



**Contribution à la planification d'expériences, à
l'estimation et au diagnostic actif de systèmes
dynamiques non linéaires : application au domaine
aéronautique**

Qiaochu Li

► **To cite this version:**

Qiaochu Li. Contribution à la planification d'expériences, à l'estimation et au diagnostic actif de systèmes dynamiques non linéaires : application au domaine aéronautique. Autre. Université de Technologie de Compiègne, 2015. Français. NNT : 2015COMP2231 . tel-01299578

HAL Id: tel-01299578

<https://theses.hal.science/tel-01299578>

Submitted on 7 Apr 2016

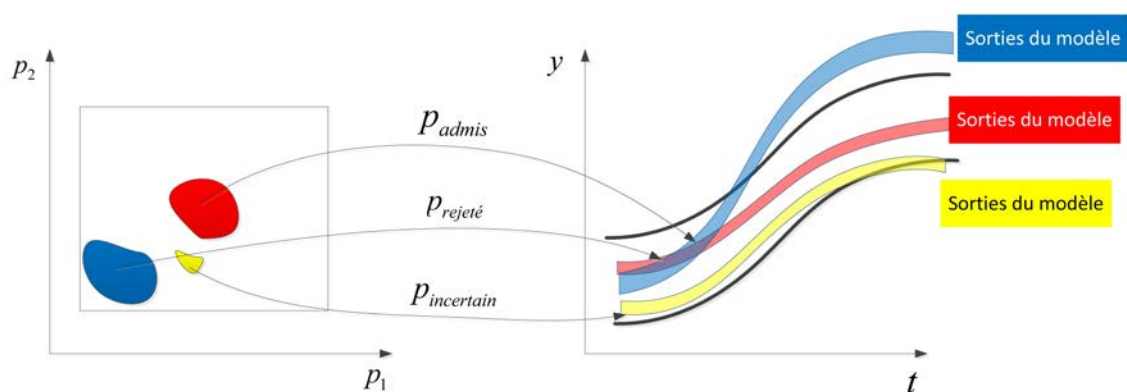
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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.

Par **Qiaochu LI**

Contribution à la planification d'expériences, à l'estimation et au diagnostic actif de systèmes dynamiques non linéaires : application au domaine aéronautique

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 10 novembre 2015

Spécialité : Technologies de l'Information et des Systèmes –
Mathématiques appliquées et Problèmes inverses

D2231

Université de Technologie de Compiègne

Technologies de l'Information et des Systèmes

Ph.D dissertation

Defended by

Qiaochu LI

Contribution à la planification d'expériences, à l'estimation et au diagnostic actif de systèmes dynamiques non linéaires. Application au domaine aéronautique

soutenance soutenue le 10 Novembre 2015 :

Membre de jury :

<i>Rapporteurs :</i>	Luc JAULIN	- Professeur, Lab-STICC
	Michel KIEFFER	- Professeur, L2S
<i>Membres du jury :</i>	Abdellatif EL BADIA	- Professeur, LMAC
	Floriane COLLIN	- Maître de conférence, CRAN
	Nathalie REVOL	- Maître de conférence, LIP
<i>Directrices :</i>	Zohra CHERFI	- Professeur, Laboratoire Roberval
	Lilianne DENIS-VIDAL	- Maître de conférence, HDR, LMAC
	Carine JAUBERTHIE	- Maître de conférence, LAAS

Declaration of Authorship

I, declare that this thesis titled, ‘Contribution à la planification d’expériences, à l’estimation et au diagnostic actif de système dynamiques non linéaire. Application au domaine aéronautique’ and the work presented in it are my own. I confirm that :

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed : 

Date : 20 Novembre 2015

“Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism.”

Dave Barry

Remerciement

Cette étude a été réalisée au sein de l'équipe d'accueil Laboratoire Mathématique Appliquée de Compiègne (LMAC), dirigée par Madame Lilianne DENIS-VIDAL. Je tiens à la remercier chaleureusement de m'avoir proposé et de superviser ce sujet, pour son accueil et les moyens donnés pour réaliser ce travail dans les bonnes conditions. Madame Carine JAUBERTHIE a également accepté la direction scientifique de cette thèse, et ce malgré ses nombreuses responsabilités. Madame Zohra CHERFI m'a continuellement donné des conseils précieux, cette expérience sera inoubliable dans toute ma vie. Je ne sais comment les remercier pour leurs disponibilités, leur patience, leurs conseils et tout ce qu'elles ont pu m'apprendre au cours de ces trois années.

Je remercie vivement Messieurs Luc JAULIN et Michel Kieffer qui m'ont fait l'honneur d'accepter de Rapporter sur cette thèse et je tiens à leur exprimer ma profonde gratitude. Je remercie aussi messieur Abdellatif EL BADIA et mesdames Nathalie REVOL et Floriane COLLIN d'avoir accepté d'être membre du jury.

Cette thèse ne se serait jamais réalisée sans l'aide de l'équipe du LMAC pour leur soutien qui n'a jamais fait défaut. Merci particulièrement à Monsieur Nicolaos LIMINIOS pour les solutions de tous les soucis administratifs ; merci à Batoul, Chrysanthi, Shousheng, pour leur amitié ; Je tiens à remercier toutes les personnes du service de l'UTC pour leur accueil et leur sympathie. Je voudrais aussi remercier tous mes amis. Témoins de tant de doutes, sans vous, ce mémoire n'aurait jamais vu le jour.

En dernier lieu, je ne sais comment exprimer toute ma gratitude à ma femme Jing XU, qui soutiendra sa thèse trois mois plus tard que moi ; mes remerciements envers mes parents, pour leurs encouragements et leur soutien sans condition ; un remerciement est également adressé aux thésards de l'UTC : Batoul, Souade, Hayat, Irène, Diana, Taozui, Bihao, Xingxing, Bi'ao, liste non exhaustive. pour leur soutien sans condition, pour tous les partages entre nous.

On dit souvent entre les doctorants, que la rédaction d'une thèse est un cauchemar : il y a des moments difficiles, il y a des moment sceptiques. L'essentiel est la façon de point de vue sur la rédaction. Sachant que gagner de l'argent, c'est plus dur que rédiger une thèse.

Abstract

In this work, we will study the uncertainty integration problem in a bounded error context for the dynamic systems, whose input and the initial state have to be optimized so that some other operation could be more easily and better obtained.

This work is consisted of 6 chapters : the chapter 1 is an introduction to the general subject which we will discuss about. The chapter 2 represents the basic tools of interval analysis.

The chapter 3 is dedicated to state estimation and parameter estimation. We explain at the first, how to solve the ordinary differential equation using interval analysis, which will be the basic tool for the state estimation problem given the initial condition of studied systems. On the other side, we will look into the parameter estimation problem using interval analysis too. Based on a simple hypothesis over the uncertain variable, we calculate the system's parameter in a bounded error form, considering the operation of intervals as the operation of sets. Guaranteed results are the advantage of interval analysis, but the big time consumption is still a problem for its popularization in many non linear estimation field. We present our founding techniques to accelerate this time consuming processes, which are called contractor in constraint propagation field. At the end of this chapter, different examples will be the test proof for our proposed methods.

Chapter 4 presents the searching for optimal input in the context of interval analysis, which is an original approach. We have constructed several new criteria allow such searching. Some of them are intuitive, the other need a theoretical proof. These criteria have been used for the search of optimal initial states and le better parameter estimation results. The comparisons are done by using multiple applications and the efficiency is proved by evidence.

In chapter 5, we applied the approaches proposed above in diagnosis by state estimation and parameter estimation. We have developed a complete procedure for the diagnosis. The optimal input design has been reconsidered in an active diagnosis context. Both state and parameter estimation are implemented using an aeronautical application in literature.

The last chapter gives a brief summary over the realized subject, some further research directions are given in the perspective section.

All the algorithms are written in C/C++ on a Linux based operation system.

Résumé

Dans ce travail de thèse, nous nous focalisons sur le problème de l'intégration d'incertitude à erreurs bornées pour les systèmes dynamiques, dont les entrées et les états initiaux doivent être optimaux afin de réaliser certaines fonctionnalités.

Le document comporte 5 chapitres : le premier est une introduction présentant le panorama du travail. Le deuxième chapitre présente les outils de base de l'analyse par intervalle.

Le chapitre 3 est dédié à l'estimation d'états et de paramètres. Nous décrivons d'abord une procédure pour résoudre un système d'équations différentielles ordinaires avec l'aide de cet outil. Ainsi, une estimation des états à partir des conditions initiales peut être faite. Les systèmes différentiels considérés dépendent de paramètres qui doivent être estimés. Ce problème inverse pourra être résolu via l'inversion ensembliste. L'approche par intervalle est une procédure déterministe naturelle sans incertitude, tous les résultats obtenus sont garantis. Néanmoins, cette approche n'est pas toujours efficace, ceci est dû au fait que certaines opérations ensemblistes conduisent à des temps de calcul important. Nous présentons quelques techniques, par cela, nous nous plaçons dans un contexte à erreurs bornées permettant d'accélérer cette procédure. Celles-ci utilisent des contracteurs ciblés qui permettent ainsi une réduction de ce temps. Ces algorithmes ont été testés et ont montré leur efficacité sur plusieurs applications : des modèles pharmacocinétiques et un modèle du vol longitudinal d'avion en atmosphère au repos.

Le chapitre 4 présente la recherche d'entrées optimales dans le cadre analyse par intervalle, ce qui est une approche originale. Nous avons construit plusieurs critères nouveaux permettant cette recherche. Certains sont intuitifs, d'autres ont nécessité un développement théorique. Ces critères ont été utilisés pour la recherche d'états initiaux optimaux. Des comparaisons ont été faites sur plusieurs applications et l'efficacité de certains critères a été mise en évidence.

Dans le chapitre 5, nous appliquons les approches présentées précédemment au diagnostic via l'estimation de paramètres. Nous avons développé un processus complet pour le diagnostic et aussi formulé un processus pour le diagnostic actif avec une application en aéronautique.

Le dernier chapitre résume les travaux réalisés dans cette thèse et essaye de donner des perspectives à la recherche.

Les algorithmes proposés dans ce travail ont été développés en C++ et utilisent l'environnement du calcul ensembliste.

Table des matières

Declaration of Authorship	i
Remerciements	v
Abstract	vii
Résumé	ix
Contents	xiii
List of Figures	xvi
List of Tables	xvii
Acronymes	xix
1 Introduction	1
1.1 Introduction générale	1
1.2 Modèle mathématique étudié	2
1.2.1 L'erreur statistique et l'erreur ensembliste	3
1.2.2 Equations de sensibilité	3
1.3 L'estimation de paramètres à erreurs inconnues mais bornées	5
1.4 Le diagnostic et le diagnostic actif	6
2 Outil d'estimation dans le contexte ensembliste	9
2.1 Introduction au contexte ensembliste	9
2.1.1 Calcul avec des ensembles	10
2.1.2 Représentation d'une variable incertaine	12
2.2 Introduction à l'analyse par intervalle	12
2.2.1 Pavé et sous pavage	14
2.2.2 Arithmétique des intervalles	15
2.2.3 Propriété algébrique	17
2.2.3.1 Propriété d'associativité et de distributivité	17
2.2.3.2 Propriétés de multiplication dans certains cas	18
2.2.3.3 Propriété d'inclusion avec les matrices intervalles	19
2.2.4 Fonction d'inclusion	19
2.2.5 Pessimisme de surestimation	25
2.2.5.1 Le problème d'indépendance	25
2.2.5.2 Effet d'enveloppement	26
2.2.6 Arithmétique affine	26
2.2.7 Comparaison entre arithmétique affine et analyse par intervalle	27
2.2.8 Librairie d'intervalle	28
2.2.9 Image réciproque d'un ensemble	29

2.2.10	Image directe d'un ensemble	31
3	Estimation garantie et application aux modèles académiques	33
3.1	Estimation dans le contexte ensembliste	35
3.1.1	Estimation d'état	36
3.1.2	Estimation de paramètres	38
3.1.3	Contracteur	41
3.2	Intégration numérique garantie basée sur le développement en série de Taylor . .	43
3.2.1	Existence et unicité de la solution	46
3.2.2	Recherche d'une solution a priori	47
3.2.3	Réduction du pessimiste sur la solution	48
3.2.4	Effet d'enveloppement	49
3.2.4.1	Méthode de <i>Extended Mean Value</i>	50
3.2.5	Solveur disponible pour l'intégration garantie	52
3.3	Intégration numérique garantie basée sur le théorème de Müller	53
3.4	Application dans le domaine pharmaceutique	54
3.4.1	Estimation d'état	57
3.4.2	Estimation paramétrique	58
3.5	Application dans le domaine pharmacocinétique	64
3.6	Application dans le domaine aéronautique	67
3.6.1	Modèle du vol longitudinale d'un avion en atmosphère au repos	68
3.6.2	Estimation d'état du modèle	69
3.6.3	Estimation de paramètres du modèle	71
4	Planification d'expérience pour les systèmes non-linéaires	79
4.1	Principe de la planification d'expériences	80
4.1.1	Critère de A-optimalité-intervalle	82
4.1.2	Critère M-optimalité-intervalle	87
4.1.3	Critère D-optimalité-intervalle	89
4.1.4	Critère Migmag-optimalité-intervalle	90
4.2	Application au modèle d'avion	92
4.2.1	Effet élémentaire des optimalités différentes	94
4.2.1.1	Minimisation du nombre de point via la Trace optimalité	95
4.2.2	Résultat de l'estimation de paramètres	96
4.3	Conclusion	102
5	Diagnostic par analyse par intervalles	103
5.1	Diagnostic dans le contexte à erreurs bornées	103
5.1.1	Approche par estimation d'état	105
5.1.2	Approche par estimation de paramètres	106
5.2	Diagnostic pour le modèle d'avion	107
5.2.1	Diagnostic par estimation d'état	107
5.2.2	Diagnostic par estimation de paramètres	108
5.3	Diagnostic actif pour le modèle d'avion	109
5.3.1	Cas de faute simple	110

6 Conclusion	119
6.1 Conclusion	119
6.2 Perspective	120
A Annexe	121
A.1 Annexe pour l'algorithme de la recherche de l'entrée	121
Bibliographie	123

Table des figures

2.1	La longueur et le point milieu d'un pavé $X = (X_1, X_2)$	15
2.2	Pavé pour la borne inférieure	23
2.3	Pavé pour la borne supérieure	23
2.4	Vecteur d'intervalle de dimension 1, 2 et 3	26
3.1	L'estimation d'état : la phase de prédiction et la phase de correction	36
3.2	L'effet d'enveloppement	50
3.3	L'effet d'enveloppement calculé par la Extended Mean Value	52
3.4	Modèle de deux compartiments	54
3.5	Evolution de l'intervalle contenant x_1 fourni par VNODE-LP, VSPODE, IOLA-VABE, EMV	55
3.6	Evolution de l'intervalle contenant x_2 fourni par VNODE-LP, VSPODE, IOLA-VABE, EMV	56
3.7	Estimation d'état pour le modèle pharmaceutique	57
3.8	Estimation d'état pour le modèle pharmaceutique	57
3.9	Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite), $\varepsilon = 0.01$	58
3.10	Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite), $\varepsilon = 0.005$. . .	59
3.11	Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite), $\varepsilon = 0.001$. . .	59
3.12	Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite), $\varepsilon = 0.0005$. . .	59
3.13	Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite).	60
3.14	Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite) $\varepsilon = 0.01$	62
3.15	Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite) $\varepsilon = 0.005$	62
3.16	Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite) $\varepsilon = 0.001$	62
3.17	Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite) $\varepsilon = 0.0005$	63
3.18	Estimation d'état à partir de $p_2 = 0.35$ et $p_3 = 0.15$	63
3.19	Estimation de paramètre sur p_1 et p_2	65
3.20	Estimation de paramètre sur p_2 et p_3	65
3.21	Incidence et vecteur vitesse d'un avion.	68
3.22	L'estimation d'état du modèle d'avion.	70
3.23	Les différentes entrées pour le modèle d'avion.	72
3.24	La représentation des domaines admissibles en 3D.	72
3.25	La représentation des domaines incertains en 3D.	73
3.26	La représentation des domaines rejetés en 3D.	73
3.27	La représentation des domaines admissibles en 3D.	74
3.28	La représentation des domaines incertains en 3D.	74
3.29	La représentation des domaines rejetés en 3D.	75
3.30	Entrée pour le modèle d'avion dite optimale.	75
3.31	La représentation des domaines admissibles en 3D.	76
3.32	La représentation des domaines incertains en 3D.	76
3.33	La représentation des domaines rejetables en 3D.	77

4.1	Deux entrées optimales obtenues avec les critères A et D optimalité	94
4.2	La relation entre nombre de point par échantillonnage et la valeur μ^*	96
4.3	La représentation des domaines admissibles en 3D.	96
4.4	La représentation des domaines incertains en 3D.	97
4.5	La représentation des domaines rejetés en 3D.	97
4.6	La représentation des domaines admissibles en 3D.	98
4.7	La représentation des domaines incertains en 3D.	98
4.8	La représentation des domaines rejetés en 3D.	99
4.9	Entrée optimale via le critère Migmag.	100
4.10	Boîte acceptable et boîte incertaine avec l'entrée \hat{u}' et \hat{u} (ε_1).	101
4.11	Estimation de paramètres par Monte Carlo avec l'entrée \hat{u}' et \hat{u}	101
5.1	Approche du diagnostic par redondance analytique	105
5.2	L'estimation d'état du modèle d'avion	108
5.3	La représentation des domaines admissibles en 3D	110
5.4	La représentation des domaines incertains en 3D.	111
5.5	La représentation des domaines rejetés en 3D.	111
5.6	La représentation des domaines admissibles en 3D.	112
5.7	La représentation des domaines incertains en 3D.	112
5.8	La représentation des domaines rejetés en 3D.	113
5.9	La représentation des domaines admissibles en 3D.	113
5.10	La représentation des domaines incertains en 3D.	114
5.11	La représentation des domaines rejetés en 3D.	114
5.12	La représentation des domaines admissibles en 3D.	115
5.13	La représentation des domaines incertains en 3D.	115
5.14	La représentation des domaines rejetés en 3D.	116
5.15	La représentation des domaines admissibles en 3D.	116
5.16	La représentation des domaines incertains en 3D.	117
5.17	La représentation des domaines rejetés en 3D.	117

Liste des tableaux

2.1	Différents raffinements	21
3.1	Temps d'exécution du programme (s).	64
3.2	Volume de $p_1p_2p_3$	66
3.3	Volume de $p_1p_2p_3$	67
3.4	Volume de $p_1p_2p_3$	67
3.5	Volume de $p_1p_2p_3$	77
3.6	Volume de $p_2p_3p_4$	77
3.7	Produit de $p_1p_2p_3p_4$	77
4.1	Valeur optimale des critères avec différents horizons.	93
4.2	Type d'entrée obtenue.	94
4.3	Volume de $p_1p_2p_3p_4$	99
4.4	Valeur de la fonction coût $j(\Xi)$	99
4.5	Résultat de l'estimation des paramètres en prenant ε_1	100

Acronymes

EMI	Ensemble de Mesure Incertaine
EEC	Ensemble des Eléments Consistants
AA	Arithmétique Affine
AI	Analyse par Intervalles
INTLAB	INTerval LABoratory
FILIB++	Fast Interval LIBrary
PROFIL/BIAS	Programmer's Runtime Optimized Fast Interval Library / Basic Interval Arithmetic Subroutines
LAPACK/BLAS	Linear Algebra PACKage / Basic Linear Algebra Subprograms
VNODE	Validated numerical ordinary differential equation
VNODE-LP	Validated numerical ordinary differential equation through Iterate Programming
EMV	Extended Mean Value
VSPODE	Validated Solutions of Initial Value Problems for Ordinary Differential Equations
IOVALABE	iSAT-ODE Layer Around VNODE-LP and Bracketing Enclosures
SIVIA	Set Inversion Via Interval Analysis
LMI	Linear Matrix Inequality
EE	Elementary Effet

Introduction

Contents

1.1	Introduction générale	1
1.2	Modèle mathématique étudié	2
1.2.1	L'erreur statistique et l'erreur ensembliste	3
1.2.2	Equations de sensibilité	3
1.3	L'estimation de paramètres à erreurs inconnues mais bornées	5
1.4	Le diagnostic et le diagnostic actif	6

1.1 Introduction générale

Pour changer le monde qui nous entoure, la première chose est de le comprendre. Une façon d'y parvenir peut se réaliser par le biais de la modélisation. Le monde physique peut être modélisé mathématiquement par certaines formes d'équations dont les variables sont inconnues mais dépendent de mesures réelles. Ces mesures sont souvent obtenues par un processus ou une observation, ce qui introduit souvent des perturbations sur les signaux obtenus. Comme dans le domaine de l'automatique la plupart des machines sont contrôlées par certaines méthodes numériques qui emploient l'estimation et l'identification, ces branches ont largement évolué. L'estimation parle du problème d'inverse, qui évalue des variables inconnues dépendant de mesures données. Les données sont souvent bruitées et c'est pourquoi il est nécessaire d'évaluer de combien et comment les variables estimées influencent le problème considéré.

Dans le cas classique stochastique, les incertitudes sont supposées suivre une certaine loi probabiliste. Mais dans certains cas, la probabilité du phénomène physique est susceptible d'être incorrecte. Par exemple, le processus mesuré par des données réelles peut être complexe (non linéaire, temps variant, etc.) alors que des modèles simplifiés sont souvent utilisés dans le processus d'estimation. Ensuite, le résidu du modèle simplifié a une partie d'erreur déterministe. Cela peut être des déviations sur leurs sorties, des dérives du capteur, etc. Si nous traitons ces erreurs comme des variables aléatoires distribuées suivant une loi de distribution, alors des résultats insatisfaisants pourront être obtenus si la loi est mal choisie.

Parmi les approches alternatives, l'analyse par intervalles a attiré notre attention. Elle considère toutes les variables comme des inconnues mais bornées par une borne supérieure et une borne inférieure, ce qui est une bonne interprétation pour des phénomènes aléatoires de loi inconnue. Avec le perfectionnement des capteurs, les mesures sont faciles

à acquérir et en plus, leur fréquence est de plus en plus élevée. Malheureusement les données sont souvent sur échantillonnées et souffrent des erreurs structurelles quand la configuration du capteur n'est pas totalement satisfaite. Au lieu de prendre une valeur potentiellement bruitée, l'approche par intervalle propose d'utiliser une borne supérieure et une borne inférieure pour encadrer ces erreurs.

Nous avons un élément \hat{x} d'un problème (par exemple des paramètres d'un système dynamique ou un instant particulier durant un processus) et nous souhaitons évaluer une fonction $f(\hat{x})$ (un système dynamique ou un processus hybride) de cet élément. Nous supposons que la valeur de l'élément est incertaine, mais des informations a priori sont disponibles. Nous supposons que cet élément appartient à l'ensemble X dont les éléments sont des solutions possibles et qu'une fonction $f(\hat{x})$ est mesurée. Nous supposons de plus que les mesures exactes ne sont pas disponibles mais que les mesures y sont perturbées par une erreur e dont les bornes \bar{e} (supérieure) et \underline{e} (inférieure) sont connues. Le problème de l'estimation consiste à trouver un estimateur F fournissant une approximation $F(x) \supset f(\hat{x})$ en utilisant les données y et évaluant les mesures de cette approximation.

Dans l'analyse par intervalle, nous nous intéressons à la détermination du choix d'une entrée contrôlée dans le cas de systèmes non linéaires. Les algorithmes par intervalles proposés dédiés à de tels systèmes sont détaillés dans les chapitres qui suivent.

