Le couplage d'une méthode globale d'optimisation ensembliste pour rendre la méthode d'estimation non-linéaire d'état à horizon glissant globalement convergente.
- L'utilisation d'une prédiction de l'espace de recherche, comme domaine admissible pour la prochaine estimation. Cela réduit la complexité informatique impliquée et le temps de calcul, et autorise une réduction de la longueur de l'horizon jusqu'à la dimension de l'état.
- La modification du critère classique d'optimisation, lié au problème optimal d'estimation à horizon fuyant. Cette modification permet d'utiliser les mesures partielles de l'état obtenues hors lignes, pour recalculer le critère.
- L'estimation ensembliste sur des systèmes hybrides.
- Le couplage MATLAB-JAVA pour la manipulation des intervalles sur un horizon de temps, lié à un système dynamique.
- Les contributions relatives à l'observabilité numérique et la distinguabilité des états par intervalles.
- La détection de dysfonctionnements non prévus par le modèle, et la mise en place d'un algorithme d'estimation d'état capable d'utiliser une famille de

modèles de dysfonctionnements prévus, pour faire des estimations par intervalles.

7.1 Perspectives

Dans le cadre d'applications industrielles, une des améliorations possibles est l'implémentation de l'algorithme IMHSE avec l'ajout d'interfaces graphiques et de fonctions supplémentaires telles que des outils de modélisation de systèmes hybrides. Cela permettrait en effet d'augmenter les capacités de modélisation de l'outil et simplifierait encore la tâche d'estimation.

Au vu de l'importance du procédé de GA_3 au Chili, une autre perspective envisageable est l'implémentation expérimentale de la méthode présentée, en fermant la boucle de la commande pour la production d'acide gibbéréllique.

Dans le cadre de recherche théorique, une perspective très intéressante concerne l'étude et la mise en place d'un algorithme d'estimation, dont les modèles puissent changer pendant l'horizon de temps choisi, en fonction du type de système considéré.

Une autre perspective consiste à rendre adaptative la méthode d'optimisation globale. Par exemple couper les boîtes (états) par le côté le moins discernable, ce qui permettra de traiter les problèmes d'estimation plus rapidement.

Le développement d'une stratégie hybride d'estimation (déterministe et probabiliste) présente aussi des perspectives intéressantes notamment grâce aux propriétés de l'opérateur de Newton par intervalles. Si nous pouvons montrer par intervalles que la solution existe en début d'horizon et qu'elle est unique, nous pourrons alors utiliser une technique d'optimisation locale classique pour attendre l'optimum ou une technique heuristique d'optimisation, particulièrement lorsque la dimension de l'état est grande.

Finalement un champ de recherche reste ouvert dans l'utilisation de modèles hybrides ensemblistes pour faire des estimations. En effet, à notre connaissance, il manque des algorithmes et méthodologies à l'heure actuelle pour traiter ce problème.

Bibliographie

- [Adrot, 2000] O. Adrot, *Diagnostic à base de modèles incertains utilisant l'analyse par intervalles : l'approche bornante*, Thèse de Doctorat, Institut National Polytechnique de Lorraine, Nancy, 2000.
- [Aeyels, 1981] D. Aeyels. *On the number of samples necessary to achieve observability*. Systems Control Lett., Vol. 1, pp. 595-603, 1981.
- [Alamir, 1999] M. Alamir, *Optimization based observer revisited*, International Journal of Control, Vol. 72, No. 13, pp.1204-1217, 1999.
- [Alamir, 2001] M. Alamir, *Estimateurs d'état à horizon glissant*, Dans : J.P. Corriou (Sous la direction de), *Automatique et procédés chimiques*, Traité IC2 Information - Commande - Communications, série Systèmes Automatisés, Hermès Science Publications, pp.147-179, 2001.
- [Alvarez, 2000] J. Alvarez, *Non-linear state estimation with robust convergence*, Journal of Process Control, Vol. 10, pp. 59-71, 2000.
- [Allgöwer *et al.*, 1999] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings et S. J. Wright. *Nonlinear Predictive Control and Moving Horizon Estimation – An Introductory Overview*, In *Advances in Control Highlights of ECC'99*, Paul M. Frank (Ed), Springer-Verlag, Chapter 12, pp. 391-449, 1999.
- [Ammar, 1995] H. Ammar, *La Methode d'Optimisation Globale Multidimensionnelle ALIENOR - Solution analytique d'un Modèle de Diffusion de la Streptavidine et de la DTPA-Biotine en Immunoscintigraphie Cardiaque en Deux Etapes*, Thèse de Doctorat, Laboratoire MEDIMAT, Spécialité: Mathématiques Appliquées, Université Pierre et Marie Curie, Paris VI, Paris, 1995.

- [Atkinson, 1989] K. E. Atkinson, *An introduction to Numerical Analysis*, John Wiley & Sons, New York, 1989.
- [Bastin et Dochain, 1990] G. Bastin et D. Dochain, *On-Line Estimation and Adaptive Control of Bioreactors*, Elsevier, Amsterdam, 1990.
- [Bendiab, 1995] O. Bendiab, *Deux Nouvelles Méthodes d'Optimisation Globale : Applications à des Problèmes de Biologie*, Thèse de Doctorat, Laboratoire MEDIMAT, Spécialité: Mathématiques Appliquées, Université Pierre et Marie Curie, Paris VI, Paris, 1995.
- [Besançon, 1996] G. Besançon, *Contributions à l'Etude et à l'Observation des Systèmes Non Linéaires avec Recours au Calcul Formel*, Thèse de Doctorat, Laboratoire d'Automatique de Grenoble, INPG, Grenoble, 1996.
- [Boillereaux, 1996] L. Boillereaux, *Estimation d'Etat de Procédés Non-Linéaires : Méthode à Horizon Glissant avec Indicateur de Qualité*, Thèse de Doctorat, Laboratoire d'Automatique de Grenoble, INPG, Grenoble, 1996.
- [Boillereaux et Flaus, 2000] L. Boillereaux, J. M. Flaus, *A new approach for designing model based indirect sensors*, IEEE Control System Application, Vol. 8, No. 4, pp. 601-608, 2000.
- [Brookner, 1998] E. Brookner. *Tracking and Kalman Filtering Made Easy*, John Wiley & Sons, 1998.
- [Brown et Hwang, 1997] R. Brown et P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering : With Matlab Exercises and Solutions*, John Wiley&Sons, Third Edition, 1997.
- [Cazzador et Lubenova, 1995] L. Cazzador et V. Lubenova, *Nonlinear estimation of specific growth rate for aerobic fermentation processes*, Biotechnology and Bioengineering, Vol. 47, No. 6, pp.626-632, 1995.
- [Chamilothis, 1987] G. Chamilothis, *Techniques Adaptatives pour le suivi et la Conduite des Processus de Fermentation*, Thèse de Doctorat, Université Paul Sabatier de Toulouse (Sciences), Toulouse, 1987.
- [Charbonnier, 1994] S. Charbonnier, *Contribution de l'Automatique au Développement d'un Procédé de Production d'Enzymes: Modélisation, estimation et Commande*, Thèse de Doctorat, Laboratoire d'Automatique de Grenoble, INPG, Grenoble, 1994.

- [Chen *et al*, 1997] Guanrong Chen, Jianrong Wang, Leang Shieh, *Interval Kalman Filtering*, IEEE Transaction on Aerospace and Electronic Systems, Vol. 33, No. 1, pp. 250-259, January 1997.
- [Cherruault, 1998] Y. Cherruault, *Modèles et méthodes mathématiques pour les Sciences du Vivant*, Presses Universitaires de France (Puf), France, 1998.
- [Cherruault, 1999] Y. Cherruault, *Optimisation : Méthodes locales et globales*, Presses Universitaires de France (Puf), Collections mathématiques, France, 1999.
- [Dekker et Aarts, 1991] A. Dekker et Emile Aarts. *Global Optimization and simulated annealing*, Mathematical Programming, Vol. 50, pp.367, 1991.
- [Díaz *et al*, 1996] A. Díaz, F. Glover, H. M. Ghaziri, J. L. González, M. Laguna, P. Moscato et F. T. Tseng, *Optimización Heurística y Redes Neuronales en Dirección de Operaciones e Ingeniería - (Tutorial de Fred Glover)*, Editorial Paraninfo, Madrid, España, 1996.
- [Dimitratos *et al*, 1991] J. Dimitratos, C. Georgakis, M. El-aasser et A. Klein, *An Experimental Study of Adaptive Kalman Filtering in Emulsion Copolymerization*, Chemical Engineering Science, Vol. 46, No. 12, pp. 3203-3218, 1991.
- [Dochain, 1986] D. Dochain, *On line parameters estimation, adaptive state estimation and adaptive control of fermentation processes*, Thèse de Doctorat, Université Catholique de Louvain, Louvain, Belgique, 1986.
- [Eglese, 1990] R. W. Eglese, *Simulated Annealing: A tool for Operational Research*, European Journal of Operation Research, Vol. 46, pp.271-281, 1990.
- [Farza *et al*, 1998] M. Farza, K. Busawon et H. Hammouri, *Simple Nonlinear Observers for On-line Estimation of Kinetic Rates in Bioreactors*, Automatica Vol. 34, No. 3, pp.301-318, 1998.
- [Ferrari-Trecate *et al.*, 2000] G. Ferrari-Trecate , D. Mignone et M. Morari, *Moving Horizon Estimation for Hybrid Systems*, Proceedings of the 2000 American Control Conference (ACC), Chicago, USA, Vol. 3, pp. 1684-1688 June 2000.
- [Flaus et Boillereaux, 1995] J.M. Flaus et L. Boillereaux, *Estimation d'état pour Modèles dynamiques à Réseaux de Neurones*, Réc. Prog. Gén. Proc. Lavoisier, Vol. 37, pp: 73-86, 1995.

- [Flaus, 1998] J.M. Flaus, *Contribution à la commande de procédés - Estimation d'état par méthodes numériques - Modélisation et commande de systèmes dynamiques hybrides*, Habilitation à Diriger des Recherches, LAG-INPG, France, 1998.
- [Gatzke et Doyle III, 2002] E. P. Gatzke et F. J. Doyle III, *Use of multiple models and qualitative knowledge for on line moving horizon disturbance estimation and fault diagnosis*, Journal of Process Control, Vol. 12, pp. 339-352, 2002.
- [Gauthier et Bornard, 1981] J.P. Gauthier et G. Bornard, *Observability for any $u(t)$ of a Class of Nonlinear Systems*, IEEE Trans. On Automatic Control, vol. 26, pp.922-926, 1981.
- [Gauthier *et al.*, 1992] J. P. Gauthier, H. Hammouri et S. Othmann, *A Simple Observer for Nonlinear Systems: Application to Bioreactors*, IEEE Trans. On Automatic Control, vol. 37, No 6, pp.875-880, June 1992.
- [Gelmi, 1999] C. Gelmi, *Análisis, Modelación y Simulación del Crecimiento de Gibberella Fujikuroi y Producción de Acido Giberélico sobre Sustrato Sólido Inerte*, M.Sc. Thesis, Universidad Catolica de Chile, 1999.
- [Grewal *et Andrews*, 1993] M. Grewal et A. Andrews, *Kalman Filtering : Theory and Practice*, Prentice Hall, 1993.
- [Hadj-Sadok et Gouzé, 1999] M. Z. Hadj-Sadok et J. L. Gouzé, *Comparison between two interval observers for wastewater treatment models*, European Control Conference, ECC'99, Karlsruhe, Germany, CD-ROM, 1999.
- [Hadj-Sadok et Gouzé, 2001] M. Z. Hadj-Sadok et J. L. Gouzé, *Estimation of uncertain models of activated sludge processes with interval observers*, Journal of Process Control, Vol. 11, pp. 299-310, 2001.
- [Hammouri et Othman, 2001] H. Hammouri et S. Othman, *Observateurs de systèmes non linéaires*, Dans : J.P. Corriou (Sous la direction de), *Automatique et procédés chimiques*, Traité IC2 Information - Commande - Communications, série Systèmes Automatisés, Hermès Science Publications, pp.105-146, 2001.
- [Hansen, 1992] E. R. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker, New York, 1992.
- [Heine, 1994] G. W. Heine, *Smart Simulated Annealing*, PhD Thesis, University of Colorado at Denver, 1994.

- [Hermann et Krener, 1977] R. Hermann et A.J. Krener, *Nonlinear Controllability and Observability*, IEEE Trans. On Automatic Control, vol. 22, pp.728-740, 1977.
- [Ichida et Fujii, 1979] K. Ichida et Y. Fujii, *An interval arithmetic method for global optimization*, Computing, Vol. 23, pp. 85-97, 1979.
- [Ingber, 1989] A. L. Ingber, *Very fast simulated re-annealing*, Journal Mathematical Computer Modeling, No.12, Vol. 8, pp.967-973, 1989.
- [Jaulin et al, 1993] L. Jaulin et E. Walter, *Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis*, Journal Mathematics and Computers in Simulation, No.35, pp.123-137, 1993.
- [Jaulin, 2000] L. Jaulin, *Le Calcul Ensembliste par Analyse par Intervalles et ses Applications*, Habilitation à Diriger des Recherches (<http://www.istia.univ-angers.fr/~jaulin/>), Université d'Angers, Paris, 2000.
- [Jaulin et al, 2001] L. Jaulin, M. Kieffer, O. Didrit et E. Walter, *Applied Interval Analysis*, Springer, 2001.
- [Johnson, 1988] L. E. Johnson, *Simulated Annealing*, American Journal of Mathematical and Management Sciences, Vol. 4, No. 3-4, pp. 205-206, 1988.
- [Kalman et Bucy, 1961] R. E. Kalman et R. S. Bucy, *New results in linear filtering and prediction theory*, Trans. ASME, Vol. 83(D), pp. 95-108, 1961.
- [Kearfott, 1998] R. B. Kearfott, *Interval Newton methods*, (disponible en <http://interval.louisiana.edu/GLOBSOL/whatisop/node9.html>), 1998
- [Kieffer et al, 1998] Michel Kieffer, Luc Jaulin et Eric Walter, *Guaranteed Recursive Nonlinear State Estimation Using Interval Analysis*, Proceedings of the 37th IEEE Conference on Decision & Control, Tampa, Florida USA, pp.3966-3971, 1998.
- [Kirpatrick et al, 1983] S. Kirpatrick, C. D. Jr. Gelatt et M. P. Vecchi, *Optimization by Simulated Annealing*, Science, Vol. 220, pp. 671-680, 1983.
- [Landau et al, 2001] I. D. Landau et A. Besançon-Voda (Eds), *Identification des systèmes*, série Systèmes automatisés dirigée par Claude Foulard, Information - Commande - Communication, Hermès Science Publications, 2001.
- [Leclerc, 1992] A. P. Leclerc, *Efficient and Reliable Global Optimization*, PhD Thesis, The Ohio State University, 1992.

- [Luenberger, 1971] D. G. Luenberger, *An introduction to Observers*, IEEE Trans. On Automatic Control, Vol. AC-16, pp. 596-602, 1971.
- [Luenberger, 1973] D. Luenberger, *Introduction to Linear and Nonlinear Optimisation*, Addison Wesley, 1973.
- [Lundy et Mees, 1986] M. Lundy et A. Mees, *Convergence of an Annealing Algorithm*, Mathematical Programming, Vol. 34, pp. 111-124, 1986.
- [Ljung, 1979] L. Ljung, *Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems*, IEEE Trans. On Automatic Control, vol. 24, pp. 36-50, 1979.
- [Ljung et Soderström, 1983] L. Ljung et T. Soderström, *Theory and practice of recursive identification*, MIT Press, 1983.
- [Michalewicz, 1999], Z. Michalewicz, *Genetic algorithms + Data structures = Evolution programs*, Springer-Verlag, 1999.
- [Michalska et Mayne, 1995] H. Michalska et D. Q. Mayne, *Moving Horizon Observers and Observer-based Control*, IEEE Trans. On Automatic Control, vol. 40, pp. 995-1006, 1995.
- [Milanese et Vicino, 1996] M. Milanese et A. Vicino, *Optimal estimation theory for dynamic systems with set membership uncertainty: an overview*, In *Bounding approaches to system identification*, M. Milanese, J. Norton, H. Piet-Lahanier et E. Walter (Eds), Plenum Press, New York and London, Chapter 2, pp. 5-27, 1996.
- [Moore, 1966] R. E. Moore, *Interval Analysis*, Prentice Hall, Englewood Cliffs, New Jersey, 1966.
- [Moore, 1979] R. E. Moore, *Methods and Applications of Interval Analysis*, *Studies in Applied Mathematics*, Philadelphia, Pennsylvania, 1979.
- [Moraal et Grizzle, 1995] P. E. Moraal et J. W. Grizzle, *Observer Design for Nonlinear Systems with Discrete-time Measurement*, IEEE Trans. On Automatic Control, vol. 40, pp. 395-404, 1995.
- [Murty et Kabadi, 1987] K. G. Murty et S. N. Kabadi, *Some NP-complete problems in quadratic and nonlinear programming*, Mathematical Programming, pp. 117, 1987.