Nous nous orientons à tester des entrées différentes via des critères basés sur la sensibilité et aussi d'autres critères qui ont aussi été évalués. Le but est de mettre au point la théorie qui permet de formuler proprement notre démarche dans le cadre ensembliste. Il s'agit d'un travail original, destiné à être appliqué au cas réel comme le diagnostic actif.

En ce qui concerne le diagnostic actif, nous avons parcouru les travaux proposés dans la littérature. Notre but est de trouver des entrées permettant de raffiner le diagnostic de faute lorsque celui est ambigu.

1.2 Modèle mathématique étudié

Soit une fonction $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^m$ continûment différentiable de forme suivante :

$$y = f(t, p), \quad (1.1)$$

pour tout $t \in [0, T] \in \mathbb{R}$ et $p \in P \subset \mathbb{R}^n$. Alors si la fonction f dotée d'une expression explicite, cette fonction est dite exacte. Supposons $X \subset \mathbb{R}^n$ et $Y \subset \mathbb{R}^m$, alors un modèle exact a la forme suivante :

$$\begin{cases} x(t, p) = h(t, p), \\ y(t, p) = g(x(t, p)), \end{cases} \quad (1.2)$$

où $h : [0, T] \times P \rightarrow X$ et $g : X \rightarrow Y$ sont des fonctions continûment différentiables .

Dans la littérature, souvent le modèle mathématique est décrit par un système d'équations non linéaires différentielles ordinaires. Les modèles sont donnés sous la forme suivante :

$$\begin{cases} \dot{x}(t, p) = h(t, p, x(t, p)) & x(0, p) = x_0, \\ y(t, p) = g(x(t, p)), \end{cases} \quad (1.3)$$

où $h : [0, T] \times P \times X \rightarrow X$ et $g : X \rightarrow Y$ sont des fonctions. Souvent, il y aura des erreurs intrinsèques quand nous mesurons les sorties et il est peu probable que le modèle coïncide avec les mesures réelles. Afin d'introduire ces incertitudes, nous utilisons un modèle mathématique qui évalue les différences entre les sorties du modèle mathématique et du système réel.

1.2.1 L'erreur statistique et l'erreur ensembliste

Le modèle statistique est défini par

$$y(t_j) = y^m(t_j, p_0) + e_j, \quad (1.4)$$

où e_j est une variable aléatoire qui représente les erreurs sur le processus de mesure et p_0 représente le "vrai" paramètre du modèle. Nous supposons que pour tous $j = 1, \dots, N$, la variable aléatoire e_j a une distribution normale de moyenne nulle, c'est-à-dire $E(e_j) = 0$. Dans le cas scalaire, nous supposons que la variance d'erreur est $\text{var}(e_j) = \sigma^2$ pour tous $j = 1, \dots, N$ et le cas vectoriel, nous supposons que $\text{var}(e_j) = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_m^2)$ pour tous $j = 1, \dots, N$. Comme e_j est une variable aléatoire, $y(t_j)$ est aussi une variable aléatoire.

La conception ensembliste peut être appliquée dans n'importe quelle condition incertaine, quelque soit le type de la source d'erreur qui va affecter les sorties du système dynamique. Supposons que e_j suive une distribution gaussienne de moyenne μ et de variance σ^2 , alors cet ensemble peut être décrit par un intervalle :

$$[E] = \mu + 3\sigma[-1, 1] = [\mu - 3\sigma, \mu + 3\sigma], \quad (1.5)$$

qui représente 99.7% de sûreté pour la variable e avec une distribution aléatoire.

Étant donné un modèle mathématique et un ensemble de mesures d'expériences, l'étape suivante consiste à trouver un paramètre qui "adapte" les sorties du modèle mathématique aux mesures à partir de l'expérience. Ce processus est connu comme étant le problème de l'estimation de paramètres et les hypothèses sur la distribution de la variable représentant le bruit qui influence la résolution du problème. Dans le contexte stochastique, ce problème a été traité dans le cas ponctuel. Avant d'entrer dans le contexte de l'analyse par intervalles, nous présentons un court résumé concernant les équations de sensibilité définies par le modèle. Comme nous le verrons plus tard, la sensibilité joue un rôle important dans l'estimation de paramètres, la planification d'expérience et dans le diagnostic actif.

1.2.2 Equations de sensibilité

La sensibilité donne des informations sur les impacts des variations des paramètres sur la sortie. L'obtention des sensibilités d'un modèle précis est un élément important dans l'estimation de paramètres et dans la planification d'expérience.

Définition 1. Les sensibilités des sorties du modèle (1.2) et (1.3) par rapport aux paramètres du modèle sont définies comme :

$$\begin{aligned}
s_{p_1}(t, p) &= \left[\frac{\partial y_1(t, p)}{\partial p_1}, \frac{\partial y_2(t, p)}{\partial p_1}, \dots, \frac{\partial y_m(t, p)}{\partial p_1} \right], \\
s_{p_2}(t, p) &= \left[\frac{\partial y_1(t, p)}{\partial p_2}, \frac{\partial y_2(t, p)}{\partial p_2}, \dots, \frac{\partial y_m(t, p)}{\partial p_2} \right], \\
&\vdots \\
s_{p_{n_p}}(t, p) &= \left[\frac{\partial y_1(t, p)}{\partial p_{n_p}}, \frac{\partial y_2(t, p)}{\partial p_{n_p}}, \dots, \frac{\partial y_m(t, p)}{\partial p_{n_p}} \right].
\end{aligned} \tag{1.6}$$

Pour des systèmes décrits par des modèles exacts, le calcul des sensibilité est direct, nous résolvons le modèle accompagné des équations de sensibilités sous forme différentielle. Par souci de simplification des notations, nous considérons que g est une fonction linéaire réduite à l'identité. Ce qui est équivalent à $y(t, p) = x(t, p)$. Par dérivation, nous obtenons :

$$\dot{x}(t, p) = h(t, p, x(t, p)), \tag{1.7}$$

Notons que $\dot{x}_{p_j}(t, p) = \frac{\partial \dot{x}}{\partial p_j}(t, p)$ et $x_{p_j}(t, p) = \frac{\partial x}{\partial p_j}(t, p)$, en résultent les sensibilités d'états par rapport aux paramètres :

$$\begin{aligned}
\dot{x}_{p_1}(t, p) &= h_x(t, p, x(t, p))x_{p_1}(t, p) + h_{p_1}(t, p, x(t, p)), \\
\dot{x}_{p_2}(t, p) &= h_x(t, p, x(t, p))x_{p_2}(t, p) + h_{p_2}(t, p, x(t, p)), \\
&\vdots \\
\dot{x}_{p_{n_p}}(t, p) &= h_x(t, p, x(t, p))x_{p_{n_p}}(t, p) + h_{p_{n_p}}(t, p, x(t, p)).
\end{aligned} \tag{1.8}$$

Pour la condition initiale $x(0, p)$ par rapport à p , nous définissons :

$$x_p(0, p) = \left[\frac{\partial x(0, p)^T}{\partial p_1}, \frac{\partial x(0, p)^T}{\partial p_2}, \dots, \frac{\partial x(0, p)^T}{\partial p_{n_p}} \right]^T. \tag{1.9}$$

Ensuite, il s'ensuit que les sensibilités des états peuvent être résolues via le système :

$$\begin{aligned}
\dot{x}(t, p) &= h(t, p, x(t, p)), \\
\dot{x}_{p_1}(t, p) &= h_x(t, p, x(t, p))x_{p_1}(t, p) + h_{p_1}(t, p, x(t, p)), \\
\dot{x}_{p_2}(t, p) &= h_x(t, p, x(t, p))x_{p_2}(t, p) + h_{p_2}(t, p, x(t, p)), \\
&\vdots \\
\dot{x}_{p_{n_p}}(t, p) &= h_x(t, p, x(t, p))x_{p_{n_p}}(t, p) + h_{p_{n_p}}(t, p, x(t, p)) \\
[x(0, p)^T, \quad x_p(0, p)^T]^T &= [x_0^T, x_{0,p}^T]^T.
\end{aligned} \tag{1.10}$$

où $h_x(t, p, x(t, p))$ représente la matrice jacobienne par rapport à x , $h_{p_i}(t, p, x(t, p))$ représente la dérivé de h par rapport à p_i , pour $i = (1, \dots, n_p)$.

Finalement, nous obtenons les sensibilités du modèle :

$$\begin{aligned}
s_{p_1}(t, p) &= y_{p_1}(t, p) = h_x(t, p, x(t, p))x_{p_1}(t, p) + h_{p_1}(t, p, x(t, p)), \\
s_{p_2}(t, p) &= y_{p_2}(t, p) = h_x(t, p, x(t, p))x_{p_2}(t, p) + h_{p_2}(t, p, x(t, p)), \\
&\vdots \\
s_{p_{n_p}}(t, p) &= y_{p_{n_p}}(t, p) = h_x(t, p, x(t, p))x_{p_{n_p}}(t, p) + h_{p_{n_p}}(t, p, x(t, p)).
\end{aligned} \tag{1.11}$$

Si la dimension des états n est grande, la solution des équations de sensibilité peut être lourde, nous pouvons regrouper les équations de sensibilité avec les équations d'état qui résultent de plusieurs sous-systèmes afin d'alléger les calculs.

1.3 L'estimation de paramètres à erreurs inconnues mais bornées

La stratégie que nous utilisons pour estimer les paramètres inconnus d'un modèle est basée sur les hypothèses qui ont été faites sur les e_j , variables aléatoires qui représentent les erreurs dans le modèle statistique et aussi la quantité des mesures disponibles. Deux méthodes peuvent être utilisées pour l'estimation de paramètres. Chaque méthode demande des hypothèses statistiques standards sur le comportement de e_j .

- Méthode des moindres carrés.
- Méthode de maximum de vraisemblance.

Celles ci sont déterminées par les connaissances de leurs lois de distribution [Casella 2002], [Banks 2009]. Si l'on connaît approximativement la répartition de e_j , alors nous pouvons mettre en œuvre l'estimateur du maximum de vraisemblance afin d'utiliser les informations que l'on connaît sur la distribution de e_j dans l'estimation de paramètre. L'estimation des moindres carrés peut être utilisée quand la loi de distribution des e_j est inconnue [Burns 2007].

Dans un contexte ensembliste, l'estimation de paramètres est dite à erreurs bornées, toutes les variables incluent toutes les valeurs possibles sur des quantités incertaines [Jaulin 2001a] [Raïssi 2004b] [Kieffer 2011]. Avec les outils intervalles, les solutions peuvent être aussi obtenues numériquement et correctement en ne tenant compte que des erreurs d'arrondis sur l'ordinateur. Donc, quelle que soit le nombre de mesures, l'approche ensembliste permet d'obtenir des résultats garantis, à la fois numériquement et pratiquement. Ce qui est différent des solutions retenues par l'approche stochastique, nous avons des intervalles de confiance sur les résultats obtenus.

L'estimation de paramètres à erreur bornée a pour but de trouver un ensemble qui décrit les paramètres potentiellement corrects permettant de le projeter via la fonction du système à l'intérieur des données bornées ou bien de trouver un ensemble qui minimise une fonction coût choisie, par exemple, la norme de la différence entre les données mesurées et les données calculées [Jaulin 2001b]. La minimisation de cette fonction coût conduit à une seule solution : le vecteur de paramètres sur le point minimal. D'ailleurs, cette approche utilise la propriété asymptotique de l'estimateur de vraisemblance, ce qui n'est pas approprié quand le nombre de mesures est insuffisant. Une approche plus intéressante est d'utiliser l'estimation via le calcul ensembliste. L'avantage de cette méthode est qu'elle ne nécessite pas d'hypothèses sur le nombre de données, mais impose que les erreurs de mesures soient bornées. Les résultats que nous obtenons donnent une vision de la répartition des paramètres, ce qui nous permet d'appréhender plus d'information : par exemple l'état de fonctionnement, pronostiquer les fautes, etc [Pons 2008]. Comme pour chaque donnée, nous devons faire une inversion ensembliste, ce qui nécessite plusieurs opérations récursive, le temps de calcul est alors plus coûteux que dans le cas de la

méthode statistique.

Une question qui se pose souvent concerne le temps d'échantillonnage permettant d'obtenir les mesures donnant la meilleure estimation de paramètres : Si nous avons N échantillons de mesures, comment choisir les instants pour les M échantillons suivants afin d'estimer les paramètres ? C'est une question cruciale car deux ensembles de données contenant M échantillons, mais pris à des instants différents peuvent conduire à une estimation de paramètres de qualité différente. La réponse est d'utiliser la planification d'expérience. Avec cette approche, nous souhaitons planifier des expériences qui soient efficaces pour estimer au mieux les paramètres inconnus à partir du modèle. En plus elle nous permet d'identifier les conditions optimales pour les modèles correspondants.

Pour concevoir une planification d'expérience, nous devons faire quelques hypothèses sur les paramètres à estimer. Cette information peut venir d'expériences préalables ou à partir d'une étude sur le système modélisé. Peu importe d'où vient cette information, la planification d'expérience commence par la donnée d'un ensemble borné dans lequel se trouveront avec certitude les paramètres \tilde{p} . La planification d'expériences est donc liée à cet ensemble préfixé (a priori connu) de paramètres. Pour cette raison, il est crucial de bien choisir cet ensemble de paramètres au départ.

1.4 Le diagnostic et le diagnostic actif

Les systèmes automatiques deviennent de plus en plus complexes et les algorithmes de contrôle aussi. D'autre part l'efficacité, la rentabilité, la sûreté sont des points incontournables. Cela demande des développements sur le diagnostic sur ces systèmes automatiques afin de gérer la fiabilité sur des systèmes de sûretés critiques [Chen 2012]. Des systèmes de diagnostic sont alors proposés afin de détecter les fautes et diagnostiquer leur importance sur l'objet étudié. Un tel système consiste de trois missions : la détection, l'isolation et l'identification de fautes. Dans cette thèse, nous parlerons de la méthodologie du diagnostic à base de modèles analytiques [Isermann 2005].

Avec cette approche, le principe fondamental du diagnostic s'énonce ainsi :

"toute procédure de diagnostic procède nécessairement de tests de cohérence entre des signaux issus d'observation et des modèles de comportement".

Néanmoins, la vérification de cette cohérence nous donne un taux de fausses alarmes élevé en tenant compte de l'erreur incertaine importante, par contre, une seule faute résulte de fautes multiples à la phase de détection de faute, ce qui rend difficile d'isoler des fautes différentes. D'autre part, cette vérification de cohérence nécessite une redondance matérielle ou analytique qui restent le seul secours pour compenser tous ces inconvénients.

Le diagnostic actif stimule le système par une entrée appelée excitation afin de décider si dans les comportements du système apparaissent des dysfonctionnements ou pas. Il peut être utile dans les situations suivantes :

- Génération des signaux de test lors de la mise en service d'un système pour en vérifier le bon fonctionnement.
- Détection des dysfonctionnements plus rapidement que dans les conditions normales d'un système.

- Détection des dysfonctionnements cachés résultant de la régulation et conduisant à ne pas percevoir le dysfonctionnement car les sorties sont indistinguables.

Le but du diagnostic est de trouver une séquence d'entrées de telle sorte que les sorties calculées avec les systèmes dysfonctionnels et les systèmes fonctionnels deviennent différentes. Nous supposons que les sorties du système sont disponibles à chaque instant t . Une sortie du système est notée d'une manière générale Y et $Y = (y(t_1), \dots, y(t_N))^T$. Nous noterons $T = t_{n_1}, \dots, t_{n_k}$ un ensemble d'instant de mesures, $Y_i(T) = (y_i(t_{n_1}), \dots, y_i(t_{n_k}))^T$ la sortie qui subit la faute f_i .

Pour que la faute f_i soit détectable, il suffit qu'il existe un instant $t_{n_{ji}}$ telle que la différence entre $y_i(t_{n_{ji}})$ et le mode normal soit significative [Isermann 2005] :

$$|y_0(t_{n_{ji}}) - y_i(t_{n_{ji}})| \geq d \text{ pour tout } i \in \{1, \dots, n\}, \quad (1.12)$$

où $d > 0$ et n représente le nombre de fautes et $t_{n_{ji}}$ représente un élément de t_{n_1}, \dots, t_{n_k} qui correspond à l'ensemble T pour la faute f_i . $y_0(t_{n_{ji}})$ représente le système en fonctionnement normal, c'est-à-dire sans faute.

La détection de fautes et l'identification de fautes peuvent être faites par le biais de surveiller peu de mesures, le processus dynamique, le modèle de signal et l'estimation de paramètres afin de générer un symptôme [Isermann 1993]. Ce symptôme sera ensuite évalué par une procédure de diagnostic, qui est basée sur des connaissances a priori. Le taux de fausses "alarmes" est un effet inévitable dans le cadre de diagnostique probabiliste, pour couvrir plus largement le cas défaillant, de nombreuses expériences doivent être faites pour avoir une valeur relativement bonne [Chen 2012]. Pour éviter ces problèmes comme incertitudes, nous pouvons choisir un seuil pour déterminer si une faute persiste ou pas, etc. Nous proposons d'utiliser l'approche d'analyse par intervalle afin de détecter voire de diagnostiquer les fautes.

Nous appliquons deux approches différentes avec les techniques de diagnostics. Dans cette thèse : une méthode surveille les états, calculer l'écart entre la sortie sans faute et la sortie subissant des fautes en temps réel. L'autre calcule les paramètres, nous comparons l'écart entre les paramètres dans le cas normal et ceux qui du cas anormal. Pour la dernière méthode, nous adoptons un processus d'inversion ensembliste afin de trouver l'ensemble de solution d'une façon déterministe et exhaustive, peu importe le type ensemble de paramètres admissibles (convexe ou pas). Grâce à cette recherche exhaustive en utilisant "Branch and Bound", le temps d'exécution est long.

Afin d'améliorer et d'affiner les résultats du diagnostic via l'estimation de paramètres (temps et précision), nous mettons en place une technique de diagnostic actif afin de sortir plus de l'information depuis le système étudié. La conception du choix d'entrée est importante avec cette technique. Dans notre approche, cette entrée est conçue hors ligne basée sur une analyse de sensibilité qui cherche à rendre le domaine des solutions plus contractant.

Le chapitre 2 parle de l'outil d'estimation dans le contexte d'ensembliste. Ensuite, le chapitre 3 parlera de l'estimation garantie en utilisant l'analyse par intervalles. Les idées principales pour estimer les états et les paramètres du modèle dynamique seront présentées et expliquées avec des exemples concrets. Pour augmenter la précision des résultats d'estimation de paramètres, nous avons proposé d'utiliser une planification

d'expérience afin de rendre le domaine initial des paramètres à estimer le plus petit possible. Dans le chapitre 4, quelques critères d'optimalité ont été proposés et démontrés avec des exemples choisis du domaine aéronautique. Enfin, cette méthode a été amenée dans le domaine du diagnostic actif afin de trouver les fautes à partir de l'estimation de paramètre dans le chapitre 5. La conclusion et des perspectives seront données dans le dernier chapitre 6.

Outil d'estimation dans le contexte ensembliste

Contents

2.1	Introduction au contexte ensembliste	9
2.1.1	Calcul avec des ensembles	10
2.1.2	Représentation d'une variable incertaine	12
2.2	Introduction à l'analyse par intervalle	12
2.2.1	Pavé et sous pavage	14
2.2.2	Arithmétique des intervalles	15
2.2.3	Propriété algébrique	17
2.2.4	Fonction d'inclusion	19
2.2.5	Pessimisme de surestimation	25
2.2.6	Arithmétique affine	26
2.2.7	Comparaison entre arithmétique affine et analyse par intervalle	27
2.2.8	Librairie d'intervalle	28
2.2.9	Image réciproque d'un ensemble	29
2.2.10	Image directe d'un ensemble	31

2.1 Introduction au contexte ensembliste

Pour décrire l'incertitude, nous avons des outils dans les sciences pour l'ingénieur, l'une est l'approche probabiliste, plus classique et largement appliquée. Mais dans certaines applications, pour obtenir une estimation déterministe par exemple, les incertitudes sur les conditions initiales et les incertitudes sur le système d'évolution peuvent limiter cette approche à cause d'une mauvaise connaissance sur les lois de distribution. Une autre approche peut être utilisée c'est l'approche ensembliste que nous présentons dans ce chapitre.

L'approche ensembliste utilise un ensemble borné \mathbb{X} de \mathbb{R}^n , pour représenter une variable incertaine, que l'on sait contenue dans \mathbb{X} . La représentation ensembliste est plus conservatrice que la représentation probabiliste, sa manipulation est beaucoup plus aisée que celle des variables aléatoires. Par contre, elle ne demande aucune connaissance statistique, ce qui est toujours facile à manipuler en pratique. En ce qui concerne les systèmes automatiques, il arrive souvent que l'erreur du capteur soit donnée en terme de tolérance (erreur relative) et non en terme de densité de probabilité. Dans tous les cas, l'approche ensembliste est une approche naturelle pour traiter les mesures fournies par le capteur.

Il est clair que si les propriétés des variables aléatoires sont connues, une approche ensembliste semble inutile. Dans les problèmes linéaires gaussiens, par exemple, les calculs probabilistes sont élémentaires et très efficaces, alors qu'une approche ensembliste ne présente aucun intérêt.

Dans cette thèse nous traiterons des problèmes non linéaires où l'approche ensembliste est bien adaptée.

Dans les sciences pour l'ingénieur, beaucoup de problèmes consistent à trouver des solutions dans un espace de recherche. Avec l'approche ensembliste, l'espace de recherche est recouvert par un nombre fini de sous-ensembles simples, qui vérifient les principes suivants :

- Si une même propriété est établie pour chacun des sous-ensembles d'une partition de l'espace de recherche, alors cette propriété est valide pour l'espace de recherche tout entier.
- Si l'on arrive à prouver qu'un sous-ensemble ne contient aucune solution du problème, nous pouvons l'éliminer de l'espace de recherche.
- Si l'on n'arrive pas à prouver qu'un sous-ensemble ne contient aucune solution, nous pouvons toujours le décomposer en plusieurs sous-ensembles sur lesquels nous recommençons une analyse.

Avec ces principes, nous trouvons les algorithmes de type *branch and bound* ainsi que tous les algorithmes que nous présenterons dans la suite.

Dans cette thèse, seule l'approche de l'estimation ensembliste [Kosut 1992] [Kieffer 1998] [Raïssi 2004b] sera utilisée et étudiée.

2.1.1 Calcul avec des ensembles

La représentation exacte ou approchée d'un sous-ensemble de \mathbb{R}^n est fondamentale dans notre approche. Elle doit permettre des manipulations efficaces des sous-ensembles d'une manière la plus simple possible. Elle doit présenter tous les sous-ensembles qui nous intéressent et qui sont parfois non connexes et avec tout type de formes. Parmi les approches classiques pour représenter les sous-ensembles de \mathbb{R}^n , nous avons retenu la représentation par recouvrement [Jaulin 2001a]. Cette méthode convient tout d'abord pour définir une classe de sous-ensemble de \mathbb{R}^n que l'on sait représenter et manipuler aisément. Les éléments de cette classe sont appelés des "récipients" [Jaulin 2000]. Cette classe peut être l'ensemble des ellipsoïdes, des pavés, des polytopes, des zonotopes, etc. Pour une classe de "récipient" donnée, un ensemble \mathbb{X} peut généralement être représenté par une union de boîtes qui recouvrent complètement \mathbb{X} . Beaucoup de caractéristiques intéressantes de \mathbb{X} se trouvent contenues dans cette union. Nous les distinguons par une classe de boîtes telle que tout compact \mathbb{X} de \mathbb{R}^n peut être encadré par deux unions de "récipient" \mathbb{X}^- et \mathbb{X}^+ dans le sens où

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+. \quad (2.1)$$

Une connaissance du couple $[\mathbb{X}^-, \mathbb{X}^+]$ nous donne de précieuses informations sur \mathbb{X} . Dans la suite, nous allons utiliser cette représentation où les "récipients" sont des pavés.