- [Muske et Edgar, 1997] K. R. Muske et T. F. Edgar, *Nonlinear State Estimation*, In *Nonlinear Process Control*, M. A. Henson and D. E. Seborg (Eds), Prentice Hall, Chapter 6, Verlag, pp. 311-370, 1997.
- [Neumaier, 1990] A. Neumaier, *Interval Methods for Systems of Equations*, Encyclopedia of Mathematics and its Applications, Cambridge 1990.
- [Nijmeijer, 1982] H. Nijmeijer, *Observability of Autonomous Discrete-time Nonlinear Systems: A Geometric Approach*, Int. Journal of Control, vol. 36, pp. 867-874, 1982.
- [Othman, 1992] S. Othman, *Sur les observateurs des systèmes non-linéaires*, Thèse de Doctorat, Université Claude Berbard, Lyon I, 1992.
- [Papadimitriou et Steiglitz, 1998] C. H. Papadimitriou et K. Steiglitz, *Combinatorial optimization – Algorithms and complexity*, Dover Publications, Inc, 1998.
- [Pardalos et Vavasis, 1991] P. M. Pardalos et S. A. Vavasis, *Quadratic Programming with one negative eigenvalue is NP-Hard*, Journal of Global Optimization, Vol. 1, pp.15-22, 1991.
- [Peña y Lillo, 1999] M. Peña y Lillo, *Data processing and modelling of solid substrate cultivation bioreactors*, M.Sc. Thesis, Universidad Catolica de Chile, 1999.
- [Peña y Lillo *et al*, 2001] M. Peña y Lillo, R. Perez-Correa, E. Agosin et E. Latrille, *Indirect measurement of water content in an aseptic solid substrate cultivation pilot-scale bioreactor*. Biotechnology and Bioengineering, Vol. 76, No. 1, pp.44-51, 2001.
- [Pérez-Correa et Agosin, 1999] J. R. Pérez-Correa, E. Agosin, *Automation of solid substrate fermentation processes*, In: M. Flickinger, S. Drew (Eds.), *The Encyclopedia of Bioprocess Technology: Fermentation, Biocatalysis and Bioseparation*, John Wiley & Sons, pp.2429-2446, 1999.
- [Piazzi et Visioli, 1998] A. Piazzi et A. Visioli, *Global minimum-time trajectory planning of mechanical manipulators using interval analysis*, International Journal Control, Vol. 71, No. 4, pp. 631-652, 1998.
- [Piazzi et Visioli, 2000] A. Piazzi et A. Visioli, *Global minimum-Jerk trajectory planing of robot manipulators*, IEEE Transaction on Industrial Electronics, Vol. 47, No. 1, February 2000.

- [Rao *et al*, 2000] C. V. Rao, J. B. Rawlings et J. H. Lee, *Nonlinear Moving Horizon State Estimation*, in *Nonlinear Model Predictive Control, Progress in Systems and Control Theory*, F. Allgower and A. Zheng (Eds), pp. 45-70, Birkhauser, 2000.
- [Rao *et al*, 2001] C. V. Rao, J. B. Rawlings, J. H. Lee, *Constrained linear state estimation-a moving horizon approach*, *Automatica*, Vol. 37, pp. 1619-1628, 2001.
- [Reif *et al*, 1999] K. Reif, S. Günther, E. Yaz et R. Unbehauen, *Stochastic Stability of the Discrete-Time Extended Kalman Filter*, *IEEE Trans. On Automatic Control*, Vol. 44, No. 4, pp. 714-728, April 1999.
- [Rinnooy-Kan, 1984] A. H. G. Rinnooy-Kan et G.T. Timmer. *Stochastic methods for global optimization*. American Journal of Mathematical and Management Sciences. Vol. 4, No.1-2, pp. 7-40, 1984
- [Romero-Jimenez, 1996] G. Romero-Jiménez, *Modélisation Semiqualitative de Bioprocédés en vue de leur Conduite*, Thèse de Doctorat, Laboratoire d'Automatique de Grenoble, INPG, Grenoble, 1996
- [Sargantanis et Nazmul-Karim, 1994] J. Sargantanis et M. Nazmul-Karim, *Multivariable Iterative Extended Kalman Filter Based Adaptive Control: Case Study of Solid Substrate Fermentation*, *Ind. Eng. Chem. Res.* Vol. 33, pp.878-888, 1994.
- [Song et Grizzle, 1995] Y. Song et J. W. Grizzle, *The Extended Kalman Filter as a Local Asymptotic Observer for Discrete-Time Nonlinear Systems*, *IEEE Trans. on Automatic Control*, Vol. 40, pp. 395-404, 1995.
- [Sontag, 1984] H. Sontag, *A Concept of Local Observability*, *Systems & Control Letters*, vol. 5, pp. 41-47, 1984.
- [Sussmann, 1979] H. J. Sussmann, *Single Input Observability of Continuous Time Systems*, *Math. Systems Theory*, vol. 12, pp. 371-393, 1979.
- [Thévenon, 2000] L. Thévenon, *Représentation de Systèmes Hybrides Complexes par Flux de Données : Développement d'un Outil de Modélisation et de Simulation des Procédés Batch*, Thèse de Doctorat, Laboratoire d'Automatique de Grenoble, INPG, Grenoble, 2000.
- [Thévenon et Flaus, 2000] L. Thévenon et J.M. Flaus, *Modular Representation of Complex Hybrid Systems: Application to the Simulation of Batch Processes*, *Journal Simulation Practice and Theory (SIMPRA)*, Vol. 227, pp.1-25, 2000.

- [Van-Iwaarden, 1996] AR. J. Van-Iwaarden, *An improved unconstrained global optimization algorithm*, PhD Thesis in applied mathematics, University of Colorado at Denver, 1996.
- [Valdés-González et al, 2000a] H.Valdés-González, JM Flaus et G. Acuña, *Moving horizon state estimation with simulated annealing algorithm applied to a biotechnological process*, Proceedings of the Foodsim'2000, pp.166-170, Nantes, France June 2000.
- [Valdés-González et al, 2000b] H Valdés-González, G Acuña, D Bassi, R Pérez et JM Flaus, *New approaches to simultaneous state observation and parameter estimation for biotechnological processes*, Proceedings of the CONTROLO'2000 conference, Portugal, October 2000.
- [Valdés-González et Flaus, 2001a] H Valdés-González et J.M Flaus, *A globally convergent state estimation method based on interval techniques*, Proceedings of the European Control Conference, CD-ROM, ECC'2001, Porto, Portugal, 2001.
- [Valdés-González et Flaus, 2001b] H Valdés-González et JM Flaus, *State estimation in a bioprocess described by a hybrid model*, Proceedings of the IEEE Joint International Conference on Control Applications & International Symposium on Intelligent Control, Invited Session, Mexico city, Mexico, September 2001.
- [Valdés-González et al, 2001c] H Valdés-González, G Acuña, JM Flaus et Ricardo Pérez, *Heuristical approaches to state observation and parameter estimation for biotechnological processes*, Proceedings of the 8th International Conference on Computer Applications in Biotechnology (CAB 8), Quebec, Canada, June 2001.
- [Valdés-González et Flaus, 2002a] H Valdés-González et JM Flaus, *Numerical observability in dynamical systems: an approach by interval numbers*, Proceedings of the 15th IFAC World Congress, Barcelona 2002.
- [Valdés-González et al, 2002b] H Valdés-González, JM Flaus & G Acuña, *Moving Horizon State Estimation with Global Convergence Using Interval Techniques: Application to Biotechnological Processes*, (Article in press) à paraître dans la revue - Journal of Process Control, Vol. 13, No. 4, pp. 325-336, June 2003.
- [Valdés-González et al, 2002c] H Valdés-González, JM Flaus, G. Acuña et DJ. Boukraa, *State Estimation in Biotechnology: A Multi Model Interval Observer*,

Proceedings of the 10th Mediterranean Conference on Control and Automation, Lisbon, Portugal 2002.

[Valdés-González *et al*, 2003] H Valdés-González, JM Flaus, G. Acuña et R. Pérez-Correa, *Fault detection based on a multi-model interval observer: application to solid substrate cultivation*, Soumis à la revue Control Engineering Practice, 2003.

[Zimmer, 1993] G. Zimmer, *A New Algorithm for Approximating the States of Non-linear Systems*, Int. Journal of Systems Science, vol. 24, No 4, pp. 777-788, 1993.

[Zimmer, 1994] G. Zimmer, *State Observation by On-line minimization*, Int. Journal of Control, vol. 60, pp. 595-606, 1994.

Annexe A

Analyse de la complexité de l'algorithme 3 d'optimisation globale

L'algorithme d'optimisation globale ou Algorithme-3, nommé G-Optimisation(.), a une complexité informatique, qui est donné dans le pire des cas et pour un critère non différentiable par :

$$O(f(n)) \approx O\left(2^{\sum_{i=1}^n \lambda_i} \cdot n\right)$$

Voici une version de l'algorithme d'optimisation globale présentée dans la *section 3.2.6*, mais pour un critère non différentiable (voir remarque 13).

Algorithme d'OPTIMISATION GLOBALE

Entrée : $[\Omega], B_\varepsilon, V(\cdot), [\varphi](\cdot)$

$[\Omega]$: Domaine initial admissible sur \mathbb{R}^n .

B_ε : Tolérance ou précision prescrite d'optimisation.

$V(\cdot)$: Critère à optimiser.

$\varphi(\cdot)$: Ensemble des restrictions du problème (si disponibles).

Sortie : Liste $\{[x]\}$

Liste $\{[x]\}$: Liste de boîtes dans lesquelles le minimum global est contenu.

- **Ajouter $[\Omega]$ à la liste**

$$\text{Liste}\{[x]\} := \{[\Omega]\}$$

- **Calculer le côté le plus large de la première boîte dans la liste**

$$d := w([\Omega]) \quad (\text{Voir section 2.1.1})$$

- **Calculer $V(\cdot)$ sur $[\Omega]$**

$$V([\Omega]) = [LV_{[\Omega]} UV_{[\Omega]}]$$

$$UV := UV_{[\Omega]} \quad (\text{Une borne supérieure de la valeur minimale de } V(\cdot))$$

- **Boucle principale**

1. WHILE $d > B_\epsilon$

1.1 Enlever la première boîte $[x]$ de la liste $\text{Liste}\{[x]\}$

1.2 Bissection

Couper en deux la boîte $[x]$ par son côté le plus large tel que

$$[x] = [x_1] \cup [x_2]$$

1.3 Calculer $V(\cdot)$ sur $[x_1]$ afin d'obtenir $V([x_1]) = [LV_{[x_1]} UV_{[x_1]}]$

1.4 Calculer le point central $\beta_{[x_1]} = c([x_1])$ en $V(\cdot)$ afin d'obtenir $UV_{\beta_{[x_1]}} = \beta_{[x_1]}$

1.5 Calculer $V(\cdot)$ sur $[x_2]$ afin d'obtenir $V([x_2]) = [LV_{[x_2]} UV_{[x_2]}]$

1.6 Calculer le point central $\beta_{[x_2]} = c([x_2])$ en $V(\cdot)$ afin d'obtenir $UV_{\beta_{[x_2]}} = \beta_{[x_2]}$

1.7 Test de faisabilité sur $[x_1]$: Evaluer $\varphi(\cdot)$ sur $[x_1]$

1.8 Test de faisabilité sur $[x_2]$: Evaluer $\varphi(\cdot)$ sur $[x_2]$

1.9 Si $[x_1]$ est faisable

1.9.1 IF $LV_{[x_1]} < UV$

1.9.1.1 Ajouter $[x_1]$ à la fin de la $\text{Liste}\{[x]\}$

1.9.1.2 IF $UV_{\beta_{[x_1]}} < UV$

1.9.1.2.1 $UV := UV_{\beta_{[x_1]}}$

1.10 Si $[x_2]$ est faisable

1.10.1 IF $LV_{[x_2]} < UV$

1.10.1.1 Ajouter $[x_2]$ à la fin de la $\text{Liste}\{[x]\}$

1.10.1.2 IF $UV_{\beta_{[x_2]}} < UV$

1.10.1.2.1 $UV := UV_{\beta_{[x_2]}}$

1.11 Effacer de la liste, $\text{Liste}\{[x]\}$, la boîte $[x]$

1.12 Calculer la longueur de tous les côtés de la boîte.

1.13 $d :=$ la longueur maximale de l'étape 1.12.

A.1 Complexité de l'algorithme

On appelle complexité d'un algorithme, le nombre total d'opérations qui doivent être réalisées pour exécuter la totalité de l'algorithme en question. Cette complexité est désignée par $O(f(n))$ ou "de l'ordre de $f(n)$ ". L'analyse effectuée doit supposer le calcul de la complexité dans le pire des cas.

L'algorithme d'optimisation globale par intervalles proposée correspond à une méthode dans laquelle nous cherchons à trouver un ensemble de solutions, en décomposant une boîte initiale $[\Omega]$ en sous-boîtes $[x_1]$ et $[x_2]$ (toujours par bisection des boîtes). Cette décomposition est réitérée pour chacune des sous-boîtes. Nous pouvons représenter ce processus par l'arbre binaire suivant.

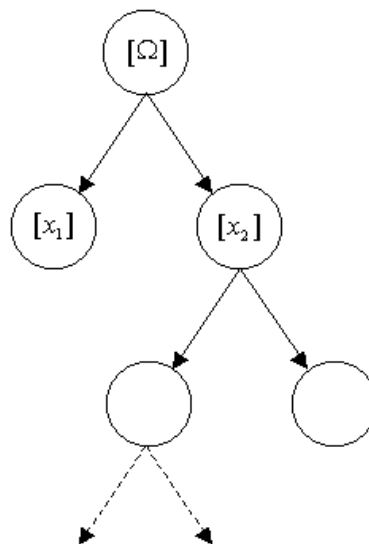


Figure A.1 : Arbre généré par les bisections des boîtes

La profondeur de l'arbre est limitée (seulement si l'espace de recherche est borné). Cette profondeur est le nombre maximal de décompositions de la boîte initiale $[\Omega]$ en boîtes faisables plus petites. Cela correspond au nombre de fois que l'étape 1 de l'algorithme sera exécuté. Nous nommerons L à cette profondeur.

Soie n la dimension de l'entrée, qui correspond ici à celle du vecteur d'état.

Le programme principal est alors inclus entièrement dans une boucle **WHILE**, et cette opération est exécutée L fois. Alors la complexité de l'étape 1 est : $O(L)$. Cette complexité, doit être multiplié par l'ordre de la complexité de toutes les étapes qui sont incluses dans la boucle **WHILE**.

- L'étape 1.1, correspond à la lecture de tous les éléments d'une matrice M ($M \in \mathbb{R}^{n \times 2}$). Cette action revient à effectuer $2n$ opérations de lecture. La complexité de cette partie est donc égale à : $O(2n) \approx O(n)$.

$$M = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{i1} & a_{i2} \\ \vdots & \vdots \\ a_{n1} & a_{n2} \end{bmatrix} \Rightarrow \text{" 2n " positions à lire.}$$

- L'étape 1.2 consiste à copier $[x]$ dans $[x_1]$ et $[x_2]$. C'est à dire, $O(2n) + O(2n)$ opérations d'écriture. Nous prenons ensuite la ligne la plus large (*ligne i*) de la matrice, et nous effectuons une opération de bisection de ses colonnes $O(2)$ (calcul de la moyenne). Donc la complexité de cette partie est égale à : $O(2n) + O(2n) + O(2) \approx O(n)$

$$[x] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{i1} & a_{i2} \\ \vdots & \vdots \\ a_{n1} & a_{n2} \end{bmatrix} \Rightarrow a'_i = \frac{a_{i2} + a_{i1}}{2} \Rightarrow [x_1]_{i2} = a'_i \quad \wedge \quad [x_2]_{i2} = a'_i$$

$$[x] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{i1} & a_{i2} \\ \vdots & \vdots \\ a_{n1} & a_{n2} \end{bmatrix} \Rightarrow [x_1] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{i1} & a'_i \\ \vdots & \vdots \\ a_{n1} & a_{n2} \end{bmatrix} \quad [x_2] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a'_i & a_{i2} \\ \vdots & \vdots \\ a_{n1} & a_{n2} \end{bmatrix}$$