Similaires au cas ponctuel, les opérations sur les ensembles peuvent s'en déduire facilement mais avec de petites modifications.

Soient \mathbb{X} un sous ensemble de \mathbb{R}^n , \mathbb{Y} un sous ensemble de \mathbb{R}^m et $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, une fonction vectorielle. Nous définissons l'image directe de \mathbb{X} par \mathbf{f} :

$$\mathbf{f}(\mathbb{X}) \triangleq \{\mathbf{y} \in \mathbb{R}^m \mid \exists \mathbf{x} \in \mathbb{X}, \mathbf{f}(\mathbf{x}) = \mathbf{y}\}, \quad (2.2)$$

l'image réciproque de \mathbb{Y} par \mathbf{f} :

$$\mathbf{f}^{-1}(\mathbb{Y}) \triangleq \{\mathbf{x} \in \mathbb{R}^n \mid \exists \mathbf{y} \in \mathbb{Y}, \mathbf{f}(\mathbf{x}) = \mathbf{y}\}. \quad (2.3)$$

Les opérateurs classiques peuvent être définis pour les ensembles. Ainsi, si \mathbb{X} et \mathbb{Y} sont deux sous ensembles de \mathbb{R}^n ,

$$\mathbb{X} + \mathbb{Y} \triangleq \{\mathbf{x} + \mathbf{y}, \mathbf{x} \in \mathbb{X}, \mathbf{y} \in \mathbb{Y}\}. \quad (2.4)$$

Notons que les multiples occurrences entraînent du pessimisme, par exemple, $\mathbb{X} - \mathbb{X}$ doit être compris comme l'ensemble $\{\mathbf{x} - \mathbf{y}, \mathbf{x} \in \mathbb{X}, \mathbf{y} \in \mathbb{X}\}$ et non pas comme l'ensemble $\{\mathbf{x} - \mathbf{x}, \mathbf{x} \in \mathbb{X}\} = \{\mathbf{0}\}$. Cependant, si $\mathbf{f}(\mathbf{x}) = \mathbf{x} - \mathbf{x}$, $\mathbf{f}(\mathbb{X}) = \{\mathbf{0}\}$.

Les opérations ensemblistes pures regroupent, par exemple, la projection canonique, l'union, l'intersection. Définissons ces opérations dans le cas où \mathbb{X} et \mathbb{Y} sont deux sous ensembles de \mathbb{R}^n et \mathbb{R}^m .

$$proj_i(\mathbb{X}) \triangleq \{x_i \in \mathbb{R} \mid \exists \mathbf{x} = (x_1, \dots, x_i, \dots, x_n)^T \in \mathbb{R}^n, \mathbf{x} \in \mathbb{X}\}, \quad (2.5)$$

$$\mathbb{X} \cap \mathbb{Y} \triangleq \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \in \mathbb{X} \text{ et } \mathbf{x} \in \mathbb{Y}\}, \text{ avec } n = m, \quad (2.6)$$

$$\mathbb{X} \cup \mathbb{Y} \triangleq \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \in \mathbb{X} \text{ ou } \mathbf{x} \in \mathbb{Y}\}, \text{ avec } n = m. \quad (2.7)$$

Les relations binaires ensemblistes classiques :

$$\mathbb{X} \subset \mathbb{Y} \Leftrightarrow \forall \mathbf{x} \in \mathbb{X}, \mathbf{x} \in \mathbb{Y}, \quad (2.8)$$

$$\mathbb{X} = \mathbb{Y} \Leftrightarrow \mathbb{X} \subset \mathbb{Y} \text{ et } \mathbb{Y} \subset \mathbb{X}. \quad (2.9)$$

Soit f une fonction $\mathbb{R}^n \rightarrow \mathbb{R}$, $(x_1, \dots, x_n) \rightarrow y$ dont nous connaissons une expression permettant de l'évaluer. Si $\mathbb{F}(\mathbb{X}_1, \dots, \mathbb{X}_n)$ est le résultat d'un calcul ensembliste sur \mathbb{X} , utilisant cette expression en remplaçant chaque x par \mathbb{X} pour f [Moore 2009]. Alors, nous avons l'inclusion :

$$f(\mathbb{X}_1, \dots, \mathbb{X}_n) \subset \mathbb{F}(\mathbb{X}_1, \dots, \mathbb{X}_n), \quad (2.10)$$

où

$$f(\mathbb{X}_1, \dots, \mathbb{X}_n) = \{f(x_1, \dots, x_n), x_1 \in \mathbb{X}_1, \dots, x_n \in \mathbb{X}_n\} \quad (2.11)$$

2.1.2 Représentation d'une variable incertaine

Maintenant, expliquons le passage d'un modèle statistique à un modèle ensembliste via un exemple académique, ceci nous permettra de comprendre pourquoi une représentation ensembliste permet une manipulation plus simple des variables aléatoires que l'approche probabiliste.

Exemple 1. *Considérons trois variables aléatoires réelles X , Y et Z reliées par la contrainte $Z = X + Y$. Cherchons à déduire les propriétés statistiques de Z à partir de celles de X et Y par l'approche probabiliste puis par l'approche ensembliste.*

— *Approche probabiliste.*

Supposons que X et Y ont des densités de probabilité $\pi(X) = 1/2$ sur $[0, 2]$ et $\pi(Y) = 1/3$ sur $[1, 3]$. Pour calculer $\pi(Z)$, il nous faut une information sur le comportement conjoint de X et Y . Nous allons donc supposer que X et Y sont indépendantes. Ainsi, la densité du couple est uniforme avec $\pi(X, Y) = 1/6$ sur le pavé $[0, 2] \times [1, 3]$.

— *Approche ensembliste.*

L'incertitude associée à X et Y est donnée par un ensemble. Pour être cohérent avec l'approche probabiliste, nous allons prendre l'intervalle $[X] = [0, 2]$ pour X et l'intervalle $[Y] = [1, 3]$ pour Y . Il est alors clair que Z est dans l'intervalle $[Z] = [0 + 1, 2 + 3] = [1, 5]$.

Sur cet exemple, l'approche ensembliste engendre des calculs plus simples et demande moins d'hypothèses sur le comportement conjoint que l'approche probabiliste.

Les manipulations non linéaires sont plus simple avec l'approche ensembliste qu'avec l'approche probabiliste. Si dans cet exemple, Z est donné par $Z = X^2Z$, le calcul de $\pi(Z)$ devient très difficile. Avec l'approche ensembliste, nous obtenons facilement, pour la variable Z , l'intervalle $[Z] = [0, 2]^2 \times [1, 3] = [0, 2 \times 2] \times [1, 3] = [0, 4] \times [1, 3] = [0, 12]$.

Pour des problèmes statistiques non linéaires, nous pouvons utiliser la règle 5σ ou 3σ pour transférer les incertitudes du type probabiliste au type ensembliste, voir l'équation (1.5), où σ désigne l'écart type en probabilités.

2.2 Introduction à l'analyse par intervalle

L'analyse par intervalle a été développée par R.E. Moore [Moore 1966] dans les années 1960. Cette analyse peut être considérée comme une extension des nombres réels, un intervalle est représenté par deux nombre réels. Cet intervalle contient tous les réels encadrés par une borne supérieure et une borne inférieure. Cette idée peut être facilement appliquée dans l'analyse d'erreur comme par exemple : le centre de l'intervalle est une approximation de la vraie valeur et la largeur de l'intervalle est la région incertaine [Moore 1966]. De nombreux travaux ont été abordés sur le sujet du calcul intervalle arithmétique dans des années 1960s. Ensuite, dans des années 1990s - 2000s [Rihm 1994] [Makino 1996] [Nedialkov 1999b] [Makino 2005], le calcul numérique garanti a été beaucoup visité d'un

point de vue mathématique via l'approche intervalle, ce qui a donné naissance de nombreux solveurs librement utilisés de par le monde. L'analyse par intervalle peut donner une vision plus naturelle sur les incertitudes et elle est utilisée dans différentes applications de l'automatique pour l'observation et l'estimation, etc. Dans cette section, des explications importantes concernant l'analyse par intervalle sont présentées. Pour une information supplémentaire, le lecteur pourra consulter le livre de Moore [Moore 2009]. L'analyse par intervalle constitue la base de l'intégration garantie et de l'analyse de sensibilité dont nous parlerons dans le prochain chapitre. Pour uniformiser les notations d'intervalle, la variable intervalle est notée : $[x]$, qui appartient à \mathbb{IR} . Une matrice d'intervalle est notée : $[X]$, qui appartient à $\mathbb{IR}^{n \times m}$ où n et m représentent respectivement le nombre de lignes et le nombre de colonnes dans cette matrice.

D'un point de vue mathématique, un intervalle est un ensemble réel borné,

$$[x] = [a, b] = \{x : a \leq x \leq b\}. \quad (2.12)$$

L'intervalle noté $[x]$ représente un intervalle où les deux bornes extrêmes sont données par la borne inférieure \underline{x} et la borne supérieure \bar{x} . Nous utilisons l'opérateur $\sup([x])$ pour signifier la borne supérieure \bar{x} et l'opérateur $\inf([x])$ pour signifier la borne inférieure \underline{x} .

Par la suite, nous illustrons certaines définitions importantes et notations dans la communauté.

La largeur d'un intervalle est définie par :

$$\omega([x]) = \bar{x} - \underline{x}. \quad (2.13)$$

Le point milieu d'un intervalle $[x]$ est noté :

$$m([x]) = \frac{(\underline{x} + \bar{x})}{2}. \quad (2.14)$$

La valeur absolue est :

$$|[x]| = \max(|\underline{x}|, |\bar{x}|). \quad (2.15)$$

Deux intervalles sont dits égaux si et seulement si leurs deux bornes sont égales :

$$[x] = [y] \quad \text{si } \underline{x} = \underline{y} \text{ et } \bar{x} = \bar{y}. \quad (2.16)$$

Nous disons qu'un intervalle est inclu dans un autre intervalle :

$$[x] \subseteq [y] \quad \text{si et seulement si } \underline{y} \leq \underline{x} \text{ et } \bar{x} \leq \bar{y}. \quad (2.17)$$

L'intersection de deux intervalles est toujours un intervalle défini par l'expression :

$$[x] \cap [y] = \{z \in \mathbb{IR} \mid z \in [x] \text{ et } z \in [y]\}, \quad (2.18)$$

et l'union de deux intervalles est définie ci-dessous :

$$[x] \cup [y] = \{z \in \mathbb{IR} \mid z \in [x] \text{ ou } z \in [y]\}. \quad (2.19)$$

Afin que le calcul ensembliste soit fermé et respecte les règles classiques, nous définissons l'enveloppe d'un sous-ensemble X de \mathbb{IR} comme le plus petit intervalle $[X]$ qui le contient. Par exemple, l'enveloppe de deux intervalles $[2, 3] \cup [4, 8]$ est égale $[2, 8]$.

L'enveloppe de deux intervalles est défini par :

$$[x] \sqcup [y] = [\min(\underline{x}, \underline{y}), \max(\bar{x}, \bar{y})]. \quad (2.20)$$

ce qui est toujours un intervalle. Alors nous avons :

$$[x] \cup [y] \subseteq [x] \sqcup [y]. \quad (2.21)$$

2.2.1 Pavé et sous pavage

Définition 2. *Un pavé $[x]$ (ou boîte) de \mathbb{IR}^n est le produit cartésien de n intervalles pouvant s'écrire sous la forme suivante :*

$$[x] = [\underline{x}_1, \bar{x}_1] \times \cdots \times [\underline{x}_n, \bar{x}_n] = [x_1] \times [x_2] \times \cdots \times [x_n]. \quad (2.22)$$

Un pavé $[x]$ est appelé un vecteur intervalle car ses n composantes sont des intervalles de \mathbb{R} . L'ensemble de tous les pavés de \mathbb{IR}^n

Cela change aussi la définition de la longueur ω , qui correspond à la longueur de son plus grand côté :

$$\omega([x]) = \max(\bar{x}_i - \underline{x}_i), \quad i = 1, \dots, n. \quad (2.23)$$

Par exemple, soit un pavé $[x] = [-2, 6] \times [3, 4]$,

$$\omega([x]) = 8.$$

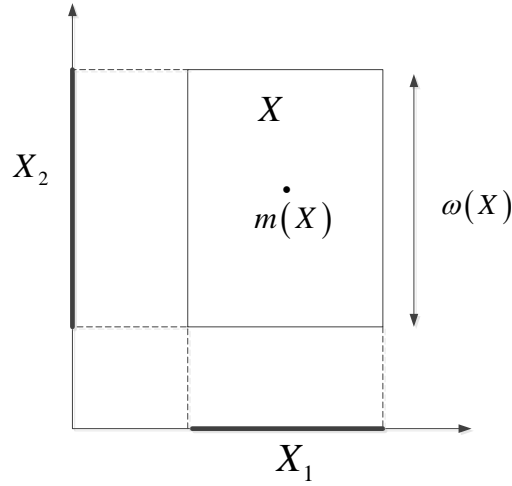
De même pour la définition du point milieu :

$$m([x]) = (m([x_i]), i = 1, \dots, n). \quad (2.24)$$

La bisection d'un pavé $[x]$ est l'opération qui consiste à couper le pavé en deux sur son plus grand côté. Nous reprenons le pavé précédent $[x]$ qui génère deux nouvelles pavés :

$$[x]_1 = [-2, 2] \times [3, 4] \quad \text{et} \quad [x]_2 = [2, 6] \times [3, 4].$$

Un sous-pavage \mathcal{P} de \mathbb{IR}^n est une union de pavés de \mathbb{IR}^n qui ne se chevauchent pas, ce qui signifie que leur intersection deux à deux est vide sauf éventuellement s'ils ont une frontière commune.

FIGURE 2.1 – La longueur et le point milieu d'un pavé $X = (X_1, X_2)$

2.2.2 Arithmétique des intervalles

L'arithmétique des intervalles est définie par la manière dont les opérations élémentaires sont appliquées, par exemple l'addition, la soustraction, la multiplication.

Les opérations d'addition et soustraction sont :

$$[\underline{x}, \bar{x}] + [\underline{y}, \bar{y}] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}], \quad (2.25)$$

$$[\underline{x}, \bar{x}] - [\underline{y}, \bar{y}] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]. \quad (2.26)$$

L'opération de multiplication :

$$[x] \times [y] = [\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})]. \quad (2.27)$$

Par exemple :

$$[1, 3] \times [-2, 5] = [3 \times (-2), 3 \times 5] = [-6, 15]. \quad (2.28)$$

$$[x]^2 = [\min(\underline{x}^2, \bar{x}^2), \max(\underline{x}^2, \bar{x}^2)],$$

si 0 n'est pas dans $[x]$, sinon $[0, \max(\underline{x}^2, \bar{x}^2)]$. Notons que $[x]^2$ peut être traité comme $[x] \times [x]$, ce qui demande une multiplication d'intervalles normale. Sa borne inférieure n'est plus toujours positive selon la définition carrée.

L'inverse d'un intervalle est défini comme :

$$\frac{1}{[x]} = \left[\frac{1}{\bar{x}}, \frac{1}{\underline{x}} \right], \quad (2.29)$$

où $[x]$ est un intervalle qui ne contient pas 0. Si $[x]$ contient 0, cet ensemble n'est pas borné et il ne peut pas être représenté par un intervalle dont les extrêmes sont des nombres réels. Cela aboutit à la perte d'information quand nous faisons la division. Néanmoins, nous pouvons diviser cet intervalle en deux ensembles : $[-\infty, C_1]$ et $[C_2, \infty]$. Cela a été implémenté par plusieurs outils dont nous parlerons dans la suite.

La division de deux intervalles est définie par :

$$\frac{[x]}{[y]} = [x] \times \frac{1}{[y]}. \quad (2.30)$$

Une matrice d'intervalles est une matrice ayant des coefficients qui sont des intervalles. Une matrice d'intervalles peut être écrite :

$$[A] = \begin{bmatrix} [a_{11}] & \cdots & [a_{1n}] \\ \vdots & \ddots & \vdots \\ [a_{n1}] & \cdots & [a_{nn}] \end{bmatrix} = ([a_{ij}]), \quad (2.31)$$

où $i, j = 1, \dots, n$.

Nous introduisons les opérations matricielles dans le cas intervalle. Si nous avons $[A], [B] \subset \mathbb{IR}^{n \times n}$, $[a], [b] \subset \mathbb{IR}$ et $i, j = 1, \dots, n$, alors nous avons :

1. $[a][A]_{ij} = ([a][a_{ij}])$
2. $([A] + [B])_{ij} = ([a_{ij}] + [b_{ij}])$,
3. $([A] - [B])_{ij} = ([a_{ij}] - [b_{ij}])$,
4. $([A][B])_{ij} = (\sum_{k=1}^n [a_{ik}][b_{kj}])$.

Une matrice carrée A d'ordre n est dite inversible s'il existe une matrice B telle que $AB = BA = I$ où I représente la matrice identité d'ordre n . Cette matrice est notée A^{-1} .

Dans le cas intervalle nous n'avons pas une définition aussi simple [Pasca 2010]. Soit $[A]$ une matrice intervalle.

Définition 3. La matrice $[A]$ est dite inversible si toutes les matrices carrées scalaires qui appartiennent à $[A]$ sont inversibles. Nous noterons $[A]^{-1}$ cette matrice inverse.

La matrice $[A]^{-1}$ est alors la matrice intervalle la plus "petite" qui contient l'ensemble des matrices A^{-1} où A appartient à $[A]$. Le sens qui est donné "à la plus petite" :

$[A]^{-1}$ est contenue dans toutes les matrices $[B]$ qui contiennent l'ensemble des matrices A^{-1} où A appartient à $[A]$.

La matrice $[A]^{-1}[A]$ est proche de la matrice identité.

Proposition 1. Nous notons $[I]_\lambda$ qui nous permet d'avoir une relation comme :

$$[A]^{-1}[A] \subseteq [I]_\lambda \quad (2.32)$$

Nous définissons une matrice identité construite avec $\lambda \in]0, 1[$, représentée par :

$$[I]_\lambda = \begin{bmatrix} [1 - \lambda, 1 + \lambda] & \cdots & [0 - \lambda, 0 + \lambda] \\ \vdots & \ddots & \vdots \\ [0 - \lambda, 0 + \lambda] & \cdots & [1 - \lambda, 1 + \lambda] \end{bmatrix}. \quad (2.33)$$

Le terme $[1 - \lambda, 1 + \lambda]$ n'apparaît que dans la ligne diagonale, les autres éléments dans cette matrice sont tous sous forme $[0 - \lambda, 0 + \lambda]$.

La norme de matrice d'intervalles [Raena 2011] est une projection $\|\cdot\|: \mathbb{IR}^{m \times n} \rightarrow \mathbb{R}$ définie par :

$$\|[A]\| = \sup\{\|A\|, A \in [A]\}, \quad (2.34)$$

$\|A\|$ représente la norme de A quelconque. Avec cette définition de norme de matrice d'intervalle, pour chaque $[A], [B] \in \mathbb{IR}^{m \times n}$, $[A] \subseteq [B]$ résulte $\|[A]\| \leq \| [B] \|$.

Si $[A] \in \mathbb{IR}^{n \times n}$, ses composantes sont notées $[a]_{i,j} \in \mathbb{IR}$, alors :

$$\|[A]\|_F^2 = \sum_{i=1}^N \sum_{j=1}^N |[a]_{i,j}|^2 \leq N * \max_i \left(\sum_{j=1}^N |a_{i,j}|^2 \right) = N \| [A] \|^2 \quad (2.35)$$

où $\|\cdot\|_F$ est la norme de Frobenius pour une matrice intervalle [Raena 2011].

2.2.3 Propriété algébrique

L'arithmétique d'intervalles a des propriétés similaires à celles des nombres réels. Dans cette section, nous allons introduire les propriétés les plus importantes [Moore 2009].

2.2.3.1 Propriété d'associativité et de distributivité

L'addition et la multiplication sont associative. C'est-à-dire que :

$$[x] + ([y] + [z]) = ([x] + [y]) + [z], \quad (2.36)$$

$$[x]([y][z]) = ([x][y])[z]. \quad (2.37)$$

La propriété de commutativité est donnée par :

$$[x] + [y] = [y] + [x], \quad (2.38)$$

$$[x][y] = [y][x]. \quad (2.39)$$

Mais la propriété de distributivité n'est pas toujours vérifiée, par exemple :

$$[2, 5] \times [0, 0] = [2, 5] \times 0 = [0, 0] = 0, \quad (2.40)$$

et

$$[2, 5] \times [1, 1] - [2, 5] \times [1, 1] = [-3, 3] \neq 0, \quad (2.41)$$

Cela implique que la fonction d'intervalle n'est pas définie de manière unique, le résultat calculé via différentes formes de fonction varie. Mais certains cas particuliers restent établis :

$$x([y] + [z]) = x[y] + x[z], \quad (2.42)$$

$$[x]([y] + [z]) = [x][y] + [x][z] \quad \text{si} \quad \inf([y][z]) > 0. \quad (2.43)$$

Mais nous avons la propriété de sous-distributivité :

$$[x]([y] + [z]) \subseteq [x][y] + [x][z]. \quad (2.44)$$

Ce qui implique que nous pouvons toujours arranger les termes dans la fonction d'intervalle, afin d'obtenir un résultat plus étroit (petit intervalle) [Nedialkov 1999a].

2.2.3.2 Propriétés de multiplication dans certains cas

Ensuite, nous introduisons quelques lemmes qui seront utiles dans le chapitre 4 :

Lemme 1. *Le produit de deux intervalles $[u]$ et $[v]$ dont l'un est symétrique $[u] = -[u]$ et l'autre un intervalle quelconque est donné par :*

$$[u] \times [v] = [u] \times \text{mag}([v]). \quad (2.45)$$

Nous rappelons que : $\text{mig}([x]) = \min(|\underline{x}|, |\overline{x}|)$, si $0 \notin [x]$, sinon $\text{mig}([x]) = 0$, quand $[x]$ devient une matrice, cette opération sera effectuée composantes par composantes et $\text{mag}([x]) = \max(|\underline{x}|, |\overline{x}|)$, pour plus de détail voir [Chabert 2014].

Démonstration. Supposons que a, b, c sont des nombres positifs et que $a \leq b \leq c$. La forme $[-a, a]$ représente un intervalle symétrique.

Considérons l'intervalle $[b, c]$, alors :

$$[-a, a] \times [b, c] = [-ac, ac] = [-a, a] \times \text{mag}([b, c]).$$

Considérons l'intervalle comme $[-c, -b]$, alors :

$$[-a, a] \times [-c, -b] = [-ac, ac] = [-a, a] \times \text{mag}([-c, -b]).$$

Considérons l'intervalle comme $[-b, c]$, alors :

$$[-a, a] \times [-b, c] = [-ac, ac] = [-a, a] \times \text{mag}([-b, c]).$$

Considérons l'intervalle comme $[-c, b]$, alors :

$$[-a, a] \times [-c, b] = [-ac, ac] = [-a, a] \times \text{mag}([-c, b]).$$

Démonstration terminée. □

Lemme 2. *Le produit de deux intervalles $[u]$ et $[v]$ tels que $0 \in [u]$ et $0 \notin [v]$ est donné par :*

$$[u] \times [v] = [u] \times \text{mag}([v]), \quad . \quad (2.46)$$

Démonstration. Supposons que a, b, c, d sont des nombres positifs tels que $a \leq b$ et $c \leq d$. La forme $[-a, b]$ représente un intervalle contenant 0.