- Les étapes 1.3 et 1.5 sont similaires. Dans ces étapes nous effectuons un nombre d'opérations qui dépend du degré du polynôme de la fonction objectif $V(\cdot)$. Soit n_j le degré du polynôme de la fonction objectif, alors la complexité de cette partie est égale à : $O(n_j)$.
- Les étapes 1.4 et 1.6 sont similaires. Dans ces étapes nous effectuons un nombre d'opérations qui dépend aussi du degré du polynôme de la fonction objectif $V(\cdot)$. Alors la complexité de cette partie est aussi égale à : $O(n_j)$.
- Les étapes 1.7 et 1.8 sont similaires. Elles correspondent à des comparaisons pour détecter la faisabilité. Le nombre de comparaisons dépend du nombre des fonctions restriction du problème. Soit n_φ le nombre de fonctions restriction du problème. La complexité de cette partie est égale à : $O(n_\varphi)$.
- Les étapes 1.9 et 1.10 sont similaires. L'opération qui est effectuée est simplement une comparaison, sa complexité est donc de l'ordre de $O(1)$, multiplié par l'ordre de la complexité de toutes ses sous-étapes.
- L'étape 1.9.1 est aussi une comparaison ; sa complexité est de l'ordre de $O(1)$, multiplié par l'ordre de la complexité de toutes ses sous-étapes.
 - L'étape 1.9.1.1 est une insertion (écriture) de tous les éléments d'une matrice M ($M \in \mathbb{R}^{n \times 2}$) dans une liste. Ceci correspond à effectuer $2n$ opérations d'insertion. Donc la complexité de cette partie est égale à : $O(2n) \approx O(n)$.
 - L'étape 1.9.1.2 est une comparaison. Sa complexité est de l'ordre de $O(1)$, multiplié par l'ordre de la complexité de l'étape 1.9.1.2.1, qui est de l'ordre de $O(1)$ car elle correspond à une opération d'assignation. Donc, la complexité de ces deux étapes est : $O(1) \cdot O(1) \approx O(1)$.
- Nous pouvons donc voir que la complexité des étapes 1.9 et 1.10 est égale à :

$$O(1) \cdot [O(1) \cdot (O(n) + O(1))] \approx O(n)$$
- L'étape 1.11 consiste à effacer tous les éléments d'une matrice M ($M \in \mathbb{R}^{n \times 2}$). Cela correspond à effectuer $2n$ opérations d'éliminations, donc la complexité de cette partie est égale à : $O(2n) \approx O(n)$.

- L'étape 1.12 correspond à calculer la soustraction des valeurs disposées en colonnes, pour chaque ligne d'une matrice M ($M \in \mathbb{R}^{n \times 2}$). Il y a donc n opérations de soustraction. La complexité de cette partie est égale à : $O(n)$
- L'étape 1.13 consiste à parcourir une liste de n éléments, puis de choisir le plus grand élément. Dans le pire des cas, il y a n opérations de comparaisons à effectuer. La complexité de cette partie, est égale à : $O(n)$
- Finalement la complexité de l'algorithme d'optimisation par intervalles, est égal à :

$$O(f(n)) \approx O(L) \cdot \left[\begin{array}{l} O(n) + O(n) + O(nj) + O(nj) + O(nj) + O(nj) + O(n\varphi) \\ + O(n\varphi) + O(n) + O(n) + O(n) + O(n) + O(n) \end{array} \right]$$

C'est à dire :

$$\begin{aligned} O(f(n)) &\approx O(L) \cdot [O(n) + O(nj) + O(n\varphi)] \\ O(f(n)) &\approx O(L \cdot [n + nj + n\varphi]) \end{aligned}$$

L'algorithme d'optimisation globale correspond donc à un algorithme dont la complexité informatique est $O(f(n)) \approx O(L \cdot [n + nj + n\varphi])$. Cependant, nous pourrions supposer, sans perdre généralité, que pour chaque exemple particulier, nj et $n\varphi$ sont des valeurs constantes. La complexité informatique est alors de la forme :

$$O(f(n)) \approx O(L \cdot n)$$

Il nous faut donc connaître la profondeur L de la liste pour connaître la complexité informatique.

A.1.1 Détermination de la profondeur L

Soit $[\Omega] \in \mathbb{I}\mathbb{R}^n$ le domaine admissible pour le problème d'optimisation. Alors en regardant l'étape numéro 1 de l'algorithme d'optimisation, nous pouvons constater que le pire des cas (profondeur maximale de la liste), sera obtenu quand tout le

domaine admissible $[\Omega]$ est faisable, et tout ce domaine est un minimum global pour n'importe quel précision d'optimisation B_ε .

Lorsqu'une une boîte est coupée en deux, la bisection est faite juste au milieu du côté le plus large de la boîte. L'algorithme nous montre qu'à chaque nouvelle itération, nous coupons une boîte par le milieu, chaque sous-boîte obtenue étant elle même coupée en deux en son milieu à l'itération suivante.

Par conséquent, la taille des sous-boîtes de la liste suit une progression géométrique :

$$a_N = a \cdot r^{N-1}$$

Où a est la largeur du côté de la boîte à couper, a_N est la taille de la boîte générée par la $N^{\text{ième}}$ coupure du côté a , r la raison de la suite. N est donc le nombre total de coupures qui doivent être effectuées sur ce côté de la boîte ($N \in \mathbb{N}$, nombres naturels). Cette progression continue jusqu'à ce que la taille de la plus petite boîte de la liste soit plus petite que B_ε .

Considérons l'étape numéro 1. Nous cherchons un N tel que :

$$B_\varepsilon > a \cdot r^{N-1}$$

Nous pouvons donc dire que le nombre de coupures N nécessaires pour attendre notre but sera :

$$N < 1 + \frac{\ln\left(\frac{B_\varepsilon}{a}\right)}{\ln\left(\frac{1}{2}\right)}$$

Comme il y a n variables d'état, nous devons calculer la quantité de coupures pour tous les n côtés de la boîte, ce qui nous donne :

$$N_i < 1 + \frac{\ln\left(\frac{EPSB}{w([\Omega]_i)}\right)}{\ln\left(\frac{1}{2}\right)} \quad (i : 1 \rightarrow n)$$

Alors le nombre de boîtes générées par les N_i coupures correspond à la profondeur L de la liste, c'est à dire :

$$L = 2^{\sum_{i=1}^n N_i}$$

Finalement, l'algorithme d'optimisation globale a une complexité informatique dans le pire des cas de la forme :

$$O(f(n)) \approx O(L \cdot n) = O\left(2^{\sum_{i=1}^n N_i} \cdot n\right)$$

Annexe B

Application de `ia.jar` à un problème différentiable d'optimisation globale

Exemple : Soit le domaine admissible $[\Omega] = [-4, 4] \times [-4, 4]$, et une précision donnée $B_\varepsilon = 0.05$ pour le problème d'optimisation globale sans restrictions qui suit :

$$\begin{aligned} \text{Minimiser (Globalement)} \quad V(x_1, x_2) &= \frac{1}{2} \sum_{j=1}^2 (x_j^4 - 16 \cdot x_j^2 + 5 \cdot x_j) \\ x_j &\in [\Omega] \end{aligned}$$

La solution est :

$$[x]^* = \begin{bmatrix} -2.9375 & -2.8750 \\ -2.9375 & -2.8750 \end{bmatrix}$$

D'abord voyons la forme du critère. Il y a trois minimums locaux et un global (figure B.1). Dans les prochaines figures nous allons voir comment fonctionne la méthode. Nous verrons en particulier que plus la tolérance B_ε définie est petite, plus la méthode est efficace pour supprimer les minimums locaux de $V(\cdot)$ dans $[\Omega]$, autrement dit, plus elle est efficace pour effacer des boîtes contenues dans la liste qui ne contiennent pas un minimum global.

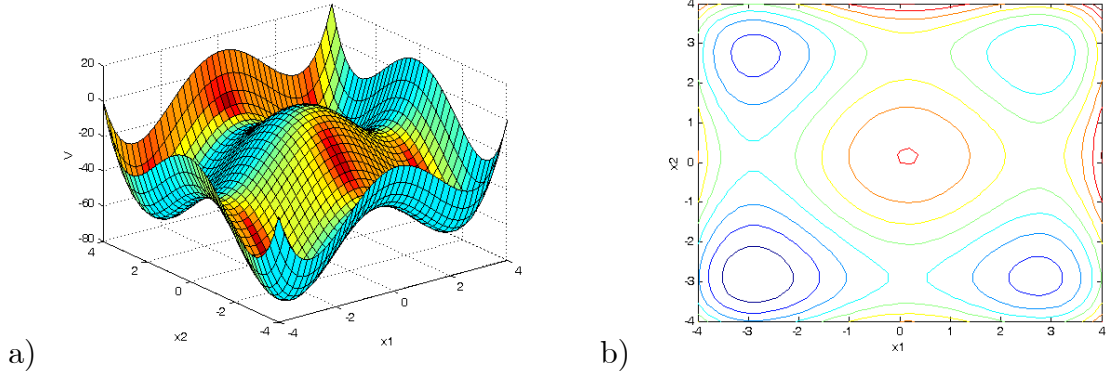


Figure B.1: a) Forme du critère b) Courbes de niveaux du critère $V(.)$

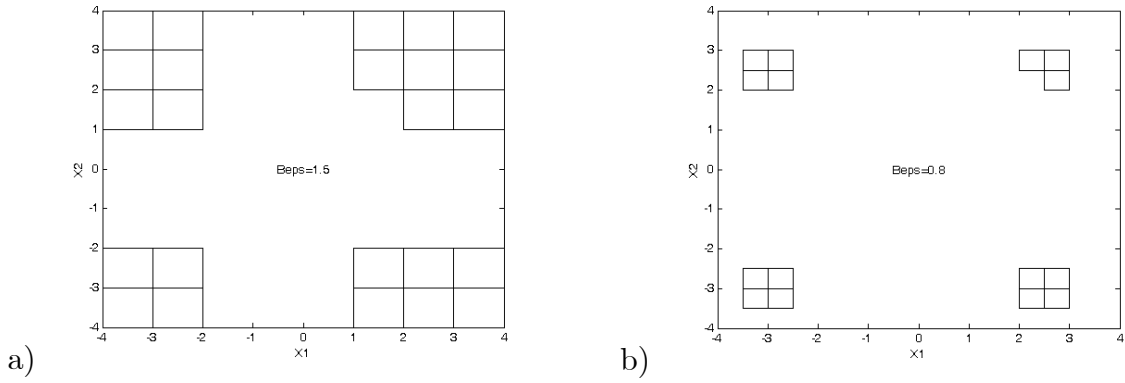


Figure B.2: a) Quatre zones candidates au minimum global avec $B_\epsilon = 1.5$ b) Quatre minimums identifiés avec $B_\epsilon = 0.8$

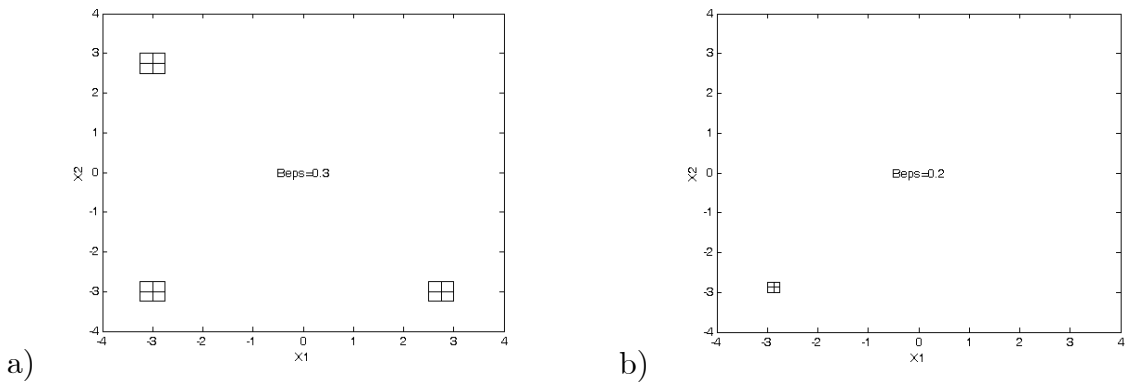


Figure B.3: a) Un minimum local effacé avec $B_\epsilon = 0.3$ b) Le minimum global est trouvé avec $B_\epsilon = 0.2$

Annexe C

L'algorithme de la méthode MHSE-BFGS

L'algorithme itératif suivant nous donne une estimation de l'état du système dynamique (1.1) à la fin de l'horizon choisi, c'est-à-dire au temps $\ell = sh + lh$. Cet algorithme utilise la terminologie suivante [Boillereaux, 1996] :

- \hat{x}_{sh}^* : Vecteur d'état en début d'horizon qui minimise le critère.
- \hat{x}_{sh}^i : Le résultat de l'estimation obtenue à la $i^{\text{ème}}$ itération.
- sh : Début de l'horizon
- lh : Longueur de l'horizon

Algorithme()

1. Initialisation du vecteur d'état : \hat{x}_{sh}
2. Recherche du vecteur d'état initial optimal : \hat{x}_{sh}^*
 - a) Calcul de la sortie prédite sur l'horizon
 - b) Calcul du gradient et du hessien du critère en \hat{x}_{sh}
 - c) Effectuer: $\hat{x}_{sh}^{i+1} = \hat{x}_{sh}^i - \lambda_i \cdot [\nabla^2 J(\hat{x}_{sh}^i)]^{-1} \cdot \nabla J(\hat{x}_{sh}^i)$, tant que $\nabla J(\hat{x}_{sh}^i) \neq 0$
 - d) Si $\nabla J(\hat{x}_{sh}^i) = 0$, alors $\hat{x}_{sh}^i = \hat{x}_{sh}^*$
3. Calcul de la solution en fin d'horizon (instant courant ℓ) : \hat{x}_ℓ^*
4. Retour à l'étape 2 pour le calcul de $\hat{x}_{\ell+1}^*$ avec :
 - a) $sh=sh+1$
 - b) \hat{x}_{sh} la prédiction à un pas de \hat{x}_{sh}^* (entre $k = sh$ et $k = sh+1$)

L'étape 3 du dernier algorithme utilise la méthode d'optimisation de Newton. Cette méthode d'optimisation présente une convergence quadratique, quand \hat{x}_{sh}^i est suffisamment proche de l'optimum \hat{x}_{sh}^* , cette condition est la solution pour garantir que le hessien reste défini positif. Alors le principal inconvénient de cette méthode d'optimisation est le besoin de calculer, garder et faire l'inversion du hessien à chaque itération.

Cependant il y en a plusieurs façons de résoudre le problème concerné par l'inversion du hessien. La solution consiste à faire une approximation de la matrice du hessien, qui sera actualisée à chaque itération. Pour faire cette approximation nommée H , ou celle de son inverse H^{-1} , les méthodes les plus connues sont les algorithmes de Davidon, Fletcher et Powell (année 1963), ou par son sigle DFP, la méthode de Broyden, Fletcher, Goldfarb et Shanno (1970) connue comme BFGS. Seule cette dernière sera présentée ici, mais les autres peuvent être retrouvées par exemple dans [Luenberger, 1973].

$$H_{k+1} = H_k + \frac{q_k \cdot q_k^T}{p_k^T \cdot q_k} - \frac{H_k \cdot p_k \cdot p_k^T \cdot H_k}{p_k^T \cdot H_k \cdot p_k}$$

$$H_{k+1}^{-1} = \left(I - \frac{p_k \cdot q_k^T}{p_k^T \cdot q_k} \right) \cdot H_k^{-1} \cdot \left(I - \frac{q_k \cdot p_k^T}{p_k^T \cdot q_k} \right) + \frac{p_k \cdot p_k^T}{p_k^T \cdot q_k}$$

où

- a) $H_0 = H_0^{-1} = I$, la matrice identité.
- b) $p_k = x_{k+1} - x_k$
- c) $q_k = \nabla J_{k+1} - \nabla J_k$

Annexe D

L'algorithme du Filtre de Kalman Etendu

Il est possible d'étendre la technique du Filtre de Kalman linéaire à des systèmes non-linéaires, en calculant les gains des estimateurs à partir du modèle du procédé par linéarisation tangentielle autour d'un point de fonctionnement particulier. Cet estimateur s'appelle Filtre de Kalman étendu (FKE) [Grewal *et* Andrews, 1993] , [Brown *et* Hwang, 1997] , [Brookner, 1998].