Considérons l'intervalle comme $[c, d]$, alors :

$$[-a, b] \times [c, d] = [-ad, bd] = [-a, b] \times \text{mag}([c, d]),$$

ceci est encore vrai pour $a \geq b$.

Considérons l'intervalle comme $[-d, -c]$, alors :

$$[-a, b] \times [-d, -c] = [-bd, ad] = [-a, b] \times \text{mag}([-d, -c]),$$

ceci est encore vrai pour $a \geq b$.

Démonstration terminée. □

2.2.3.3 Propriété d'inclusion avec les matrices intervalles

Lemme 3. si $[v] \subset [\mu]$, alors $[B][v] \subset [B][\mu]$, où $[B]$ est une matrice d'intervalles, $[v]$ et $[\mu]$ sont des vecteurs d'intervalles.

Démonstration. Nous avons l'inclusion $[v] \subset [\mu]$, cela signifie que pour tous les éléments du vecteur satisfont :

$$[v_j] \subset [\mu_j] \quad j = 1, \dots, n. \quad (2.47)$$

D'autre part nous avons une matrice d'intervalle $[B]$ avec des éléments $[B_{ij}]$ où $i, j = 1, \dots, n$, nous démontrons que dans la ligne i , nous avons

$$[B_{i1}][v_1] + \dots + [B_{in}][v_n] \subset [B_{i1}][\mu_1] + \dots + [B_{in}][\mu_n]. \quad (2.48)$$

C'est équivalent de montrer que pour toutes les lignes, nous avons :

$$[b_{ij}][v_1] \subset [b_{ij}][\mu_1]. \quad (2.49)$$

Ceci est la propriété d'intervalle arithmétique [Moore 2009]. □

Ce lemme sera utilisé dans le chapitre 4.

2.2.4 Fonction d'inclusion

La notion de fonction a été étendue dans l'analyse par intervalle. Nous notons $[f]$ la fonction d'inclusion sous forme intervalle avec n variables $[x_1], \dots, [x_n]$. $[f]$ est une fonction d'inclusion de f quand :

$$f([x_1], \dots, [x_n]) \subseteq [f]([x_1], \dots, [x_n]) \quad (2.50)$$

Cela implique qu'une fonction d'inclusion sous forme intervalle $[f]$ encadre les bornes de cette fonction quand les variables de cette fonction sont comprises dans les intervalles $[x_1], \dots, [x_n]$.

Pour toutes les fonctions rationnelles, les fonctions ont des formes dites naturelles comme fonction d'inclusion sous forme intervalle. Nous pouvons remplacer les variables réelles dans la fonction par des variables intervalles avec les arithmétiques d'intervalle.

Par exemple [Moore 1979], considérons un polynôme :

$$p(x) = 1 - 5x + \frac{1}{3}x^3. \quad (2.51)$$

Nous voulons trouver le rayon pour toutes les valeurs de $p(x)$ quand x est dans l'intervalle $[2, 3]$. Une forme étendue naturelle d'intervalle de p est :

$$P([x]) = 1 - 5[x] + \frac{1}{3}[x] \times [x] \times [x]. \quad (2.52)$$

Calculons $P([2, 3])$ avec l'arithmétique d'intervalle, nous obtenons :

$$P([2, 3]) = 1 - 5[2, 3] + \frac{1}{3}[8, 27] = [-\frac{34}{3}, 0]. \quad (2.53)$$

C'est-à-dire que cet intervalle inclut toutes les possibilités de x dans l'intervalle $[2, 3]$ pour la fonction $p(x)$.

Nous allons maintenant évaluer un processus qui approche le résultat de $f(x_1, \dots, x_n)$ à partir d'une union d'intervalles. Nous commençons par la continuité :

Définition 4. [Moore 2009] Une extension d'intervalle $[f]$ est Lipschitz en x_0 , si et seulement si il existe une constante L satisfaisant $\omega([f]([x])) \leq L\omega([x])$ pour tout les $[x] \subseteq [x_0]$

Ici, $[x]$ peut être aussi un vecteur d'intervalle comme $[x] = ([x_1], \dots, [x_n])$. Nous pouvons déduire que la fonction d'inclusion naturelle est Lipschitz en $[x_0]$.

Définition 5. [Moore 2009] La fonction $[f] = [f]([x_1], \dots, [x_n])$ est inclusion isotonique si

$$[y_i] \subseteq [x_i] \text{ pour } i = 1, \dots, n \Rightarrow [f]([y_1], \dots, [y_n]) \subseteq [f]([x_1], \dots, [x_n]). \quad (2.54)$$

Notons que pour le vecteur d'intervalle, cette définition demande aussi pour tous les sous-intervalles, d'avoir l'inclusion isotonique. En particulier, l'opération d'extension d'intervalle doit être satisfaite.

Définition 6. [Moore 2009] Soit $[f]([x])$ une inclusion isotonique, Lipschitz, étendue aux intervalles pour $[x] \subseteq [x_0]$, alors la réunion d'intervalles ci-dessous

$$[f]_{(N)}([x]) = \bigcup_{j=1}^N [f]([x_{1,j_1}], \dots, [x_{n,j_n}]), \quad (2.55)$$

est appelée un raffinement N de $[f]$ pour tout $[x]$, N est un entier positif et $[x_{i,j}] = [\underline{x}_i + (j-1)\omega([x]_i)/N, \underline{x}_i + j\omega([x]_i)/N]$, $j = 1, \dots, N$.

Exemple 2. Considérons une fonction $f(x) = x - x^2$ pour $x \in [0, 1]$. Il est facile à vérifier que $f([0, 1]) = [0, \frac{1}{4}]$. La fonction d'inclusion peut être décrite de la manière suivante :

TABLE 2.1 – Différents raffinements

N	$[f]$	Temps d'exécution
1	$[-1, 1]$	0.007700
2	$[-0.5, 0.75]$	0.008375
10	$[-0.1, 0.35]$	0.016579
100	$[-0.01, 0.26]$	0.123618
1000	$[-0.001, 0.251]$	0.969033
10000	$[-0.0001, 0.2501]$	9.180341

$$[f]([x]) = [x] - [x] \times [x]. \quad (2.56)$$

Nous avons $[f]([0, 1]) = [-1, 1]$, dont le rayon excède le rayon de $f([0, 1])$ de $2 - \frac{1}{4} = \frac{7}{4}$. Soit l'intervalle $[0, 1]$ divisé en n sous-intervalles $[x_i] = [(i-1)n, i/n]$, $1 \leq i \leq n$. Nous obtenons le Tableau 2.1.

Nous pouvons voir que la méthode de raffinement est une manière de calculer arbitrairement fine la borne d'une fonction d'inclusion, mais il est plus important d'être capable de l'obtenir rapidement.

[Skelboe 1974] a introduit un algorithme plus efficace pour calculer le raffinement. Nous cherchons tout d'abord, une borne inférieure sur l'inclusion de $[f]$, ensuite, nous répétons ce processus pour la borne inférieure sur $-[f]$. Pendant la bisection, nous évaluons $[f]$ seulement sur le sous ensemble qui produit la plus petite borne. Ici, c'est avantageux d'utiliser la forme centrée et la forme en valeur moyenne pour que la simplicité de calcul.

Nous présentons l'Algorithme 1 de Skelboe-Moore.

Dans l'Algorithme 1, $[f]([x])$ représente la borne inférieure pour tous $[x_0]$, \mathcal{C} représente la liste des pavés traités et \mathcal{L} représente la liste des pavés à traiter. f_{bs} enregistre la borne inférieure connue jusqu'à présent. L'union de deux listes est $[x_0]$. La fonction *ExtraireSommet* représente l'opération de sortie sur le premier sommet dans la liste \mathcal{L} .

La méthode de raffinement "straightforward" divise le domaine de l'intervalle en N sous intervalles en dimension n qui nécessite N^n fois le calcul de $f([x])$. Pour la plupart des fonctions avec des extrêmes isolés sur x_0 , l'algorithme ci-dessus, afin d'arriver à une précision satisfaisante, demande $Cn(\log_2 N)$ nombre d'évaluations de la fonction d'inclusion où C est une constante indépendante de n et N .

Exemple 3. Nous utilisons la méthode de Skelboe Moore pour trouver les bornes extrêmes de la fonction qui suit :

$$f(x, y) = x^2 - 2y^2 + 5 \quad (2.57)$$

sur le pavé $[x] = ([-1, 1], [-1, 1])$.

Nous obtenons la réponse dans la suite :

Algorithme 1 Borne inférieure de $([f]([x]))$ **Input:** $[f], \varepsilon, [x_0]$;**Output:** $[f]([x]), \mathcal{C}, \mathcal{L}$;initialization : $f_{bs} := f(m([x_0]))$; $[x] := [x_0]$;**while** $\mathcal{L} \neq \emptyset$ **do** Bisection over the maximum width side $[x]$, $[x] = [x_1] \cup [x_2]$; $f_{bs} := \min\{f_{bs}, \overline{f(m([x_1]))}, \overline{f(m([x_2]))}\}$ **if** $\max\{\overline{f([x_1])}, \overline{f([x_2])}\} - \min\{\underline{f([x_1])}, \underline{f([x_2])}\} < \varepsilon$ **then** Put $[x_1]$ et $[x_2]$ into the list \mathcal{C} **if** $\mathcal{L} \neq \emptyset$ **then** $[x] := \text{ExtraireSommet}(\mathcal{L})$ **if** $f([x]) > f_{bs}$ **then** Return the lower bound on the top of the lists \mathcal{C} , \mathcal{C} and \mathcal{L} **else** Return the lower bound on the top of the lists \mathcal{C} and \mathcal{C} . **else** Put $([x_1], f([x_1]))$ et $([x_2], f([x_2]))$ on the top of the list \mathcal{L} $[x] := \text{ExtraireSommet}(\mathcal{L})$

Pour la valeur maximale :

Borne inférieure = $[3.0000, 3.0000]$ Pavé correspondant = $[0.0000, 0.0001]$ $[-1.0000, -0.9999]$

Pour la valeur minimale :

Borne inférieure = $[-6.0000, -6.0000]$ Pavé correspondant = $[0.9999, 1.0000]$ $[-0.0001, 0.0000]$

Pour une même fonction f , il existe différentes fonctions d'inclusion $[f]$. Parmi ces fonctions d'inclusion, il est intéressant de rechercher des fonctions d'inclusion convergentes. Une fonction d'inclusion est convergente si et seulement si, pour un pavé $[x]$ de \mathbb{R}^n :

$$\lim_{\omega([x]) \rightarrow 0} \omega([f]([x])) = 0 \quad (2.58)$$

Différentes méthodes peuvent être mises en oeuvre pour obtenir une fonction d'inclusion convergente. La méthode la plus simple consiste à remplacer dans l'expression les nombres par les intervalles auxquels ils appartiennent et les fonctions élémentaires réelles par leurs extensions aux intervalles. La fonction d'inclusion obtenue est la **fonction d'inclusion naturelle**.

Considérons la fonction :

$$f(x) = x^2 + 3x + 2 \quad (2.59)$$

une fonction d'inclusion naturelle de f est donnée par :

$$[f]([x]) = [x] \times [x] + 3 \times [x] + 2. \quad (2.60)$$

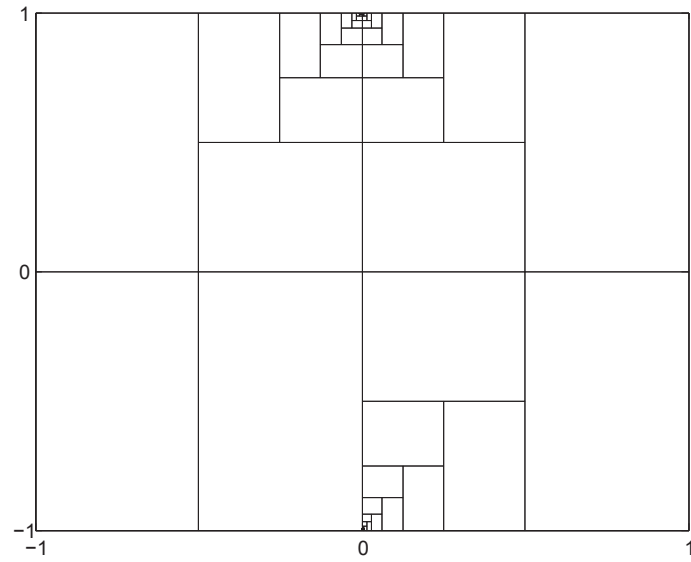


FIGURE 2.2 – Pavé pour la borne inférieure

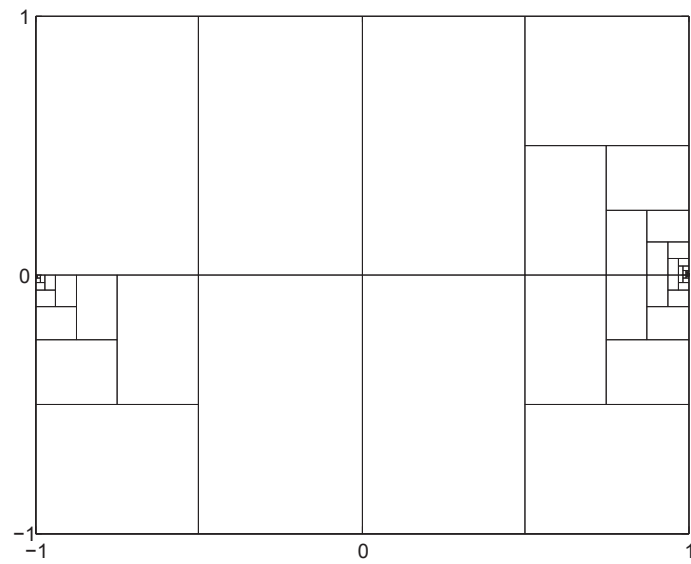


FIGURE 2.3 – Pavé pour la borne supérieure

En évaluant f pour $[x] = [-2, 1]$, nous obtenons

$$[f]([-2, 1]) = [-2, 1] \times [-2, 1] + 3 \times [-2, 1] + 2 = [-2, 4] + [-4, 5] = [-6, 9], \quad (2.61)$$

ou bien :

$$[f]([-2, 1]) = [-2, 1]^2 + 3 \times [-2, 1] + 2 = [0, 4] + [-6, 3] + 2 = [0, 4] + [-4, 5] = [-4, 9]. \quad (2.62)$$

Remarquons que l'image de l'intervalle $[x]$ par f est $f([-2, 1]) = [-1/4, 6]$. La relation d'inclusion (2.50) est bien vérifiée.

La fonction d'inclusion naturelle est rarement minimale. Un pessimisme est souvent introduit par celle-ci provenant du fait que chaque occurrence d'une variable intervalle est considérée comme variant indépendamment des autres.

Reprenons l'exemple traité dans [Kieffer 1999] pour comparer les performances des fonctions d'inclusion à partir de quatre formulations d'une même fonction.

$$\begin{aligned} f_1(x) &= x(x+1) \\ f_2(x) &= x \times x + x \\ f_3(x) &= x^2 + x \\ f_4(x) &= \left(x + \frac{1}{2}\right)^2 - \frac{1}{4} \end{aligned}$$

Pour $[x] = [-1, 1]$, nous obtenons

$$\begin{aligned} [f_1]([x]) &= [x]([x] + 1) = [-2, 2] \\ [f_2]([x]) &= [x] \times [x] + [x] = [-2, 2] \\ [f_3]([x]) &= [x]^2 + [x] = [-1, 2] \\ [f_4]([x]) &= \left([x] + \frac{1}{2}\right)^2 - \frac{1}{4} = \left[-\frac{1}{4}, 2\right] \end{aligned}$$

Ainsi les expressions $[x] \times [x]$ et $[x]^2$ ne donnent pas le même résultat, car la variable intervalle intervient deux fois dans la première expression. Chaque occurrence de $[x]$ peut donc varier indépendamment, ce qui introduit un pessimisme dans l'évaluation de la fonction f . La quatrième formulation fournit le plus petit intervalle solution, ainsi $[f_4]$ est minimale.

La longueur des intervalles obtenus dépend donc assez fortement du nombre d'occurrences de chaque variable dans l'expression initiale de la fonction f . Une fonction d'inclusion qui fournit un résultat qui n'est pas égale à l'évaluation intervalle de la fonction réelle sur l'intervalle considéré est dite pessimiste.

Pour réduire ce pessimisme, il faut réécrire l'expression de manière à réduire le nombre d'occurrences de chaque variable. Si chacune des variables n'intervient qu'une seule fois dans l'expression de f alors la fonction d'inclusion naturelle sera minimale. Comme il n'existe pas de méthode systématique pour trouver la fonction d'inclusion minimale, nous utiliserons la fonction d'inclusion de la forme centrée qui est plus efficace en terme de calcul et plus robuste en terme d'inclusion que les autres types de fonction d'inclusions.

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, une fonction dérivable sur un pavé $[x]$ et m la valeur au milieu de $[x]$. Le théorème de la valeur moyenne donne

$$\forall x \in [x], \exists z \in [x] \mid f(x) = f(m) + f'(z)(x - m). \quad (2.63)$$

où $f'(z)$ est une matrice dans l'espace $\mathbb{R}^{1 \times n}$ correspondant à la dérivée de f en z . Par conséquent,

$$f(x) \in f(m) + f'([x])(x - m) \quad (2.64)$$

d'où

$$f([x]) \subseteq f(m) + [f']([x])([x] - m) \quad (2.65)$$

La fonction intervalle définie par

$$[f_c]([x]) = f(m) + [f']([x])([x] - m), \quad (2.66)$$

où $[f']$ une fonction d'inclusion de la dérivée f' de f , est donc une fonction d'inclusion de f . Cette fonction est appelée la forme centrée de f sur l'intervalle $[x]$. Elle donne des résultats souvent moins pessimistes que la fonction d'inclusion naturelle lorsque $[x]$ a une longueur faible et que le problème de dépendance se pose pour f [Neumaier 2003].

Il existe aussi d'autres formes de fonction d'inclusion, veuillez les trouver dans le livre de [Moore 2009].

2.2.5 Pessimisme de surestimation

Les bornes évaluées par les fonctions d'inclusion intervalles sont souvent plus larges que les bornes réelles de cette fonction. Cela induit du pessimisme dans l'analyse par intervalle. Il y a un effet d'enveloppement qui introduit aussi du pessimisme dans l'analyse par intervalles.

2.2.5.1 Le problème d'indépendance

L'analyse par intervalles ne peut pas savoir les multis occurrences de la même variable. Cela va introduire la surestimation dans certains cas. Par exemple, $x - x = 0$ pour tous $x \in [2, 5]$, mais quand nous évaluons $[2, 5] - [2, 5]$ via l'analyse par intervalle, le résultat est $[-3, 3]$. La solution $[0, 0]$ est incluse dans cet intervalle, mais avec une large surestimation de la borne réelle. Ce type de problème est appelé : le problème d'indépendance par la communauté de l'analyse par intervalle.

Une forme différente changera la solution d'intervalle. Nous pouvons toujours réécrire la fonction d'inclusion pour diminuer cette surestimation. Mais est-ce que cette forme sera la 'meilleure' dans le sens où la surestimation est la plus petite possible ? La réponse immédiate oui, mais cela n'est pas accessible tout le temps, la meilleure forme n'existe pas toujours dans les fonctions traitées.

Au lieu de réécrire la fonction d'intervalle, le problème d'indépendance peut aussi être réduit via la subdivision. Cette méthode propose de diviser les intervalles à plusieurs sous

intervalles et combiner les résultats calculés par ces sous intervalles. Un temps de calcul est plus long est le seul désavantage de cette méthode, cet indice est proportionnel à l'augmentation du nombre de n de la dimension [Alessi 2007].

2.2.5.2 Effet d'enveloppement

Un intervalle de dimension un peut être considéré comme un segment de l'axe des abscisses. Un vecteur d'intervalle de dimension deux est un rectangle et un vecteur d'intervalle est un polytope avec les côtés parallèles aux axes orthogonaux.

$$[x], \quad [[x], [y]], \quad [[x], [y], [z]] \quad (2.67)$$

$$[x] = [1, 4], \quad [y] = [1, 4], \quad [z] = [1, 4]. \quad (2.68)$$

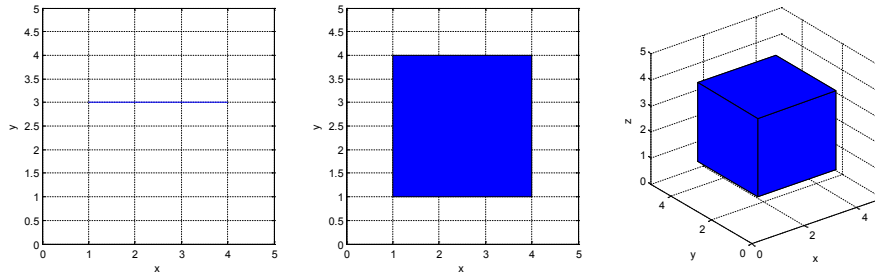


FIGURE 2.4 – Vecteur d'intervalle de dimension 1, 2 et 3

Si nous regardons le cas de dimension 2, supposons que l'on veuille encadrer une zone non rectangulaire, le vecteur d'intervalle (boîte) choisi est toujours plus large (à l'extérieur de la vraie zone). Cela a été présenté dans la Figure 2.4.

Cet effet d'enveloppement ainsi qu'une méthode de "diminution" seront présentés dans le chapitre 3, où nous considérons l'effet d'enveloppement dans un processus itératif comme l'intégration garantie.

2.2.6 Arithmétique affine

L'arithmétique affine (AA) est une méthode similaire à l'analyse par intervalle. Elle calcule les enveloppes garanties pour des valeurs données, elle est aussi capable d'évaluer une fonction avec un ensemble de valeurs, justement comme ce qui se fait par l'analyse par intervalles. Cette méthode a beaucoup moins de surestimation [Stolfi 2003].

La surestimation par l'analyse par intervalles dépend des signes des occurrences des variables et de la fonction dérivée lorsque nous prenons la forme centrée comme fonction d'inclusion. L'arithmétique affine essaye de réduire le problème de dépendance pour l'analyse d'intervalle par le biais de la surveillance des occurrences sur les variables [Stol 1997].

"The approximation error incurred in each AA operation normally has a quadratic dependency on the size of the input intervals, even when the operands are correlated." [Stolfi 2003]

Dans l'arithmétique affine, une variable x est présentée par une expression de la forme :

$$\hat{x} = x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n. \quad (2.69)$$

C'est un polynôme d'ordre 1 avec le symbole de bruit ε_i avec un coefficient de point flottant x_i , où ε est toujours entre -1 et 1.

Le calcul de AA est similaire à celui de l'AI, les opérations algébriques standards peuvent être trouvées dans les publications de Stolfi [Stolfi 2003] et Figueiredo [de Figueiredo 2004].

L'arithmétique affine est plus complexe et onéreuse par rapport au temps de calcul, mais le problème de dépendance peut être réduit significativement par rapport au calcul normal comparé avec l'analyse par intervalles. La librairie pour AA est moins mature que celles de l'analyse d'intervalle. Libaffa [Gay 2006] a cessé sa mise à jour en 2006. Des librairies plus récente peuvent être trouvées, telles que *aaflib* et *YalAA*.