Si nous considérons le système non-linéaire suivant :

- Modèle dynamique non-linéaire :

$$x_k = f(x_{k-1}, k-1) + \omega_k$$
$$\omega_k \sim N(0, Q_k)$$

- Modèle non-linéaire de la sortie :

$$z_k = h(x_k, k) + v_k$$
$$v_k \sim N(0, R_k)$$

Alors l'implémentation non-linéaire des équations de l'algorithme récursif du FKE est le suivant :

- Obtention de la prédiction de l'état estimé :

$$\hat{x}_k^{(-)} = f(\hat{x}_{k-1}^{(+)}, k-1)$$

- Calcul de la prédiction de la sortie :

$$\hat{z}_k = h(x_k^{(-)}, k)$$

- Approximation linéaire de la prédiction :

$$\Phi_{k-1}^{[1]} \approx \left. \frac{\partial f(x, k-1)}{\partial x} \right|_{x=\hat{x}_{k-1}^{(-)}}$$

- Approximation linéaire de la sortie :

$$H_{k-1}^{[1]} \approx \left. \frac{\partial h(x, k)}{\partial x} \right|_{x=\hat{x}_k^{(-)}}$$

- Obtention de la prédiction de l'état estimé conditionné :

$$\hat{x}_k^{(+)} = \hat{x}_k^{(-)} + K_k \cdot (z_k - \hat{z}_k)$$

- Obtention à priori de la matrice de covariance :

$$P_k^{(-)} = \Phi_{k-1}^{[1]} \cdot P_{k-1}^{(+)} \cdot \Phi_{k-1}^{[1]T} + Q_{k-1}$$

- Obtention du gain de Kalman :

$$K_k = P_k^{(-)} \cdot H_k^{[1]T} \cdot [H_k^{[1]} \cdot P_k^{(-)} \cdot H_k^{[1]T} + R_k]^{-1}$$

- Obtention à posteriori de la matrice de covariance :

$$P_k^{(+)} = \{I - K_k \cdot H_k^{[1]}\} \cdot P_k^{(-)}$$

Annexe E

L'algorithme heuristique MHSE-SA et une application

L'algorithme heuristique suivant nous donne une estimation de l'état du système dynamique (1.1) à la fin de l'horizon choisi, c'est-à-dire au temps $\ell = sh + lh$. Cet algorithme utilise la terminologie suivante [Valdés-González et al, 2000a], [Valdés-González et al, 2001c] :

- \hat{x}_{sh}^* : Vecteur d'état en début d'horizon qui minimise le critère.
- \hat{x}_{sh}^i : Le résultat de l'estimation obtenue à la $i^{\text{ième}}$ itération.
- sh : Début de l'horizon
- lh : Longueur de l'horizon

Algorithme()

1. Initialisation du vecteur d'état : \hat{x}_{sh}
2. Recherche du vecteur d'état initial optimal : \hat{x}_{sh}^*
 - a) Calcul de la sortie prédite sur l'horizon
 - b) Evaluation de la procédure heuristique d'optimisation SA en \hat{x}_{sh}
3. Calcul de la solution en fin d'horizon (instant courant ℓ) : \hat{x}_ℓ^*
4. Retour à l'étape 2 pour le calcul de $\hat{x}_{\ell+1}^*$ avec :
 - a) $sh=sh+1$
 - b) \hat{x}_{sh} la prédiction à un pas de \hat{x}_{sh}^* (entre $k = sh$ et $k = sh+1$)

E.1 Procédure d'optimisation SA

L'algorithme de base recuit simulé ou SA (Simulated Annealing) qui cherche à minimiser un critère $J(.) \in \mathbb{R}$ [Michalewicz 1999], [Kirpatrick, 1983], [Díaz *et al*, 1996] [Dekker et Aarts, 1991] est le suivant :

- T_0 : Température initiale
- T_f : Température finale
- $\bar{\alpha}$: Vitesse de refroidissement
- L : Nombre d'itérations possibles entre chaque changement de température

Algorithme($T_0, \bar{\alpha}, L, T_f$)

1. $T \leftarrow T_0$
2. $S_a \leftarrow$ *Solution Initiale*
3. WHILE $T \geq T_f$ DO
 - FOR $i = 1 \rightarrow L(T)$
 - $S_c \leftarrow$ Génération d'une solution depuis S_a
 - $\delta = J(S_c) - J(S_a)$
 - IF $N(0,1) < e^{(-\delta/T)} \vee \delta < 0$
 - $S_a \leftarrow S_c$
 - END
- END
- $T \leftarrow \bar{\alpha} \cdot T$
- END
4. La solution au problème d'optimisation est la meilleure solution candidate visitée.

Dans cet algorithme :

- S_a : Solution actuelle
- S_c : Solution candidate
- $\bar{\alpha} \in [0,1]$, normalement fixé à $\bar{\alpha} = 0.9$ [Kirpatrick, 1983]

Pour compléter le programme, la température initiale T_0 peut être trouvée empiriquement. Le nombre L d'itérations entre chaque changement de température peut être aussi déterminé empiriquement. Pour l'algorithme décrit sa convergence en probabilité a été prouvée, indépendamment de la solution initiale [Lundy *et* Mees, 1986], [Diaz *et al.* 1996].

E.2 Application

Considérons le modèle du bioprocédé sur substrat solide présenté au cas 4 du chapitre 6, dont le modèle est décrit par les équations (6.9)-(6.16). Deux tests différents ont été considérés pour évaluer la performance et la robustesse de la méthode MHSE couplée à l'algorithme d'optimisation heuristique SA. Les tests considèrent une biomasse initiale de $2.47e-3$ (gr/gr), les autres valeurs sont nulles :

- a) Résultats sous conditions idéales (0% erreur et aucune perturbation).
- b) Résultats avec une erreur initiale de 500% sur l'état variable et affecté par un bruit gaussien de 2% sur les variables CO_2 et O_2 mesurées.

Résultats test a) :

Une très bonne cohérence peut être trouvée entre les valeurs simulées et estimées de la biomasse (X), le substrat (NI) et le taux de croissance spécifique de croissance (μ). Ces estimation sont montrées dans les figures E.1, E.2 et E.3.

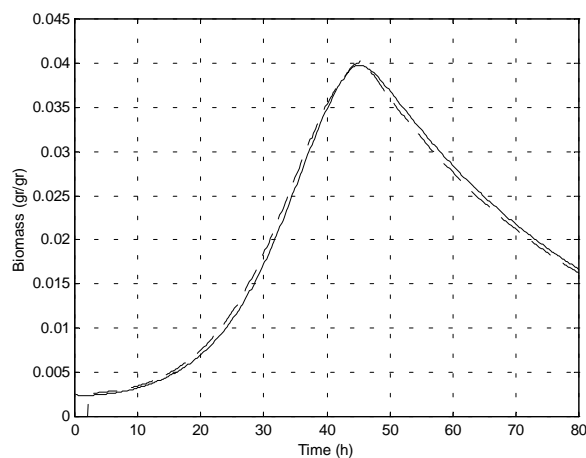


Figure E.1 : Biomasse simulée (ligne continue) et estimée (ligne pointillée)

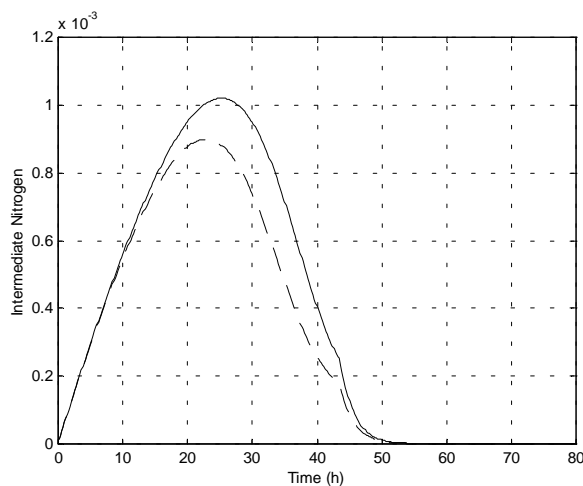


Figure E.2 : Azote intermédiaire simulé (ligne continue) et estimé (ligne pointillée)