2.2.7 Comparaison entre arithmétique affine et analyse par intervalle

Nous constatons que le problème de dépendance entraîne beaucoup de surestimation quand nous calculons la fonction d'inclusion à partir des variables d'intervalles. Un exemple d'un polynôme montre bien ce problème :

$$F([x]) = x - x \quad (2.70)$$

La librairie de YalAA arithmétique affine et Profil/Bias la librairie de calcul d'intervalle ont été utilisées pour voir les différences entre ces deux approches.

Les résultats calculés par AA et IA sont présentés sous la forme d'une largeur d'intervalle.

Rayon d'intervalle of $[x]$	Arithmétique d'intervalle	Arithmétique affine
[1,1]	[0,0]	[0,0]
[1,3]	[-2,2]	[0,0]
[-2,3]	[-5,5]	[0,0]

Dans le domaine informatique, il est important de trouver l'enveloppe pour l'inclusion de la fonction [Neumaier 2011] :

$$f := \frac{a(\omega^2 + x^2 - y^2 - z^2) + 2b(xy - wz) + 2c(xz + wy)}{\omega^2 + x^2 + y^2 + z^2}. \quad (2.71)$$

Nous nous intéressons à l'encadrement de l'enveloppe avec ces variables d'intervalle a, b, c, ω, x, y, z :

$$\begin{aligned} a &\in [7, 9], b \in [-1, 1], c \in [-1, 1], \omega \in [-0.9, -0.6] \\ x &\in [-0.1, 0.2], y \in [0.3, 0.7], z \in [-0.2, 0.1] \end{aligned} \quad (2.72)$$

Le résultat donné par l'arithmétique affine est l'intervalle : $[-14.5626687, 23.0269865]$ ce qui n'est pas meilleur que celui que nous avons obtenu par l'arithmétique d'intervalle

en utilisant la forme centrée : $[-7.4889, 19.2889]$. Ce qui implique que l'arithmétique affine n'est pas toujours mieux que l'arithmétique d'intervalle.

Un autre exemple a été pris depuis [Stolfi 2003] :

$$g(x) = \sqrt{x^2 - x + 1/2} / \sqrt{x^2 + 1/2} \quad (2.73)$$

[x]	Arithmétique d'intervalle $[g]([x])$	Arithmétique affine $[g]([x])$
[5,5.1]	[0.8772,0.9185]	[0.8790,0.9164]
[5,6]	[0.7309,1.1115]	[0.7221,1.0991]
[5,10]	[0.3927,1.9353]	[0.0088,2.1683]
[5,20]	[0.1171,3.9383]	$[-\infty, \infty]$

Nous pouvons apercevoir que lorsque les intervalles d'entrées sont larges, les résultats obtenus par l'arithmétique affine ne sont pas meilleurs que ceux obtenus par l'arithmétique d'intervalle. Même si cette méthode a certains avantages quand la dépendance est simple, mais la librairie actuelle ne peut pas nous donner un résultat prometteur.

2.2.8 Librairie d'intervalle

Il y a nombreux outils gratuits disponible sur Internet à télécharger ou par commande. Leur vitesse et environnement du système d'opérations compatible varie, y compris les précisions et la méthode d'erreur arrondie. Dans cette section nous allons parler des librairies d'intervalles et donner une comparaison préliminaire de ces outils afin de donner l'outil de base pour l'intégrateur numérique garantie dans les sections qui suivent.

Dans le page de l'université de Texas at El Paso (<http://www.cs.utep.edu/interval-comp/>), un catalogue complet de librairies d'intervalle est mis à jour, il résume tous les logiciels et les applications relatives avec les intervalles. La plupart des logiciels sont écrits sous C, C++ ou Matlab. Nous allons présenter quelques librairies qui nous intéressent.

C/C++ Les librairies le plus utilisées par le monde sont écrites sous C/C++, par exemple Profil/Bias [Keil 2009], Gaol [Gaol 2013], Filib++ [Lerch 2006], Boost interval. Elles sont toutes gratuites pour une utilisation non commerciale.

Profil/Bias utilise le type 'double' pour désigner les deux bornes durant son calcul. Filib++, Gaol, Boost appliquent le type 'long double', mais cela nécessite le support de compilateur sur place. Les autres différences entre ces librairies sont le traitement d'erreur arrondie, la présentation des valeurs infinies et la flexibilité de changer le type de variable au nombre entier ou flottant selon chaque configuration.

Boost et Gaol sont entretenues à jour, les modifications personnelles sont faciles à mettre en place, la technique utilisée pour les arrondis est robuste sans prendre de précaution spécifique sur un ordinateur différent. Ces deux librairies sont les plus intéressantes à tester, mais un manque de documentation rend leur utilisation difficile. Comme les solveurs VNODE (un intégrateur garanti, sa version plus récente est VNODE-LP) [Nedialkov 1999a] [Nedialkov 1999b] [Nedialkov 2001] et VSPODE (un

autre intégrateur garanti) [Lin 2007] ont intégré les bibliothèques d'intervalle. Nous reparlerons de ces 2 solveurs dans le chapitre 3.

Profil/Bias est une bibliothèque spécialement optimisée pour les opérations élémentaires d'intervalle grâce à son utilisation du code assemblage. Mais les fonctions mathématiques sont lentes par rapport aux autres bibliothèques, parce qu'un Fast Look up algorithme pouvant raccourcir certains calculs de sinus et cosinus, n'a pas été utilisé. Il faut quand même constater que ces tableaux de look up sont moins précis [Zilinskas 2005] et produisent des surestimations inattendues. Quand nous utilisons non seulement les opérations basiques, Filib++ est le plus rapide et fiable selon une comparaison faite par Zilinskas [Zilinskas 2005]. Dans les sections suivantes, nous appliquons la bibliothèque Filib++ pour les algorithmes par C++ avec VNODE-LP. Pour le VSPODE, il a son propre bibliothèque intervalle.

Matlab Les packages d'intervalle sous Matlab sont appelées INTLAB et b4m. INTLAB est utilisé par la plupart de gens qui travaillent avec le calcul d'intervalle. Malheureusement, INTLAB en version 7 n'est plus gratuit à utiliser. b4m (Bias for Matlab) est basé sur Bias dont nous avons parlé tout à l'heure. L'INTLAB fournit des fonctions d'optimisation et la plupart des fonctions élémentaires, il est favorable de réaliser quelque code de test afin de comparer les performances de différents algorithmes. Nous allons utiliser l'INTLAB pour réaliser certains algorithmes afin de rendre une image complète sur les avancements récents dans le domaine analyse par intervalle.

Conclusion La bibliothèque en C++ sera appliquée dans cette thèse parce que les autres solveurs pour l'intégration garantie sont écrits sous C++. La bibliothèque FILIB++ sera utilisée dans la suite, VNODE sera compilé avec cette bibliothèque. Certains algorithmes sont écrits sous INTLAB en Version 6 de Matlab pour le but de comparer certains algorithmes.

2.2.9 Image réciproque d'un ensemble

Pour calculer l'image réciproque d'un ensemble, il est nécessaire de résoudre un problème d'inversion ensembliste en caractérisant l'ensemble \mathbb{X} défini par :

$$\mathbb{X} = \{x \in \mathbb{R}^n \mid f(x) \in \mathbb{Y}\} = f^{-1}(\mathbb{Y}) \quad (2.74)$$

où f est une fonction connue continue de \mathbb{R}^n dans \mathbb{R}^m ayant une fonction d'inclusion isotonique $[f](.)$ et où \mathbb{Y} est un ensemble connu inclus dans \mathbb{R}^m .

Ce problème d'inversion ensembliste peut être résolu de manière approximative mais garantie par l'analyse par intervalle et l'algorithme SIVIA (Set Inversion Via Interval Analysis) proposé dans [Jaulin 1993]. Cet algorithme permet la construction de deux sous-pavages $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$, un encadrement intérieur et un encadrement extérieur de l'ensemble solution \mathbb{X} qui vérifient les inclusions suivantes :

$$\underline{\mathbb{X}} \subset \mathbb{X} \subset \overline{\mathbb{X}} \quad (2.75)$$

L'algorithme SIVIA, qui permet d'obtenir les deux sous-pavages précédents avec une précision prédéfinie ε , se compose essentiellement de trois étapes :

- le choix d'un pavé initial \mathbb{X}_0 supposé contenir toutes les solutions recherchées.
- les tests d'inclusion
- la bisection

Tests d'inclusion Un test d'inclusion consiste à vérifier l'appartenance d'un intervalle obtenu par une fonction d'inclusion à un ensemble connu a priori. Pour un pavé quelconque $[x]$, 3 situations sont envisageables :

$$[f]([x]) \subset \mathbb{Y} \rightarrow [x] \subset \mathbb{X} \text{ et } [x] \text{ est dit admissible}$$

$$[f]([x]) \cap \mathbb{Y} = \emptyset \rightarrow [x] \cap \mathbb{X} \text{ et } [x] \text{ est dit mauvais}$$

$$[f]([x]) \cap \mathbb{Y} \rightarrow [x] \neq \emptyset \text{ et } [x] \text{ est dit indéterminé}$$

Bisection Bissecter un pavé signifie couper le pavé en deux selon l'axe de son plus grand côté. Lorsque qu'un pavé est dit incertain (ou encore indéterminé), une bisection est réalisée sur celui-ci. les deux nouveaux pavés générés sont à leur tour testés. Si ceux-ci sont à nouveau incertains, nous les bissectons. Nous recommençons cette opération jusqu'à ce qu'il ne reste plus de pavé incertain ou jusqu'à ce que les pavés restant à analyser aient un diamètre inférieur à la précision définie ε que nous l'aurons fixé afin d'arrêter l'algorithme. La solution de notre problème d'inversion ensembliste sera donnée par l'union des pavés admissibles et des pavés incertains jugés trop petits.

Nous utilisons l'algorithme SIVIA pour déterminer les sous-ensembles $\underline{\mathbb{X}}$ contenant les pavés solutions et $\overline{\mathbb{X}}$ contenant les pavés admissibles et les pavés indéterminés de petite taille. Cet algorithme utilise deux piles :

- Une pile \mathcal{P}_{int} dans laquelle sont rangés les intervalles à analyser.
- Une pile \mathcal{P}_{ad} dans laquelle sont rangés les intervalles acceptables ou incertains mais de petite taille.

La pavé initial $[x](0) \in \mathbb{X}_0$ supposé contenir toutes les solutions recherchées est placé dans la pile \mathcal{P} . Lors du déroulement de l'algorithme, le premier élément de la pile est sorti dans un pavé intermédiaire $[x]$. Cette opération consiste à extraire l'élément au sommet de la pile. Son image par la fonction d'inclusion $[f]$ est ensuite testée dans l'ensemble connu a priori \mathbb{Y} . Si le pavé $[x]$ est indéterminé et que son diamètre est supérieur à ε , il est bissecté. Les deux pavés générés $[x]_1$ et $[x]_2$ sont ensuite rangés au sommet de la pile \mathcal{P}_{int} . Si le pavé $[x]$ est admissible, la pile \mathcal{P}_{ad} est augmentée de $[x]$ et si $[x]$ est mauvais, il est éliminé.

Algorithme 2 Set Inversion Via Interval Analysis**Input:** $f, [x(0)], \mathbb{Y}, \varepsilon$;**Output:** $\underline{\mathbb{X}}, \overline{\mathbb{X}}$;

```

1: Initialisation :  $\mathcal{P}_{ad} := \emptyset, \mathcal{P}_{int} := [x](0)$ ;
2: while  $\mathcal{P}_{int} \neq \emptyset$  do
3:    $[x] := \text{pop } \mathcal{P}_{int}$ ;
4:   if  $[f]([x]) \subset \mathbb{Y}$  then
5:      $\mathcal{P}_{ad} = \mathcal{P}_{ad} \cup [x]$ 
6:   if  $[f]([x]) \cap \mathbb{Y} \neq \emptyset$  et  $\omega([x]) < \varepsilon$  then
7:      $\mathcal{P}_{ad} := \mathcal{P}_{ad} \cup [x]$ 
8:   if  $[f]([x]) \cap \mathbb{Y} \neq \emptyset$  et  $\omega([x]) > \varepsilon$  then
9:     bisect  $[x]$ ,  $\mathcal{P}_{int} := \mathcal{P}_{int} \cup [x]_1, \mathcal{P}_{int} := \mathcal{P}_{int} \cup [x]_2$ 

```

2.2.10 Image directe d'un ensemble

Nous introduisons la détermination de l'image directe d'un ensemble. Ce qui est considéré également plus difficile que celle de l'image réciproque d'un ensemble.

En effet, cela consiste à caractériser l'ensemble d'approximation extérieure de \mathbb{Y}

$$\mathbb{Y} = f(\mathbb{X}) \quad (2.76)$$

où \mathbb{X} est un ensemble connu, f est une fonction connue continue et $[f]$ est une fonction d'inclusion convergente de f .

Un algorithme basé sur l'analyse par intervalles et l'inversion ensembliste permet la caractérisation de cet ensemble. L'algorithme *ImageSP* développé par M. Kieffer [Kieffer 1999]. Il construit un sous-pavage régulier $\bar{\mathbb{Y}}$ de \mathbb{Y} qui contient l'image par la fonction d'inclusion $[f]$ d'un sous-pavage régulier \mathbb{X}_ε de \mathbb{X} donc,

$$\begin{aligned} \mathbb{Y} &= f(\mathbb{X}) \cup \bar{\mathbb{Y}} \\ \bar{\mathbb{Y}} &\subset [f](\mathbb{X}) \end{aligned} \quad (2.77)$$

Cet algorithme évolue en trois étapes :

1. Le hachage consiste à construire, à partir de bisections successives des pavés \mathbb{X} , un pavage régulier non minimal \mathbb{X}_ε qui ne contient que des pavés dont la longueur est inférieure à une précision requise ε , avec $\varepsilon > 0$ et suffisamment petit.
2. L'évaluation calcul l'image par la fonction $[f]$ de tous les pavés $[x]$ contenus dans \mathbb{X}_ε . Toutes les boîtes résultantes sont alors stockées dans une liste \mathbb{U} .
3. La régularisation calcule un sous-pavage régulier $\bar{\mathbb{Y}}$ contenant l'ensemble \mathbb{U} , l'union de tous les pavés de la liste \mathbb{U} . La régularisation fait appel à l'algorithme SIVIA qui permet d'inverser l'ensemble \mathbb{U} par la fonction identité. En effet si $f(\mathbb{X}) \subset \mathbb{U}$, alors $f(\mathbb{X}) \subset Id^{-1}\mathbb{U}$, nous nous ramenons donc à un problème d'inversion ensembliste.

L'algorithme *ImageSP* permet donc de calculer un sous-pavage régulier $\bar{\mathbb{Y}}$ garanti de contenir l'ensemble \mathbb{Y} .

Algorithme 3 Direct Image

Input: $f, \mathbb{X}, \varepsilon$;**Output:** $\bar{\mathbb{Y}}$; $\mathbb{X}_\varepsilon := chop(\mathbb{X}, \varepsilon)$ $\mathcal{U} := \emptyset$, \mathcal{U} is a list and \mathbb{U} is all boxes contained in \mathcal{U} For every $[x] \in \mathbb{X}_\varepsilon$, put $[f]([x])$ in the list of \mathcal{U} SIVIA ($[y] \in \mathcal{U}, Id, \varepsilon$)

L'analyse par intervalles permet la prise en compte des incertitudes (de mesure ou autres). Elle donne des résultats garantis en calculant un intervalle contenant à coup sûr la solution du problème considéré. Par contre, nous avons vu que la longueur des intervalles calculés par les méthodes ensemblistes énoncées dans cette section dépend fortement de l'expression initiale de la fonction d'évaluation. C'est un inconvénient de l'analyse par intervalles.

Dans le chapitre suivant, nous allons présenter quelques méthodes d'intégrations garanties basées sur l'analyse par intervalles et les techniques utilisant les développements en série de Taylor. Ces méthodes sont garanties dans le sens où les résultats sont fournis sous forme d'intervalles contiennent à coup sûr la solution du problème traité dans la thèse. Avec les outils de l'analyse par intervalles exposés précédemment, nous allons aborder l'estimation d'état et de paramètres.

Estimation garantie et application aux modèles académiques

Contents

3.1	Estimation dans le contexte ensembliste	35
3.1.1	Estimation d'état	36
3.1.2	Estimation de paramètres	38
3.1.3	Contracteur	41
3.2	Intégration numérique garantie basée sur le développement en série de Taylor	43
3.2.1	Existence et unicité de la solution	46
3.2.2	Recherche d'une solution a priori	47
3.2.3	Réduction du pessimiste sur la solution	48
3.2.4	Effet d'enveloppement	49
3.2.5	Solveur disponible pour l'intégration garantie	52
3.3	Intégration numérique garantie basée sur le théorème de Müller	53
3.4	Application dans le domaine pharmaceutique	54
3.4.1	Estimation d'état	57
3.4.2	Estimation paramétrique	58
3.5	Application dans le domaine pharmacocinétique	64
3.6	Application dans le domaine aéronautique	67
3.6.1	Modèle du vol longitudinale d'un avion en atmosphère au repos	68
3.6.2	Estimation d'état du modèle	69
3.6.3	Estimation de paramètres du modèle	71

Aujourd'hui, la plupart des processus ou des systèmes peut être considéré comme suivant certaines lois de fonctionnement, ce qui introduit une augmentation de la demande en ce qui concerne la modélisation mathématique. D'une part, la simulation s'effectue sur ces modèles mathématiques ce qui est moins coûteux en terme de matériaux et de temps, d'autre part cela demande trop d'expériences ce qui est parfois trop coûteux et parfois dangereux dans certains domaines d'applications.

La modélisation a pour but la conception d'un modèle. Ce modèle est souvent donné par un ensemble d'équations devant représenter le comportement du système de façon plus concrète. Nous espérons que le modèle du système construit puisse être appliqué aux diverses circonstances : le modèle pour un dosage de médicament peut correspondre à la mesure du glucose, le modèle d'un planeur peut être facilement associé au modèle de l'avion réel, en modifiant quelques grandeurs de paramètres. La question qui se pose

est la suivante : comment obtenir les états du fonctionnement, comment pouvons nous calculer ces paramètres afin d'obtenir les bons paramètres pour le modèle envisagé ?

Pour des systèmes non linéaires, la sortie du système et l'entrée du système reflètent les propriétés externes du système, mais le fonctionnement du système dépend des états internes qui ne sont souvent pas accessibles d'une façon directe. Dans ce cas, nous avons recours à l'estimation de l'évolution de l'état à partir des mesures de l'entrée et de la sortie du système. Une estimation d'état à la base du modèle est nécessaire afin de connaître les conditions de fonctionnement pour un système étudié. Généralement, l'estimation peut se faire avec le filtre de Kalman étendu [Lewis 1986], des linéarisations doivent être prises en compte pour des systèmes non linéaires, une approximation polynômiale avec la formule de l'interpolation de Stirling [Magnus 2000], etc. Ces techniques demandent des connaissances sur des distributions statistiques des perturbations. Dans certaines applications, le bruit ne peut pas être décrit par une loi de probabilité (nous n'avons pas suffisamment de données par exemple). L'approche de l'estimation dans le contexte à erreurs bornées fournit une alternative intéressante aux méthodes statistiques. Le problème est de caractériser, d'une manière garantie, toutes les valeurs du vecteur d'état compatibles avec les mesures aux instants donnés et avec les bornes d'erreurs connues a priori. Nous citons quelques travaux dédiés aux modèles non linéaires : [Jaulin 1993], [Kieffer 1998], [Kieffer 2003], [Kieffer 2005]. Nous évoquerons la dernière approche dans la suite de ce travail.

Il faut que nous soyons capable d'observer les sorties du système réel, soit par des expériences soit par des essais. Avec les mesures obtenues, nous pouvons estimer les futures sorties du modèle avec une estimation par exemple de moindres carrées ou bien de l'estimation Bayésienne ou de vraisemblance. Avec l'outil de l'analyse par intervalles, nous pouvons considérer que toutes les variables sont incertaines et entourées, chacune par deux bornes. Cela nous permet de calculer directement les estimations à partir d'un système dynamique avec un solveur correspondant. Dans nos travaux, le système est décrit par un ensemble d'équations différentielles ordinaires avec la valeur initiale donnée. Les outils disponibles sont par exemple VNODE ou VSPODE, ce qui nous permet de calculer aisément les sorties du modèle dynamique qui nous intéressent via une approche ensembliste. Par la suite, les sorties mesurées sur le système sont notées $y(t_i)$, les t_i représentent les paramètres de l'expérience (instant de mesure, positions du capteur, etc.) ; Les sorties prévues par le modèle sont $y^m(t_i, p)$, où p est le vecteur des paramètres. Enfin, la différence entre les sorties mesurées et les sorties données par le modèle est notée $e(t_i, p)$ avec

$$e(t_i, p) = y(t_i) - y^m(t_i, p). \quad (3.1)$$

Cette erreur provient d'une part de l'erreur structurelle et d'autre part des erreurs de mesure, par exemple la limite de l'appareil de mesure, les perturbations, etc. Dans ce travail, les modèles considérés sont tous non-linéaires avec des paramètres constants, de la forme :

$$y^m(t_i, p) = g(t_i, p), \quad (3.2)$$

où $g(t_i, p)$ est une fonction quelconque, dont la forme est également connue pour tous t_i .

Le problème qui nous intéresse est l'estimation de paramètres faite par l'analyse d'intervalle. Il y a deux manières possibles pour estimer les paramètres en considérant ce problème comme un problème inverse, la recherche s'effectue souvent par la minimisation d'une fonction coût en faisant varier le vecteur des paramètres. Les paramètres retenus sont dit paramètres optimaux, mais uniquement au sens du critère appliqué, le vecteur de paramètres peut varier suivant la fonction coût choisie. Cette approche est dite aussi l'approche par optimisation.

Une autre approche de recherche les paramètres admissibles en vérifiant certains critères dans un sens déterministe, avec des erreurs de sortie ou de mesure, supposées appartenir à des intervalles connus. Le résultat de l'estimation des paramètres n'est alors plus un élément ponctuel de l'espace des paramètres, mais il est constitué d'un ensemble de composantes qui sont ici des intervalles.

Le choix de la méthode à employer pour la recherche des paramètres sera en partie dicté par les hypothèses faites sur l'erreur de sortie, mais également par l'usage qui sera faite par les résultats estimés obtenus. Par exemple, lorsqu'une valeur ponctuelle suffit, les méthodes par minimisation seront envisageables. Lorsqu'il faudra garantir une propriété pour toutes les valeurs possibles des paramètres, disposer de l'ensemble des valeurs compatibles avec les mesures s'avère très utile.

Dans la suite de ce chapitre, nous aborderons ces deux méthodes.

3.1 Estimation dans le contexte ensembliste

Avec les outils présentés dans les chapitres précédents, l'estimation pour les systèmes dynamiques dans le contexte ensembliste via l'analyse par intervalle est faisable. Il y a différentes méthodes pour avoir les ensembles atteignables ou les bassins de capture, c'est-à-dire l'ensemble des points pour lesquels il existe une trajectoire du système les visitant [Bokanowski 2010], l'intégration garantie en est une [Nedialkov 2001]. Dans cette thèse, nous utilisons l'intégration garantie pour trouver les ensembles atteignables.

Considérons le système décrit par les équations d'état suivantes :

$$\begin{cases} \dot{x}(t) &= f(x(t), p), \\ y^m(t) &= g(x(t), p), \\ x(t_0) &\in [X_0], \\ p &\in [P_0], \end{cases} \quad (3.3)$$

où $x(t) \in \mathbb{R}^{n_x}$ et $y^m(t) \in \mathbb{R}^{n_y}$ représentent respectivement les variables d'états et les sorties du modèle. Les fonctions f et g sont non linéaires et différentiables par rapport au x et p , $[X_0]$ est le domaine a priori connu pour la condition initiale $x(t_0)$ et $[P_0]$ est le domaine initial pour le vecteur de paramètre $p \in \mathbb{R}^{n_p}$.

Les sorties mesurées $y(t_j)$ sont supposées disponibles à chaque instant $t_j \in \{t_1, \dots, t_N\}$. Dans un contexte à erreurs bornées, il existe un domaine admissible $[Y_j]$, à chaque instant de mesure t_j , pour le vecteur des sorties $y(t_j)$:

$$\forall t_j \in \{t_1, \dots, t_N\}, \quad y^m(t_j) = g(x(t_j), p) \in [Y_j] \quad (3.4)$$

Le vecteur des sorties $[y](t_j) = ([y_1](t_j), [y_2](t_j), \dots, [y_{n_y}](t_j))$. De même, il existe un domaine admissible $[e]$ défini pour l'erreur entre les sorties mesurées $y(t_j)$ et les sorties du modèle $y^m(t_j)$.

Donc, il est possible de définir un intervalle $[y](t_i)$ satisfaisant :

$$y^m(t_i, p) \in [y](t_i) = [\underline{y}(t_i), \overline{y}(t_i)], \quad (3.5)$$

et tel que $[y](t_i)$ contienne de manière garantie (selon notre exigence) la valeur de la sortie à l'instant t_i même si l'expérience est répétée.

3.1.1 Estimation d'état

Nous utilisons la méthode d'intégration numérique garantie, pour évaluer la sortie du modèle et le vecteur d'état aux instants de mesure $t_j \in \{t_1, \dots, t_N\}$ de manière à ce que la relation (3.4) soit vérifiée. Pour résoudre le problème d'estimation d'état, le vecteur de paramètres p est considéré connu.

Lorsque l'état initial $[x_0]$ est relativement bien connu, c'est-à-dire que sa longueur est faible (jusqu'à 5% de la valeur exacte), un estimateur causal peut être mis en œuvre. Pour réaliser une estimation en ligne de l'état, nous utilisons une méthode qui se compose de deux phases :

- une phase de prédiction qui calcule l'ensemble atteignable pour l'état $[x_{j+1}^+]$.
- une phase de correction qui élimine toutes les parties incohérentes de l'état prédit $[x_{j+1}^+]$, avec le domaine admissible $[Y_j]$ pour la sortie du modèle $[y_{j+1}^m]$.

Les phases de prédiction et de correction sont illustrées dans un cas simple sur la Figure 3.1 :

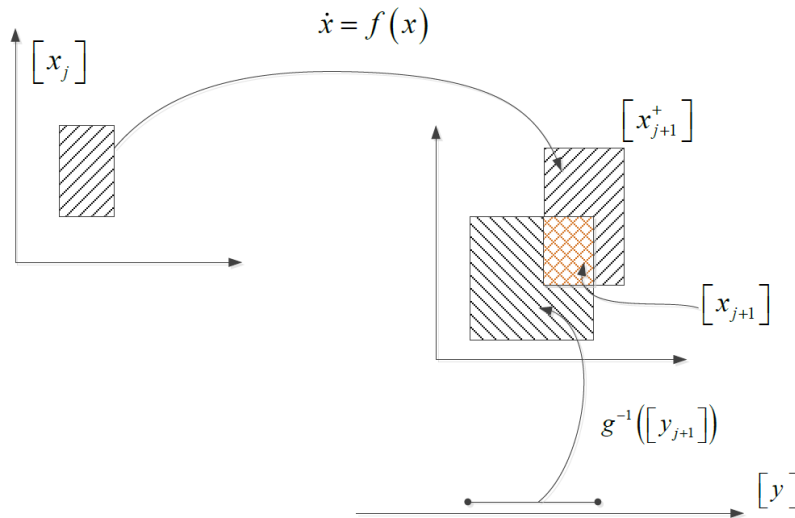


FIGURE 3.1 – L'estimation d'état : la phase de prédiction et la phase de correction

Pour calculer l'ensemble atteignable pour l'état dans l'étape de prédiction, nous utilisons les résultats obtenus à l'aide d'un intégrateur garanti, soit VNODE-LP

[Nedialkov 2001] ou bien VSPODE [Lin 2007], ACADO toolkit [Houska 2013] par exemple. L'ensemble de ces solveurs est appelé ODE dans la suite. Sans précision spécifique, nous utilisons VNODE-LP comme intégrateur garanti pour l'analyse par intervalle.

L'ensemble admissible y_{j+1} cohérent avec les sorties du modèle est noté $[y_{j+1}]$. L'algorithme pour une estimation d'état est le suivant :

Algorithme 4 Causal Estimation ($[x]$)

Input: $[x](0)$, $[p_0]$, $[y]$;

Output: $[x]$;

- 1: initialization : $[x]_e(0) := ([x](0), [p_0])$;
 - 2: $i := 1$;
 - 3: **while** $i \leq N$ **do**
 - 4: $[x]_e^+(i) := ODE([x]_e(i-1))$;
 - 5: $j := i$;
 - 6: $[x]_e(i) = [x]_e^+(i) \cap [g]^{-1}([y_j])$
 - 7: $i := i + 1$;
-

Pour des applications réelles, une modification sur la fréquence de l'intersection entre $[x]_e^+(i)$ et $g^{-1}([y_j])$ est possible. Le mode choisi, par exemple, ici, nous prenons $j = i$, est préliminaire. Pour des systèmes qui souffrent de pessimisme pendant le calcul, nous pouvons aussi choisir un critère de démarrage pour vérifier les états estimés.

Souvent, les résultats capturés par les capteurs sont plus précis que les états observés par le modèle. Une relation assez simple peut être utilisée pour contrôler le moment de démarrage pour l'étape de correction dans l'algorithme 4. Nous proposons la condition qui suit.

Proposition 2. *Une condition de démarrage pour l'étape de correction peut être appliquée pour que le système soit plus robuste :*

$$\omega([y^m](t_i, [x_i], [p])) < \omega([E](t_i)) \quad (3.6)$$

où $[E](t_i)$ représente l'erreur admissible sur les valeurs accessibles à l'instant t_i .

Algorithme 5 Robust Causal Estimation ($[x]$)

Input: $[x](0)$, $[p_0]$, $[y]$;

Output: $[x]$;

- 1: initialization : $[x]_e(0) := ([x](0), [p_0])$;
 - 2: $i := 1$;
 - 3: **while** $i \leq N$ **do**
 - 4: $[x]_e^+(i) := ODE([x]_e(i-1))$;
 - 5: $j := 1$
 - 6: **if** $\omega([g]([x]_e(i))) > \omega([E](i))$ **then**
 - 7: $[x]_e(i) = [x]_e^+(i) \cap [g]^{-1}([y_j])$
 - 8: $i := i + 1$;
-

Pour le cas avec plusieurs sorties, cette condition est évaluée composante par composante. La convergence de l'estimateur causal robuste est évidente, la valeur absolue sur la différence entre les états estimés et celui des inversions mesures est asymptotiquement nulle, ceci est dû à l'étape 6 de l'algorithme 5.

3.1.2 Estimation de paramètres

Identifier les paramètres d'un modèle sur un système ayant les sorties mesurées $[y](t_i)$ revient à trouver l'ensemble des valeurs des p satisfaisant :

$$y^m(t_i, p) \in [y](t_i) \quad \text{pour } i = 1, \dots, n. \quad (3.7)$$

L'erreur entre les mesures de la sortie et la sortie du modèle est alors incluse :

$$e(t_i, p) = y(t_i) - y^m(t_i, p) \in [y](t_i) - y^m(t_i, p) \quad (3.8)$$

L'estimation de paramètres dans un contexte à erreurs bornées peut se poser par deux approches différentes. L'une est celle que nous l'avons rencontrée souvent dans le cas stochastique : l'optimisation. Nous utilisons une fonction coût pour évaluer les paramètres qui minimisent ou maximisent cette fonction de la différence entre la sortie réelle et la sortie du modèle avec une norme i qui exploite :

$$c = \|y - y^m\|_i \quad i = 1, 2, \infty, \text{ etc}, \quad (3.9)$$

sachant que $\|[v]\|_2 = (\sum_{i=1}^n \text{mag}([v_i])^2)^{1/2}$ et $\|[v_i]\|_1 = \sum_{i=1}^n \text{mag}([v_i])$.

Dans l'analyse par intervalle, nous prenons l'exemple de la norme $i = 2$, la fonction c sur tous les instants t_i de mesures donnés, elle peut écrire comme :

$$c = \sum_{i=1}^N |[y](t_i) - [y^m](t_i, [p])|^2 = \sum_{i=1}^N |[e](t_i)|^2. \quad (3.10)$$

Nous cherchons $[p] \subset [P_0]$ tel que (3.10) soit minimale. Ci-dessous, nous donnons l'algorithme suivant :

Algorithme 6 Parameter estimation ($[h], [y], \mathcal{P}_{admis}, \varepsilon$)**Input:** $[c], \varepsilon, [x_0]$;**Output:** $Min(\overline{c}([x])), \mathcal{C}, \mathcal{L}$;

```

1:  $c_{bs} := f(m([x_0])), [x] := [x_0]$ ;
2: while  $\mathcal{L} \neq \emptyset$  do
3:   bisection follows the maximal width of  $[x]$ ,  $[x] = [x_1] \cup [x_2]$ ;
4:    $c_{bs} := \min\{c_{bs}, \overline{c}(m([x_1])), \overline{c}(m([x_2]))\}$ ;
5:   if  $\max\{\overline{c}([x_1]), \overline{c}([x_2])\} - \min\{c([x_1]), c([x_2])\} < \varepsilon$  then
6:     Put  $[x_1]$  and  $[x_2]$  in the list of  $\mathcal{C}$  with increasing order
7:     if  $\mathcal{L}$  n'est pas vide then
8:        $[x] := ExtraireSommet(\mathcal{L})$ 
9:       if  $\overline{c}([x]) > c_{bs}$  then
10:        return the lower bound on the head of  $\mathcal{C}$  and the lists  $\mathcal{C}$  and  $\mathcal{L}$ 
11:       else
12:        return the lower bound on the head of  $\mathcal{C}$  and the list  $\mathcal{C}$ 
13:     else
14:       put  $([x_1], \overline{c}([x_1]))$  and  $([x_2], \overline{c}([x_2]))$  on head in increasing order in the list of  $\mathcal{L}$ 
15:        $[x] := pop(\mathcal{L})$ 

```

où la fonction d'inclusion $[c]$ est calculée par l'arithmétique d'intervalle, les listes \mathcal{C} et \mathcal{L} enregistrent à la fois l'intervalle $[x]$ et la borne supérieure $\overline{f}([x])$. Si la fonction coût $[c]$ a pris une forme de moindres carrées, alors ε devient une variable difficile à définir parce que l'écart entre les sorties du modèle $[y^m]$ et les mesures seront inconnues, l'incertitude introduite par $[x]$ rend difficile le choix de ε . Une autre stratégie possible est de rajouter un compteur d'itération afin de terminer l'algorithme dans une période finie. Dans ce cas, le choix du ε devient indépendant de la longueur de $[x]$, il représente la précision requise pour trouver les bons paramètres.

Notons qu'entre les lignes 3 et 4 de l'algorithme 6, il est possible de rajouter un contracteur [Kieffer 2011] afin d'éliminer les $[x]$ inconsistants. Cette technique peut rendre le processus d'estimation plus rapide et peut être le seul choix pour contourner la difficulté NP dans l'estimation de paramètres en utilisant l'analyse par intervalles.

Avec l'algorithme 6, nous trouvons souvent au final un vecteur d'intervalle optimal, ce qui ne nous donne pas de possibilité de voir les autres intervalles qui sont aussi proches de la valeur minimale de c , alors nous nous intéressons aussi à une autre approche déterministe faite par l'inversion ensembliste.

Cette approche est faite en vérifiant les consistances entre les sorties du modèle et les sorties du système en cherchant l'ensemble S des paramètres satisfaisant :

$$S = \{p \in \mathcal{P} \mid y^m(t_i, p) \in [y](t_i), i = 1, \dots, n\} = \{p \in \mathcal{P} \mid [y^m]([p]) \subset [y]\}, \quad (3.11)$$

où \mathcal{P} est l'ensemble de paramètres admissibles cohérents avec les mesures y . Nous appelons S l'ensemble des solutions qui contient l'ensemble de ce type de paramètres. Cet ensemble peut être obtenu de manière garantie à partir de l'erreur $e(t_i, p)$ (voir équation 3.8) :

$$S = \{p \in \mathcal{P} \mid e(t_i, p) \in [e](t_i), i = 1, \dots, n\} = \{p \in \mathcal{P} \mid [e]([p]) \subset [e]\}. \quad (3.12)$$

Nous cherchons à estimer le vecteur de paramètres p , initialement compris dans un domaine de recherche a priori $[P_0]$, tel que les sorties prédites du modèle $[y^m](t_i, [p])$ soient cohérentes avec les mesures expérimentales $[y](t_i)$ et les bornes d'erreur acceptables.

Pour résoudre ce problème, nous pouvons utiliser l'algorithme SIVIA, qui estime les paramètres par l'inversion ensembliste. Elle consiste à trouver l'ensemble image réciproque de l'ensemble d'erreur admissible $[e]$ par la fonction d'inclusion connue $[e]([p])$. Ce problème d'inversion ensembliste peut être résolu de manière approchée, mais avec une précision arbitraire par l'algorithme SIVIA qui est présenté au chapitre 2.

Cet algorithme nécessite trois étapes :

- le choix du pavé initial $[P_0]$ qui est censé contenir toutes les valeurs du vecteur de paramètres recherché,
- les tests d'inclusion qui vérifient que l'erreur entre les sorties mesurées $[y](t_i)$ et les sorties du modèle $[y^m](t_i, [p])$ calculées par l'algorithme de la valeur moyenne étendue est un vecteur intervalle appartenant au domaine d'erreur admissible $[e]$ dont les bornes \underline{e}_{min} et \bar{e}_{max} sont connues,
- la bisection du vecteur de paramètres $[p]$ sera effectuée lorsque aucune conclusion peut être faite.

Le vecteur de paramètres sera obtenu sous forme d'intervalles dont la longueur $\omega([p])$ sera inférieure ou égale à la précision requise ε fixée par l'utilisateur. Cette précision de découpage ε arrête les bisections du vecteur de paramètres.

Si S est trouvé vide, cela signifie qu'il n'est pas possible d'encadrer des valeurs \mathcal{P} satisfaisant l'équation (3.7) pour tous t_i , ou cela peut être à cause du découpage qui n'est pas suffisamment petit. Cette méthode ne demande pas de contrainte sur l'ensemble cherché (convexe, non convexe, forme arbitraire), la simplicité d'utilisation l'a rendu populaire.

Dans la suite, nous avons intégré le processus d'estimation d'état dans l'estimation de paramètres. Et afin d'éviter le pessimisme durant le calcul de l'EDO, nous proposons d'utiliser les mesures jusqu'au j ème élément obtenu à l'étape de vérification de consistance via l'EDO dans l'algorithme 7 [Li 2014] :

Algorithme 7 Parameter estimation ($[h], [y], \mathcal{P}_{admis}, \varepsilon$)

Input: $[x](0), [p](0);$ **Output:** $\mathcal{P}_{admis}, \mathcal{P}_{uncertain}, \mathcal{P}_{rejected};$

```

1: initialization :  $P_{list} := [p](0), [x]_e(0) := ([x](0), [p](0));$ 
2: while  $P_{list} \neq \emptyset$  do
3:    $[p] := Pop(\mathcal{P}_{list});$ 
4:    $i := 1;$ 
5:   while  $i \leq N$  do
6:      $[x]_e(i) := ODE([x]_e(i-1));$ 
7:      $j := i;$ 
8:      $i := i + 1;$ 
9:     if  $[h]([x]_e(1:j)) \subset [y](1:j)$  then
10:       $\mathcal{P}_{admis} := \mathcal{P}_{admis} \cup [p];$ 
11:     else if  $[h]([x]_e(1:j)) \cap [y](1:j) = \emptyset$  then
12:       $\mathcal{P}_{rejected} := \mathcal{P}_{rejected} \cup [p];$ 
13:     else if  $w([p]) < \varepsilon$  then
14:       $\mathcal{P}_{uncertain} := \mathcal{P}_{uncertain} \cup [p];$ 
15:     else
16:        $bisectBox([p])$ 
17:        $\rightarrow \{[p]_1, [p]_2 \mid [p]_1 \cup [p]_2 = [p]\};$ 
18:        $\mathcal{P}_{list} := \mathcal{P}_{list} \cup [p]_1, \mathcal{P}_{list} := \mathcal{P}_{list} \cup [p]_2;$ 

```

Cette vérification de consistance peut aussi être faite par l'algorithme de Forward-Backward avec l'EDO du système. Pour plus de détails, veuillez voir [Tomas 2008].

3.1.3 Contracteur

Supposons que les paramètres à estimer doivent satisfaire une contrainte d'inégalité vectorielle $B(p) \geq 0$. Le contracteur est une fonction C de l'espace \mathbb{R}^n dans l'espace \mathbb{R}^n vérifiant :

$$\begin{aligned} C([p]) &\subseteq [p], \\ \{p \in [p] \mid B(p) \geq 0\} &\subset C([p]). \end{aligned} \tag{3.13}$$

Ces deux équations assurent que premièrement, le domaine $[p]$ est contracté, deuxièmement, il n'y a pas de paramètre consistant avec cette relation qui soit perdu.

Si nous avons une forme analytique de l'équation d'observation, des contracteurs comme Newton, Krawczyk pourront être utilisés [Moore 2009]. Dans la plupart des cas, lorsqu'une forme analytique n'est pas disponible, la forme centrée peut être appliquée dans la plupart des cas non linéaires pour les systèmes dynamiques [Kieffer 2011].

Reprenons la relation fondamentale :

$$y^m(p) \in [y] = [\underline{y}, \bar{y}]. \tag{3.14}$$

La k_{ieme} composante de la sortie $[y^m](x, [p])$ est notée $[y_k^m](x, [p])$, où $[p] \in \mathbb{S} \subset [P_0]$ et $m \in [P_0]$, où m est le point au milieu de $[p]$. Nous pouvons avoir l'équation 3.15 qui suit :

$$y_k^m(m) + \sum_{j=1}^{n_p} ([p_j] - m_j) \left[\frac{\partial y_k^m}{\partial p_j} \right]([p]) \subset [\underline{y}_k, \bar{y}_k], \quad (3.15)$$

pour $k = 1, \dots, n_y$. Le contracteur peut être constitué par une formule décrite avec l'équation :

$$[p_i]^{new} = [p_i] \cap \left(\frac{[\underline{y}_k, \bar{y}_k] - y_k^m(m) - \sum_{j \neq i} ([p_j] - m_j) \left[\frac{\partial y_k^m}{\partial p_j} \right]([p])}{\left[\frac{\partial y_k^m}{\partial p_i} \right]([p])} + m_i \right), \quad (3.16)$$

pour $i = 1, \dots, n_p$.

Avec l'algorithme intervalle standard, la division est faisable quand le dénominateur ne contient pas de zéro : $0 \notin \left[\frac{\partial y_k^m}{\partial p_i} \right](t, [p])$. Une fois que $0 \in \left[\frac{\partial y_k^m}{\partial p_i} \right](t, [p])$, il n'y aura pas de contraction parce que la division par zéro est impossible. Pour pallier cet inconvénient, l'algorithme d'intervalle étendu doit être appliqué. Quand le dénominateur contient zéro, il sera découpé en deux parties : une partie négative et une partie positive. Cela nous permet de faire le calcul avec le numérateur et puis l'intersection avec l'ancien ensemble $[p]$. Souvent, quand zéro est contenu dans le dénominateur, nous ne pouvons pas diminuer le domaine de paramètres.

Maintenant, il reste à obtenir les dérivées par rapport aux paramètres. Nous définissons la sensibilité d'ordre 1 de x_j par rapport au p_k , nous la notons s_{jk} :

$$s_{jk}(t, p) = \frac{\partial x_j}{\partial p_k}(t, p). \quad (3.17)$$

Considérons le système d'observation complet, donné par :

$$y(t, p) = x(t, p). \quad (3.18)$$

Pour calculer s_{jk} , nous utilisons la formule :

$$s'_{jk} = \sum_{i=1}^{n_x} \frac{\partial f_j(x, p)}{\partial x_i} s_{ik} + \frac{\partial f_j(x, p)}{\partial p_k}. \quad (3.19)$$

Comme les états initiaux x_0 sont connus, la condition initiale pour la sensibilité est :

$$s_{jk}(t_0) = \frac{\partial x(t_0)}{\partial p_k} = 0. \quad (3.20)$$

Ensuite nous devons combiner les équations de sensibilités avec l'équation d'état ce qui construit un système d'état sous forme étendue. Les valeurs initiales sont données précédemment.

Le contracteur sert à éliminer les domaines rejetables avec des calculs supplémentaires sur les sensibilités. Une utilisation abusive ralentit fortement cet avantage pendant le processus d'inversion ensembliste, car le temps consacré à l'élimination ne peut être compensé

3.2. Intégration numérique garantie basée sur le développement en série de Taylor 143

par l'effet donné par le contracteur. Une modération d'utilisation ou une utilisation qu'en cas de nécessité doit être respectée afin de minimiser le temps de calcul. Ici, nous proposons d'utiliser un indicateur volumique afin d'éviter des calculs de sensibilité inutiles [Li 2015d] :

$$\prod_{i=1}^{n_p} \text{rad}([p_i]) > \prod_{i=1}^{n_p} (C_i \times \varepsilon), \quad (3.21)$$

où C_i est une constante, que nous prenons en général supérieure à 2. La contribution du contracteur est d'éliminer le plus possible d'ensembles inconsistants. Lorsque nous arrivons à obtenir les ensembles admissibles, les ensembles incertains entourent. Dans cette phase, il n'est pas efficace d'utiliser le contracteur comme qu'il y a pas beaucoup d'ensembles rejetables. Ceci explique l'utilisation de ε dans le deuxième membre de droite de l'inéquation (3.21). Cette condition peut être exprimée à l'algorithme 8 :

Algorithme 8 A volumetric starting criterion for contractor $([p], C, \varepsilon)$

Input: $[x](0), [p](0)$;

Output: true, false ;

1: $\text{Cond1} := \prod_{i=1}^{n_p} \text{rad}([p_i])$;

2: $\text{Cond2} := \prod_{i=1}^{n_p} C_i \varepsilon_i$;

3: **if** $\text{Cond1} > \text{Cond2}$ **then**

4: $\text{CondContract} := \text{true}$;

5: **else**

6: $\text{CondContract} := \text{false}$;

Avec cette critères, nous proposons l'algorithme suivant afin d'équilibrer le problème entre la précision de découpage et l'efficacité du temps quand nous appliquons un contracteur, voir l'algorithme 9.

3.2 Intégration numérique garantie basée sur le développement en série de Taylor

Considérons un système dynamique autonome non linéaire de la forme (3.3), les conditions étant les mêmes que dans le paragraphe (3.1).