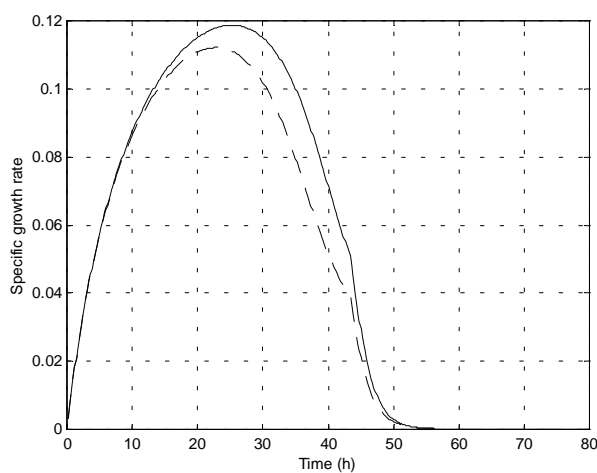


Figure E.3 : Taux spécifique de croissance simulée (ligne continue)
et estimée (ligne pointillée)

Résultats test b) :

Une perturbation initiale sur l'état (variable biomasse) est rapidement compensée et filtrée par la méthode à horizon fuyant. Cette estimation est faite en présence du bruit additif de 2% qui affecte la sortie. L'erreur de 500% sur la condition initiale de la biomasse nous aide à montrer clairement les propriétés de convergence et stabilité de la méthode, bien qu'une perturbation d'une telle magnitude ne soit pas réaliste. Les résultats sont montrés dans les figures E.4, E.5 et E.6.

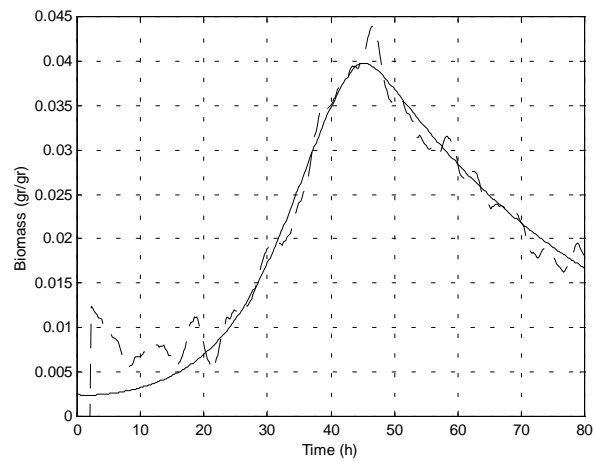


Figure E.4 : Biomasse simulée (ligne continue) et estimée (ligne pointillée)

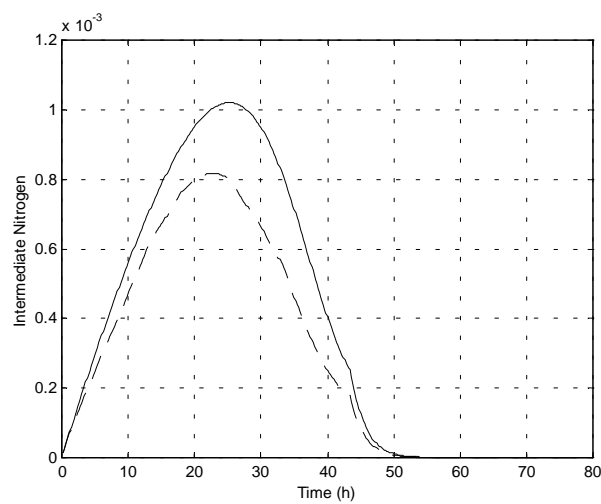


Figure E.5 : Azote intermédiaire simulé (ligne continue) et estimé (ligne pointillée)

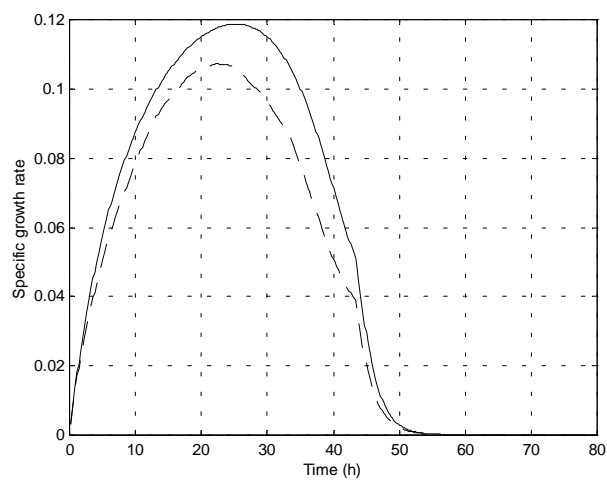


Figure E.6 : Taux spécifique de croissance simulée (ligne continue)
et estimée (ligne pointillée)

RESUME

Cette thèse propose une méthode originale d'estimation ensembliste d'états de procédés non-linéaires discrets, qui est globalement convergente. La méthode est basée sur une technique d'estimation à horizon glissant par intervalles (IMHSE), couplé à une technique d'optimisation globale de fonctions non-linéaires qui utilise l'arithmétique par intervalles.

En d'autres termes, la méthode IMHSE résout le problème d'estimation d'état d'un système dynamique par un problème statique d'optimisation globale non-linéaire par intervalles, sur un horizon de temps prédéfini. Les mesures faites hors ligne dans un procédé peuvent être utilisées facilement dans cet observateur ensembliste pour reconstruire les variables de l'état qui sont représentés par intervalles. Ce travail considère aussi la détection de dysfonctionnement d'un modèle en utilisant un observateur IMHSE multi-modèles (une propriété de plus donnée à notre observateur). L'objectif de cette approche multi-modèles est de détecter les variations dynamiques des paramètres du modèle dans le temps. Ces variations sont prises en considération en utilisant plusieurs modèles différents. Ces modèles seront commutés par notre observateur ensembliste pour reconstruire les états du système. Mis d'une façon simple, cette approche consiste à utiliser un modèle nominal pour l'état et d'autres modèles pour décrire les situations possibles de fonctionnement anormal (paramètres perturbés). L'algorithme nous permet de connaître en ligne quel est le meilleur modèle qui décrit le comportement réel du système. La technique proposée a été appliquée sur des modèles de procédés complexes biotechnologiques tel que la fermentation sur substrat solide, et à des bioprocédés décrit par des modèles hybrides. Les résultats obtenus par simulation montrent que ce type d'observateur a des avantages sur les autres observateurs et filtres, et qu'il peut être facilement appliqué dans un contexte industriel.

A CONTRIBUTION TO THE MOVING HORIZON STATE ESTIMATION BY INTERVAL TECHNIQUES: APPLICATIONS TO SUPERVISION AND MALFUNCTION DETECTION IN BIOPROCESSES

ABSTRACT

This thesis proposes an original method to estimate states in non-linear discrete-time systems with global convergence properties. The method is based on an Interval Moving Horizon State Estimation Method (IMHSE), which is coupled to a technique of global optimisation of non-linear functions that uses interval arithmetic. In other words, the principal idea is to transform the problem of state estimation from a dynamic system into a static problem of global non-linear optimisation over a considered time horizon by interval analysis. Offline measures or delayed measures can be easily used in this interval observer to reconstruct the state variables that are described using a representation by interval numbers. The work also considers model fault detection by a multimodel IMHSE (as an extra property given to our observer). The goal of this multimodel observer approach is to detect dynamic variations of the involved model parameters in time. These variations are taken into account using several different models that are commuted and used by our interval observer to reconstruct the states of the system. Put simply, this approach consists of using a model for the nominal dynamic state(s) and other models to describe situations of anomalous working (perturbed parameters). The algorithm allows us to know on line which model best describes the behaviour of the system. The proposed technique is applied to biotechnological complex process models such as solid substrate fermentation, and to bioprocesses described by a hybrid model. The results obtained through experimental and computer simulation demonstrate that this kind of estimator has advantages over other observers and filters, and that it can be easily implemented in an industrial context.

SPECIALITE : AUTOMATIQUE - ROBOTIQUE

MOTS-CLES : Estimation d'état, MHSE, Horizon glissant / fuyant, Méthodes ensemblistes,
Systèmes complexes, Observabilité, Multi-modèles.

LABORATOIRE D'AUTOMATIQUE DE GRENOBLE - UJF - INPG
ENSIEG
Rue de la houille blanche – BP46
38402 SAINT MARTIN D'HERES CEDEX