La condition initiale $x(t_0)$ est inconnue mais elle est supposée appartenir à un domaine connu a priori \mathbb{X}_0 . Les sorties $y(t)$ sont calculées aux instants $t_i \in \{t_1, \dots, t_N\} \subseteq [t_0, T]$ et sont notées $[y_j]$ pour $j = 1, \dots, N$. Dans la suite de ce chapitre 3, pour simplifier les écritures, l'indexation pour l'état ou l'observation par j ou i représentera le temps. Dans le cas contraire cela sera mentionné.

Remarque 1. *Nous ne considérons que les systèmes autonomes, mais ce choix n'est pas restrictif. En effet, un système non autonome :*

$$\dot{x}(t) = f(x(t), t), x(t_0) \in [x_0], \quad (3.22)$$

Algorithme 9 Parameter estimation $([h], [y], \mathcal{P}_{admis}, C, \varepsilon)$

Input: $[x](0), [p](0);$

Output: $\mathcal{P}_{admis}, \mathcal{P}_{uncertain}, \mathcal{P}_{rejected}, C;$

```

1: initialization :  $P_{list} := [p](0), [x]_e(0) := ([x](0), [p](0));$ 
2: while  $P_{list} \neq \emptyset$  do
3:    $[p] := Pop(\mathcal{P}_{list});$ 
4:    $i := 1;$ 
5:    $CondContract := Contract([p], C, \varepsilon);$ 
6:   if  $CondContract = \text{true}$  then
7:      $Contract([p]);$ 
8:     while  $i \leq N$  do
9:        $[x]_e(i) := ODE([x]_e(i-1));$ 
10:       $j := i;$ 
11:       $i := i + 1;$ 
12:      if  $[h]([x]_e(1:j)) \subset [y](1:j)$  then
13:         $\mathcal{P}_{admis} := \mathcal{P}_{admis} \cup [p];$ 
14:      else if  $[h]([x]_e(1:j)) \cap [y](1:j) := \emptyset$  then
15:         $\mathcal{P}_{rejected} := \mathcal{P}_{rejected} \cup [p];$ 
16:      else if  $w([p]) < \varepsilon$  then
17:         $\mathcal{P}_{uncertain} := \mathcal{P}_{uncertain} \cup [p];$ 
18:      else
19:         $bisectBox([p])$ 
20:         $\rightarrow \{[p]_1, [p]_2 \mid [p]_1 \cup [p]_2 = [p]\};$ 
21:         $\mathcal{P}_{list} := \mathcal{P}_{list} \cup [p]_1, \mathcal{P}_{list} := \mathcal{P}_{list} \cup [p]_2;$ 

```

peut être transformé en un système autonome en construisant un vecteur d'état étendu :

$$x^e = [x, t]^T \text{ avec } \dot{x}_{n+1}^e = 1, \text{ et } x_{n+1}^e(t_0) = t_0, \quad (3.23)$$

où n est la dimension du vecteur d'état du système non autonome.

Remarque 2. Dans le cas d'un système qui dépend d'un vecteur de paramètres inconnu p que nous cherchons à estimer :

$$\dot{x}(t, p) = f(x(t), p), \quad x(t_0) \in [x_0], \quad p \in \mathbb{P}, \quad (3.24)$$

nous pouvons aussi considérer un vecteur d'état étendu

$$x^e = [x, p]^T \text{ avec } \dot{x}_{n+i}^e = 0, \text{ et } x_{n+i}^e(t_0) \in \mathbb{P}_i, \text{ avec } i = 1, \dots, n_p, \quad (3.25)$$

où n_p est la dimension du vecteur de paramètres à estimer.

Nous souhaitons calculer les vecteurs intervalles $[y_j]$ pour $j = 1, \dots, N$ qui garantissent contenir la solution des équations d'état 3.3 à chaque instant d'intégration $t_i \in \{t_1, \dots, t_N\} \subseteq [t_0, T]$.

Naturellement, deux méthodes classiques pouvant être appliquées (pour des comparaisons détaillées, voir [Sandretto 2015]), une méthode appelée Euler, une autre méthode appelée Taylor. Ici, nous ne traitons que la méthode des séries de Taylor pour intégrer de manière garantie l'équation différentielle ordinaire :

$$\dot{x}(t) = f(x(t)), \quad x(t_0) \in [x_0], \quad (3.26)$$

où la solution $x(t_j)$ à l'instant t_j sera notée x_j .

Le développement en série de Taylor à l'ordre k pour un point solution x_{j+1} à l'instant t_{j+1} connaissant la solution x_j à l'instant t_j s'écrit :

$$x_{j+1} = x_j + \sum_{i=1}^{k-1} h^i f^{[i]}(x_j) + h^k f^{[k]}(x(t_\xi)), \quad (3.27)$$

où $x(t_\xi)$ est la solution à $t_\xi \in [t_j, t_{j+1}]$ et $h = t_{j+1} - t_j$ est le pas de discrétisation choisi pour le modèle. Nous l'avons choisi constant pour intégrer notre modèle mais il n'est pas nécessaire qu'il le soit.

Les coefficients de Taylor $f^{[i]}$ sont calculés à partir de la formule récursive suivante :

$$f^{[i]} = \frac{1}{i} \frac{\partial^{(i-1)}}{\partial x} f = \frac{1}{i!} x^{[i]}, \quad i \geq 1. \quad (3.28)$$

Dans notre étude, l'état initial x_j n'est pas parfaitement connu, il est contenant dans un pavé $[x_j]$. Pour avoir une solution garantie, il faut trouver un ensemble contenant $x(t_\xi)$ au lieu d'une approximation ponctuelle pour $x(t_\xi)$. Nous prenons l'ensemble de cette approximation pour toute la trajectoire entre t_j et t_{j+1} , ici $[\xi]$, pouvant être considérée comme une solution a priori pour l'équation 3.3 à l'instant t_{j+1} . La formule du développement de Taylor peut être étendue aux intervalles de la manière suivante :

$$[x_{j+1}] = [x_j] + \sum_{i=1}^{k-1} h^i f^{[i]}([x_j]) + h^k f^{[k]}([\xi]). \quad (3.29)$$

Les fonctions d'inclusion $[f]$ sont seulement notées f .

La méthode d'intégration garantie basée sur les séries de Taylor se décompose en deux étapes [Nedialkov 1999b] :

1. la recherche d'une solution a priori qui nous garantit l'existence et l'unicité de la solution de l'équation différentielle ordinaire
2. la réduction de la solution par des formulations spécifiques des développements en série de Taylor.

3.2.1 Existence et unicité de la solution

Pour prouver l'existence et l'unicité d'une solution, nous utilisons une méthode basée sur le théorème du point fixe de Banach et sur l'opérateur de Picard-Lindelöf que nous définissons.

Théorème 1. *(du point fixe de Banach) Soit \mathbb{X} , un espace métrique complet non vide muni d'une distance d . Si une fonction $\phi : \mathbb{X} \rightarrow \mathbb{X}$ satisfait la condition de Lipschitz :*

$$d(\phi(x), \phi(y)) \leq \lambda d(x, y). \quad (3.30)$$

$\forall x, y \in \mathbb{X}$ et $\lambda \in [0, 1[$, alors la fonction ϕ possède un et un seul point fixe $y^* \in \mathbb{X}$ défini par

$$\phi(y^*) = y^*. \quad (3.31)$$

Définition 7. Soit $u(\cdot)$, une fonction appartenant à \mathbb{U} l'ensemble des fonctions continues définies sur $[t_j, t_{j+1}]$, l'opérateur ϕ sur \mathbb{U} défini par

$$\phi(u(t)) = u(t_j) + \int_{t_j}^t f(u(\tau)) d\tau, \quad (3.32)$$

est appelé opérateur de Picard-Lindelöf. Cet opérateur admet un unique point fixe dans \mathbb{U} . La fonction non linéaire lipschitzienne f est la fonction du système d'équations 3.3 et $t \in [t_j, t_{j+1}]$.

Propriété Soit u^* , le point fixe de l'opérateur de Picard-Lindelöf ϕ , alors :

$$\phi(u^*(t)) = u^*(t) \Leftrightarrow u^*(t) = u^*(t_j) + \int_{t_j}^t f(u^*(\tau)) d\tau. \quad (3.33)$$

En dérivant $u^*(t)$ par rapport au temps, nous obtenons :

$$\frac{d}{dt}(\phi(u^*(t))) = \dot{u}^*(t) \Leftrightarrow \dot{u}^*(t) = f(u^*(t)). \quad (3.34)$$

Il vient d'être démontré que le point fixe (fonction de \mathbb{U}) de ϕ est la solution de l'équation différentielle ordinaire (3.26).

3.2.2 Recherche d'une solution a priori

Pour avoir une solution garantie, il faut trouver l'ensemble $[\xi]$ contenant toutes les trajectoires entre un instant t_j et un instant t_{j+1} . Cet ensemble $[\xi]$ contient l'ensemble des points fixes de ϕ entre t_j et t_{j+1} si et seulement si,

$$\phi([\xi]) \subseteq [\xi]. \quad (3.35)$$

Cet ensemble sera considéré comme une solution a priori de notre équation différentielle à l'instant t_{j+1} .

Proposition 3. Soient $h_j = t_{j+1} - t_j$ le pas de discrétisation de valeur positive et $[\tilde{x}_j] \subseteq \mathbb{R}^{n_x}$ tels que :

$$x_j + [0, h_j]f([\tilde{x}_j]) \subseteq [\tilde{x}_j], \quad (3.36)$$

alors le pavé $[\tilde{x}_j]$ contient toutes les trajectoires $x(t)$ solutions de (3.3) entre t_j et t_{j+1} .

Proposition 4. Si l'état initial x_j n'est pas connu mais appartient à un ensemble $[x_j]$ et si la relation d'inclusion suivante est vérifiée :

$$[x_j] + [0, h_j]f([\tilde{x}_j]) \subseteq [\tilde{x}_j]. \quad (3.37)$$

Alors nous pouvons dire que le pavé $[\tilde{x}_j]$ contient de manière garantie la solution de l'équation différentielle quelque soit la condition initiale $x_j \in [x_j]$, des preuves peuvent être trouvées à [Raïssi 2004a]. Nous rappelons que le pas d'intégration h_j est ici considéré constant. S'il est possible de calculer un pavé $[\tilde{x}_j]$ vérifiant l'inclusion, alors l'équation différentielle (3.3) possède une solution unique à chaque instant de $t \in [t_j, t_{j+1}]$.

Pour calculer le pavé $[\tilde{x}_j]$, il faut commencer par calculer l'expression suivante :

$$[\tilde{x}_j] = [x_j] + [0, h_j]f([x_j]). \quad (3.38)$$

En pratique, cette expression ne contiendra pas la trajectoire solution. L'intervalle \tilde{x}_j est alors agrandi jusqu'à satisfaire la relation d'inclusion (3.36), le pavé \tilde{x}_j sera noté $[\tilde{x}_j]$. Le pavé $[\tilde{x}_j]$ contiendra la solution de $x(t)$ et sera considéré comme une solution a priori de notre équation différentielle ordinaire (3.3).

L'algorithme 10 calcule $[\tilde{x}_j]$ et agrandit l'intervalle si celui-ci ne vérifie pas l'inclusion décrite par l'équation (3.36).

Algorithme 10 Enclosure

Input: $[x_j]$, h , α ;

Output: $[\tilde{x}_j]$;

- 1: initialization : $[\tilde{x}_j] := [x_j] + [0, h]f([x_j])$;
 - 2: **while** $([x_j] + [0, h]f([\tilde{x}_j])) \not\subseteq [\tilde{x}_j]$ **do**
 - 3: $[\tilde{x}_j] := [\tilde{x}_j] + [-\alpha, +\alpha] * |[\tilde{x}_j]|$;
 - 4: $h := h/2$;
-

où $||[\tilde{x}_j]| = \max(|\underline{\tilde{x}}_j|, |\overline{\tilde{x}}_j|)$ représente la norme du vecteur intervalle $[\tilde{x}_j]$, $h_j = h$ où nous considérons que le pas d'intégration est identique tout le temps.

Le coefficient de gonflement α doit être assez faible pour que le pavé $[\tilde{x}_j]$ soit à peine plus gros que celui obtenu à gauche. Mais il ne doit pas être nul, sinon, la condition d'inclusion de l'itération ne sera jamais vérifiée. Le temps de calcul dépend du choix de α et sera d'autant plus grand que α sera petit.

Comme cet algorithme échoue quand la condition entrée de boucle n'est pas satisfaisante, Nedialkov a proposé dans VNODE d'utiliser la méthode appelée : HOE (High-Order Enclosure) [Nedialkov 2001]. Cette méthode a moins de pessimisme que celle ci-dessus et elle utilise le développement en de série de Taylor d'ordre plus élevé que nous avons introduit précédemment. Elle génère automatiquement le pas d'intégration [Nedialkov 2001]. VSPODE [Lin 2007] a utilisé la même démarche que celui de VNODE.

3.2.3 Réduction du pessimisme sur la solution

L'ensemble $[\xi]$ contenant toutes les solutions possibles entre l'instant t_j et l'instant t_{j+1} peut être considéré comme une solution a priori à l'instant t_{j+1} de notre équation différentielle ordinaire pouvant être calculée avec l'algorithme *Enclosure*. La formule de Taylor étendue aux intervalles devient donc

$$[x_{j+1}] = [x_j] + \sum_{i=1}^{k-1} h^i f^{[i]}([x_j]) + h^k f^{[k]}([\tilde{x}_j]). \quad (3.39)$$

Par linéarité de l'opérateur $\omega(\cdot)$, nous obtenons :

$$\omega([x_{j+1}]) = \omega([x_j]) + \sum_{i=1}^{k-1} h^i \omega(f^{[i]}) + h^k \omega(f^{[k]}(\tilde{x}_j)) \geq \omega([x_j]). \quad (3.40)$$

Notons que le pas d'intégration h est positif et que la longueur d'un intervalle défini par l'opérateur $\omega(\cdot)$ l'est aussi. Nous remarquons que la taille de l'intervalle solution augmente à chaque pas d'intégration. Pour éviter ce phénomène, nous utilisons la forme centrée des coefficients de Taylor donnée par la formule :

$$f^{[i]}([x_j]) = f^{[i]}(\hat{x}_j) + J(f^{[i]}, [x_j])([x_j] - \hat{x}_j), \quad (3.41)$$

où $J(f^{[i]}, [x_j])$ est une fonction d'inclusion de la Jacobienne du i^{ieme} coefficient de Taylor et \hat{x}_j est le milieu du pavé $[x_j]$.

Nous remplaçons la nouvelle expression des coefficients dans la formule du développement de Taylor étendu aux intervalles (Équation 3.39), nous obtenons l'expression :

$$[x_{j+1}] = [x_j] + \sum_{i=1}^{k-1} h^i (f^{[i]}(\hat{x}_j) + J(f^{[i]}, [x_j])([x_j] - \hat{x}_j)) + h^k f^{[k]}([\tilde{x}_j]), \quad (3.42)$$

qui peut aussi s'écrire :

$$[x_{j+1}] = \hat{x}_j + \sum_{i=1}^{k-1} h^i f^{[i]}(\hat{x}_j) + h^k f^{[k]}([\tilde{x}_j]) + \{I + \sum_{i=1}^{k-1} h^i J(f^{[i]}, [x_j])\}([x_j] - \hat{x}_j). \quad (3.43)$$

Cette expression donne la possibilité de calculer l'itération à l'aide de la méthode de la valeur moyenne détaillée dans l'algorithme suivant :

Algorithme 11 Validated Integration by *Mean value*

Input: $[x_j]$, h , $[\tilde{x}_j]$;

Output: $[x_{j+1}]$;

- 1: $\hat{x}_j := m([x_j])$;
 - 2: $[v_{j+1}] := \hat{x}_j + \sum_{i=1}^{k-1} h^i f^{[i]}(\hat{x}_j) + h^k f^{[k]}([\tilde{x}_j])$;
 - 3: $[S_j] := I + \sum_{i=1}^{k-1} h^i J(f^{[i]}, [x_j])$;
 - 4: $[x_{j+1}] := [v_{j+1}] + [S_j]([x_j] - \hat{x}_j)$;
-

Notons qu'à la prochaine étape $\hat{x}_{j+1} = m(v_{j+1})$, parce que le vecteur $[S_j]([x_j] - \hat{x}_j)$ est symétrique. Le produit de la matrice intervalle $[S_j]$ par le vecteur intervalle $([x_j] - \hat{x}_j)$ au pas 3 de l'algorithme introduit un effet de grossissement du pavé solution $[x_{j+1}]$. Lorsque nous utilisons l'algorithme avec des vecteurs intervalles, cet effet provoque une augmentation de la taille du pavé solution à chaque instant d'intégration.

3.2.4 Effet d'enveloppement

Comme nous l'avons déjà expliqué au chapitre 1, ici, nous présentons cet effet sur un exemple détaillé.

Exemple 4. L'exemple de Moore [Moore 1966] illustre cet effet à chaque pas d'intégration. Considérons l'équation différentielle suivante :

$$\dot{u} = [1, 0] \times [-1, 0], u(0) \in [-0.1, 0.1] \times [0.9, 1.1]. \quad (3.44)$$

La solution de l'équation différentielle est un cercle de rayon unité dans le plan de phase (u_1, u_2) défini par

$$u(t) = \begin{bmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{bmatrix} u(0). \quad (3.45)$$

A chaque itération, l'algorithme de la valeur moyenne provoque une rotation du pavé solution qui est alors enveloppé dans un autre pavé dont les faces sont parallèles aux axes u_1 et u_2 . La taille du pavé solution augmente à chaque instant d'intégration alors que la vraie solution ne change pas. La Figure 3.2 illustre l'augmentation de la largeur du pavé solution. Dans cette Figure, le $h = 0.1$ et le $a = 0.001$. Le nombre d'itération est 20.

Dans la méthode d'intégration de *Mean value*, l'effet d'enveloppement apparaît au niveau du produit de la matrice $[S_j]$ avec le vecteur intervalle $([x_j] - \hat{x}_j)$. En effet, nous calculons

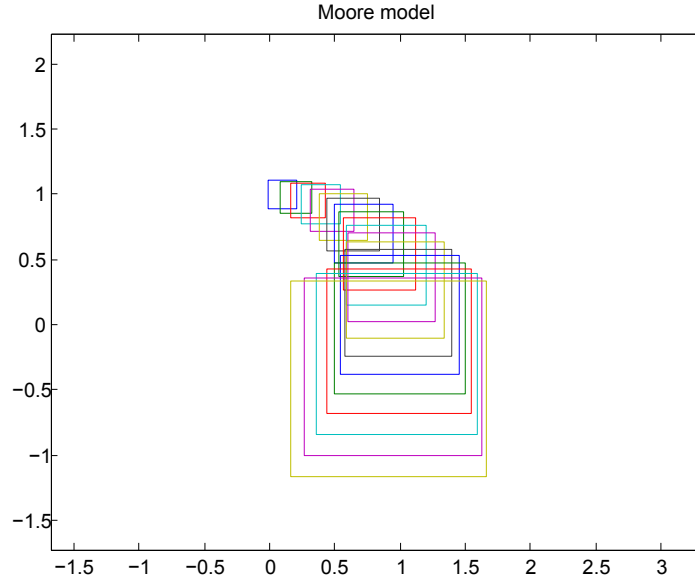


FIGURE 3.2 – L'effet d'enveloppement

$$\begin{aligned}
[x_1] &= [v_1] + [S_0]([x_0] - \hat{x}_0), \\
[x_2] &= [v_2] + [S_1]([x_1] - \hat{x}_1) \\
&= [v_2] + [S_1]([v_1] - \hat{x}_1 + [S_0]([x_0] - \hat{x}_0)) \\
&= [v_2] + [S_1]([v_1] - \hat{x}_1) + [S_1][S_0]([x_0] - \hat{x}_0), \\
&\vdots \\
[x_{j+1}] &= [v_{j+1}] + [S_j]([x_j] - \hat{x}_j) \\
&= [v_{j+1}] + [S_j]([v_j] - \hat{x}_j) + [S_j]([S_{j-1}](v_{j-1} - \hat{x}_{j-1})) + \dots \\
&\quad + [S_j]([S_{j-1}] \dots ([S_1]([S_0]([v_0] - \hat{x}_0))) \dots),
\end{aligned} \tag{3.46}$$

où $[v_0] = [x_0]$. Le vecteur symétrique $([v_j] - \hat{x}_j)$ provoque donc bien un enveloppement du pavé solution à chaque pas d'intégration.

Un nouvel algorithme sera présenté de manière à réduire cet effet d'enveloppement en effectuant une transformation linéaire sur les matrices.

3.2.4.1 Méthode de *Extended Mean Value*

Nous présentons la méthode *Extended Mean Value* développée par Rihm [Rihm 1994]. Cette méthode est basée sur un pré-traitement des matrices à l'aide de transformations linéaires pour réduire l'effet d'enveloppement. L'algorithme utilisé est le suivant

Algorithme 12 Validated Integration by *Extended Mean Value*(EMV)

Input: $[x_j], \hat{x}_j, h, [\tilde{x}_j], [v_j], [p_j], A_j$;

Output: $[x_{j+1}], \hat{x}_{j+1}, [v_{j+1}], [p_{j+1}], A_{j+1}$;

- 1: $[v_{j+1}] := \hat{x}_j + \sum_{i=1}^{k-1} h^i f^{[i]}(\hat{x}_j) + h^k f^{[k]}(\hat{x}_j)$;
 - 2: $[S_j] := I + \sum_{i=1}^{k-1} h^i J(f^{[i]}, [x_j])$;
 - 3: $[q_{j+1}] := ([S_j]A_j)[p_j] + [S_j]([v_j] - \hat{x}_j)$;
 - 4: $[x_{j+1}] := [v_{j+1}] + [q_{j+1}]$;
 - 5: $A_{j+1} := m([S_j]A_j)$;
 - 6: $[p_{j+1}] := A_{j+1}^{-1}([S_j]A_j)[p_j] + A_{j+1}^{-1}[S_j]([v_j] - \hat{x}_j)$
 - 7: $\hat{x}_{j+1} := m([x_{j+1}])$
-

Valeurs des différentes variables de l'algorithme à l'instant $j = 0$

- $\hat{x}_0 = m([x_0])$
- $[p_0] = 0$
- $A_0 = I$ où I est la matrice identité
- $[v_0] = [x_0]$

Il existe deux méthodes pour choisir la matrice A_{j+1} : la factorisation QR et la méthode parallélépipédique. Ces méthodes ont été proposées par Löhner [Löhner 1988]. Nous présentons la méthode parallélépipédique pour sa simplicité. Dans ce cas, la matrice A_{j+1} peut être calculée par la formule suivante :

$$A_{j+1} = m([S_j]A_j). \quad (3.47)$$

Pour contrôler l'effet d'enveloppement, la matrice A doit satisfaire la condition suivante :

$$A_{j+1}^{-1}([S_j]A_j) \approx I, \quad (3.48)$$

où I est la matrice identité.

Remarque 3. Soit $\hat{S}_j = m([S_j])$, le point milieu de $[S_j]$. Nous introduisons la matrice d'intervalle $[E_j]$ telle que $[E_j] = [S_j] - \hat{S}_j$, alors :

$$A_{j+1} = \hat{S}_j A_j, \quad (3.49)$$

A_j est une matrice non singulière dont les composantes sont des valeurs ponctuelles et non des intervalles donc $m(A_j) = A_j$.

$$\begin{aligned} A_{j+1}^{-1} \hat{S}_j A_j &= A_{j+1}^{-1} (\hat{S}_j A_j), \\ A_{j+1}^{-1} ([S_j] - [E_j]) A_j &= A_j^{-1} \hat{S}_j^{-1} (\hat{S}_j A_j), \\ A_{j+1}^{-1} ([S_j] A_j) &= A_j^{-1} \hat{S}_j^{-1} (\hat{S}_j A_j + [E_j] A_j) \\ &= I + A_j^{-1} \hat{S}_j^{-1} [E_j] A_j. \end{aligned} \quad (3.50)$$

Comme

$$\|A_j^{-1} \hat{S}_j^{-1} [E_j] A_j\| \leq \text{cond}(A_j) \|\hat{S}_j^{-1} [E_j]\|. \quad (3.51)$$

Si $\|\hat{S}_j^{-1}[E_j]\|$ est assez petit et que $\text{cond}(A_j)$ n'est pas trop grand, alors $A_{j+1}^{-1}([S_j]A_j) \approx I$

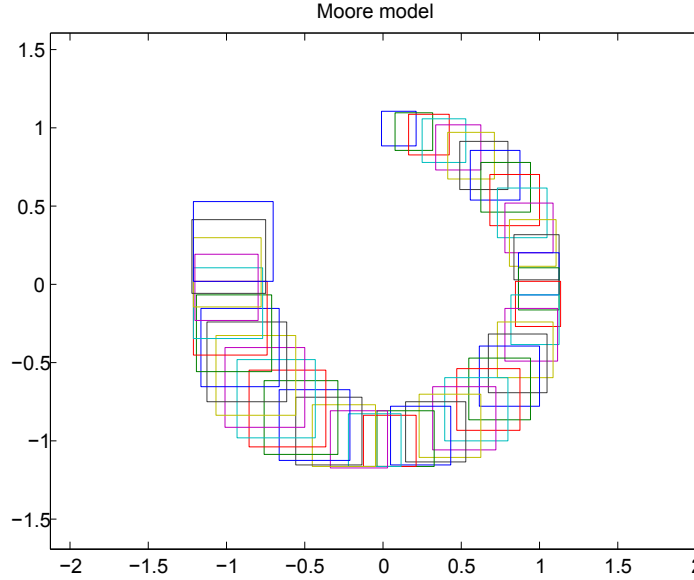


FIGURE 3.3 – L'effet d'enveloppement calculé par la Extended Mean Value

Par conséquent, nous pouvons augmenter le nombre d'itérations de la méthode de Taylor comme la Figure 3.3, le nombre d'itérations est égale 50, les autres conditions initiales sont identiques que pour l'exemple à la Figure 3.2.

Basé dans le même principe, des améliorations sur l'algorithme ont été prises au fil de l'eau. Par exemple, l'utilisation le développement en série d'Hermite-Obreschkoff à Darboux [Darboux 1876] et Hermite [Hermite 1874] quand nous déployons la forme Taylor d'une forme centrée [Nedialkov 2001] ou bien utiliser d'autre forme pour contrôler l'évolution de l'erreur [Lin 2007] [Houska 2013], etc. Afin de comparer ces différentes approches, nous développons l'algorithme 12 sous INTLAB sur un exemple tiré du domaine pharmaceutique.

3.2.5 Solveur disponible pour l'intégration garantie

Il existe de nombreux solveurs pour faire l'intégration garantie, mais certaines bibliothèques ne prennent pas en compte les erreurs d'arrondis durant le calcul. Aussi les solveurs ne sont pas tous strictement fiables. Par exemple, quand nous calculons l'inverse d'une matrice, tous les solveurs que nous avons testés ont utilisé LAPACK / BLAS, qui est une bibliothèque permettant de faire des transpositions et des inversions pour des matrices, même si les erreurs d'arrondis sont presque nulles dans des ordinateurs modernes elles sont de l'ordre 2^{-52} , cet effet sera cumulé si nous ne le prenons pas en compte sérieusement.

Les solveurs (ou algorithmes) que nous allons tester sont EMV, VNODE-LP [Nedialkov 2001], VSPODE [Lin 2007]. Ces solveurs sont faciles à trouver sur Internet,

ou bien à demander aux auteurs.

Les performances du solveur seront testées sur un modèle pharmaceutique et puis sur le modèle d'avion. Le temps de calcul et le diamètre des solutions obtenues seront comparées.

3.3 Intégration numérique garantie basée sur le théorème de Müller

Pour encadrer le modèle dynamique comme (3.3) avec les paramètres incertains $[p] \in [\underline{p}_0, \bar{p}_0]$ et les conditions initiales incertaines $[x_0] \in [\underline{x}_0, \bar{x}_0]$, nous pouvons trouver deux systèmes qui englobent les extrémités de toutes les solutions possibles avec ces conditions selon le théorème suivant [Müller 1927], [Kieffer 2011] :

Théorème 2. *Supposons que f soit continue sur :*

$$\mathbb{T} : \begin{cases} \omega(t) \leq x(t) \leq \Omega(t), \\ \underline{p}_0 \leq p \leq \bar{p}_0, \\ t_0 \leq t \leq t_1. \end{cases}$$

où $\omega_i(t)$ et $\Omega_i(t)$, $i = 1, \dots, n_x$ sont continues sur $[t_0, t_1]$, comme :

- $\omega(t_0) = \underline{x}(t_0)$ et $\Omega(t_0) = \bar{x}(t_0)$,
- La dérivée à gauche $D^-\omega_i(t)$ et $D^-\Omega_i(t)$ et la dérivée à droite $D^+\omega_i(t)$ et $D^+\Omega_i(t)$ de $\omega_i(t)$ et $\Omega_i(t)$ satisfont, pour $i = 1, \dots, n_x$ et $t \in [t_0, t_1]$

$$\begin{cases} D^\pm \omega_i(t) \leq \min_{\mathbb{T}_i(t)} f_i(x, p, t), \\ D^\pm \Omega_i(t) \geq \max_{\mathbb{T}_i(t)} f_i(x, p, t), \end{cases}$$

où $\mathbb{T}_i(t)$ est le sous ensemble de \mathbb{T} défini par :

$$\mathbb{T}_i(\tau) : \begin{cases} x_i = \omega_i(\tau), \\ \omega_j(\tau) \leq x_j \leq \Omega_j(\tau) \quad j \neq i, \\ \underline{p}_0 \leq p \leq \bar{p}_0, \\ \tau = t \end{cases}$$

et $\bar{\mathbb{T}}_i(t)$ est le sous ensemble de \mathbb{T} défini par :

$$\bar{\mathbb{T}}_i(\tau) : \begin{cases} x_i = \Omega_i(\tau), \\ \omega_j(\tau) \leq x_j \leq \Omega_j(\tau) \quad j \neq i, \\ \underline{p}_0 \leq p \leq \bar{p}_0, \\ \tau = t \end{cases}$$

Alors, pour n'importe quel $x(t_0) \in [\underline{x}_0, \bar{x}_0]$ et $p \in [\underline{p}_0, \bar{p}_0]$, une solution du système 3.3 existe comme :

$$\omega(t) \leq x(t) \leq \Omega(t) \quad \forall t \in [t_0, t_1]. \quad (3.52)$$

Suite à cette proposition, un solveur IOLAVABE [Eggers 2012] a été proposé en intégrant les deux types de l’algorithme : l’un le théorème de Müller et l’autre le théorème utilisant les séries de Taylor si cela est nécessaire. Quand la monotonicité existe, le solveur utilise la propriété sur le champ différentiel pour générer les tubes atteignables, sinon, l’intégration avec la série de Taylor va être appliquée pour calculer les solutions garanties. Dans le paragraphe suivant, nous présentons un exemple permettant de comparer les différents solveurs. Nous montrerons aussi comment s’appliquer le théorème de Müller sur cet exemple.

3.4 Application dans le domaine pharmaceutique

Nous cherchons à estimer les états d’un modèle de comportement de deux compartiments à partir de données simulées [Kieffer 1999]. Les données aberrantes seront donc écartées. Notre simulation est effectuée sur 20 secondes et les sorties sont enregistrées toutes les 0.1s.

Considérons le modèle simple décrit par la Figure 3.4. La dynamique est linéaire, mais ses sorties dépendent de manière non linéaire en les paramètres. Son comportement peut être décrit par le système suivant :

$$x' = f(x, p, u), \quad (3.53)$$

où $p = (p_1, p_2, p_3) = (k_{21}, k_{12}, k_{01})^T$ avec

$$\begin{cases} \dot{x}_1 &= -(p_1 + p_3)x_1 + p_2x_2 + u, \\ \dot{x}_2 &= p_1x_1 - p_2x_2. \end{cases} \quad (3.54)$$

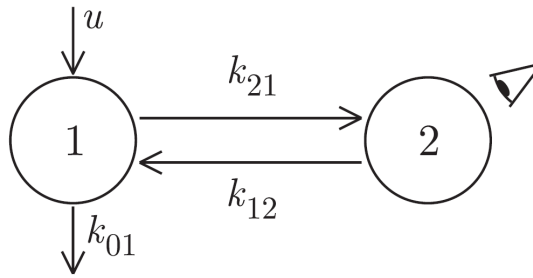


FIGURE 3.4 – Modèle de deux compartiments

Pour estimer les états, de nombreux solveurs sont disponibles dans le domaine de l’analyse par intervalles, les plus souvent utilisés sont VNODE-LP (Interval Taylor Series), IOLAVABE (Monotone System) et VSPODE (Taylor Residus). Les résultats donnés par la méthode EMV (Extended Mean Value) sous INTLAB sont fournis et comparés. Il faut constater que la méthode du système monotone nécessite une préconnaissance sur la monotonicité du système et VNODE-LP n’admet pas une entrée d’intervalle très

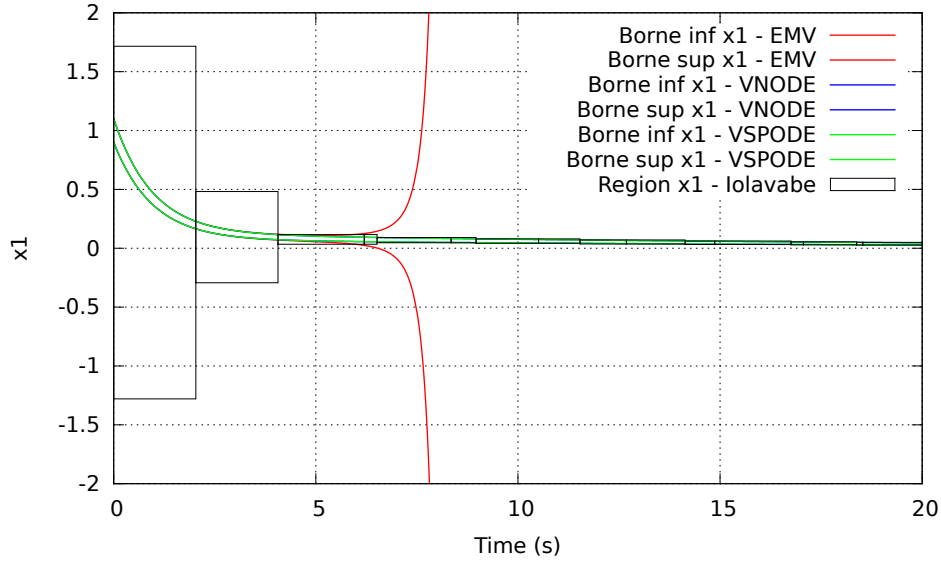


FIGURE 3.5 – Evolution de l'intervalle contenant x_1 fourni par VNODE-LP, VSPODE, IOLAVABE, EMV

large elle conduit rapidement aux solutions inutiles du calcul, voire l'infini sur les bornes d'intervalles solutions.

Comme nous l'avons présenté dans le chapitre 2, VNODE-LP, VSPODE, IOLAVABE seront nos trois candidats à la comparaison. IOLAVABE est une bibliothèque gratuite sur internet, il écrit automatiquement le système sous forme monotone $(\underline{\mathbb{T}}, \overline{\mathbb{T}})$ selon le théorème de Müller, ce qui évite à l'utilisateur de l'écrire à la main.

Dans la suite, nous allons comparer les performances de chaque méthode sur l'exemple de compartiment qui sera illustré par la suite.

Pour une raison de simplicité, prenons un système sans l'entrée u avec les paramètres $p_1 = 0.6$, $p_2 = 0.15$, $p_3 = 0.35$. Notons que seuls les états sont sous formes d'intervalles.

Alors, notre système devient :

$$\begin{cases} \dot{x}_1 &= -(p_1 + p_3)x_1 + p_2x_2, \\ \dot{x}_2 &= p_1x_1 - p_2x_2. \end{cases} \quad (3.55)$$

IOLAVABE nous permet de créer le nouveau système d'état sous la forme suivante :

$$\begin{aligned} \underline{x}'_1 &= -(p_1 + p_3)\underline{x}_1 + p_2\underline{x}_2, \\ \underline{x}'_2 &= p_1\underline{x}_1 + p_2\underline{x}_2, \\ \overline{x}'_1 &= -(p_1 + p_3)\overline{x}_1 + p_2\overline{x}_2, \\ \overline{x}'_2 &= p_1\overline{x}_1 - p_2\overline{x}_2. \end{aligned} \quad (3.56)$$

Pour comparer ces différentes méthodes, les états initiaux ont été pris comme $x_1 = [0.9, 1.1]$ et $x_2 = [-0.1, 0.1]$.

Le calcul est fait par les trois bibliothèques VNODE-LP, VSPODE, IOLAVABE et par l'algorithme EMV : extended mean value (qui est développé sous INTLAB).

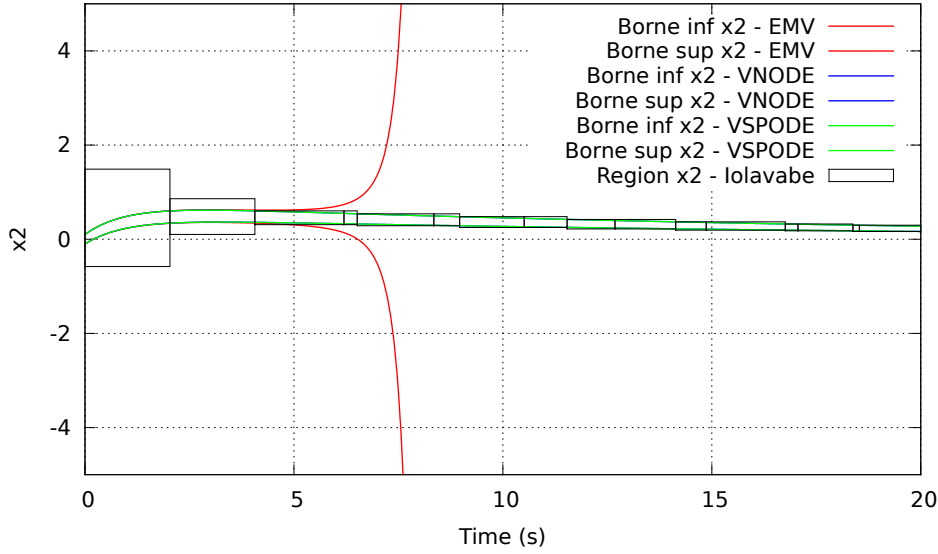


FIGURE 3.6 – Evolution de l'intervalle contenant x_2 fourni par VNODE-LP, VSPODE, IOLAVABE, EMV

C'est clair que la vraie solution a bien été encadrée par les méthodes proposées, sauf que la méthode EMV a explosé à $t = 7.6s$ pour le modèle de compartiment. Les méthodes réalisées par VNODE-LP, VSPODE et IOLAVABE sont arrivées à la fin de simulation. Les résultats de VSPODE sont quasiment identiques à ceux qui ont été obtenus par VNODE-LP.

IOLAVABE ne peut pas fournir des résultats avec un pas fixé et ceci pose un problème dans l'inversion ensembliste, car nous ne pouvons pas obtenir un résultat à l'instant donné. Le temps de calcul de VSPODE est très long par rapport à celui de VNODE-LP. Finalement, dans la suite nous utilisons l'outil VNODE-LP pour générer de façon garantie les solutions de l'EDO dans la suite de l'analyse.

L'état dans le compartiment 2 est considérée comme observable, alors

$$y^m(t_i, p) = x_2(t_i, p), i = 1, \dots, n_t. \quad (3.57)$$

Nous considérons que l'entrée u est nulle, les états initiaux sont $x_0 = (1, 0)^T$.

Les paramètres que nous allons estimer sont alors p_1, p_2 et p_3 . Le système a été transformé en un problème d'estimation paramétrique.

Nous générons les mesures par un solveur de l'EDO en fixant les paramètres comme $(p_1, p_2, p_3) = (0.6, 0.15, 0.35)$. Nous supposons que seul x_2 est observable.

Nous supposons que les observations sont perturbées par des bruits de mesures inconnus $e(t_j, p)$ vérifiant l'équation suivante :

$$e(t_j, p) = y(t_j) - y^m(t_j, p), \text{ pour } j = 1, \dots, N, \quad (3.58)$$

où $y(t_j)$ représente le vecteur des mesures et $y^m(t_j, p)$ celui des observations et où $e(t_i, p)$ correspond aux bruits de mesures inconnus mais dont les bornes inférieures et supérieures

sont connues (selon les propriétés d'un capteur par exemple). Ces bornes pourront varier selon les capteurs, dans notre cas, elles sont supposées constantes égales à $[-0.005, 0.005]$.

3.4.1 Estimation d'état

Nous pouvons utiliser VNODE-LP ou IOLAVABE afin de trouver le domaine garanti. Mais actuellement, IOLAVABE ne fournit pas de moyen qui nous permette de recommencer l'intégration après l'intersection avec les mesures données à l'instant défini. De plus, VNODE-LP est plus facile à modifier et à comprendre. Sans précision de notre part, tous les résultats qui nécessitent l'utilisation de l'intégration garantie seront obtenus avec VNODE-LP. Pour illustrer, nous supposons que les états initiaux sont connus et qu'ils appartiennent à l'ensemble suivant :

$$[X_0] = [0.9, 1.1] \times [-0.1, 0.1]. \quad (3.59)$$

La Figure 3.7 représente l'estimation d'état. Les paramètres sont supposés connus, la seule incertitude se trouve sur les états initiaux. La ligne du milieu de chaque tuyau est la solution exacte donnée par le VNODE-LP, calculée à partir d'un état ponctuel : $x = (1, 0)$.

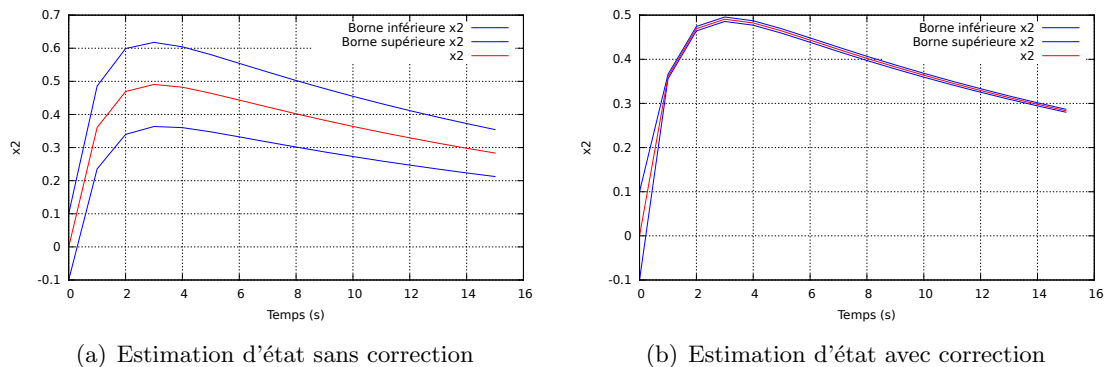


FIGURE 3.7 – Estimation d'état pour le modèle pharmaceutique

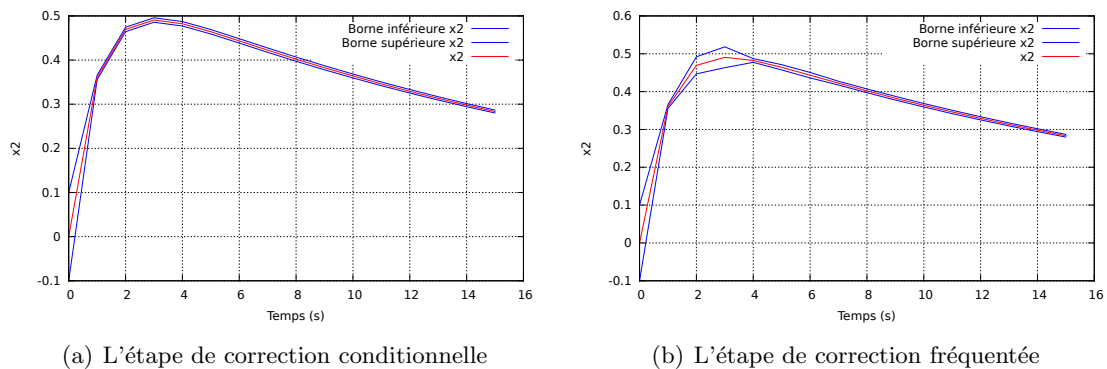


FIGURE 3.8 – Estimation d'état pour le modèle pharmaceutique

L'instant d'intersection avec les mesures sur la Figure 3.8(a) est $t(i)$ avec $i \in (1, 2, 3, 4, 5, 6)$, celui de la Figure 3.8(b) est $t(i)$ avec $i \in (1, 4, 7, 10, 13)$. Pour ces deux figures, il y a 15 instants de mesures. Nous voyons que, avec l'algorithme 5, nous avons obtenu des résultats plus précis et diminué le nombre d'appel à l'étape de correction. La solution exacte est incluse dans toutes ces figures, ce qui montre que la solution garantie est correcte.

3.4.2 Estimation paramétrique

Les mesures sont artificiellement générées à partir du solveur VNODE-LP avec les états initiaux $x = (x_1, x_2) = (1, 0)$ et les paramètres exacts $p = (0.6, 0.15, 0.35)$ pendant la durée de simulation de $t = 15s$. Les mesures sont disponibles à chaque seconde. Supposons que les erreurs de mesures soient bornés et appartiennent à un intervalle constant $[-0.005, 0.005]$. Alors l'idée est de caractériser les paramètres selon les sorties du modèle dont leurs états estimés sont enveloppés par les mesures : estimation à erreur bornée.

Un ensemble de paramètre p admissible est tel que

$$e_{inf} \leq y(t) - y^m(t, p) \leq e_{sup}, \quad (3.60)$$

où dans notre cas, la borne inférieure des erreurs est $e_{inf} = -0.005$ et la borne supérieure des erreurs est $e_{sup} = 0.005$. C'est-à-dire que nous pouvons vérifier si les paramètres sont cohérents avec les mesures en utilisant l'équation (3.60). Nous pouvons utiliser l'inversion ensembliste pour trouver les ensembles de paramètres $[p]$ tels que les sorties du modèle appartiennent à l'ensemble des mesures $[y]$.

$$\mathbb{S} = \{[p] \subset [P_0] \mid [p] \text{ admissible}\} = \{[p] \in [P_0] \mid y^m([p]) \subset [y]\}. \quad (3.61)$$

Avec cet algorithme, nous prenons comme erreurs sur les mesures $[y] = (x_2 + Err)$, où $Err = \pm 0.005$, l'état x_2 est calculé avec les valeurs exactes : $x = (1, 0)$ et $p = (0.6, 0.15, 0.35)$. Les paramètres sont supposés vérifier $[p] \in [0.01, 1]^3$.

Les figures 3.9, 3.10, 3.11 et 3.12 sont seulement les pavés de paramètres admissibles et incertains. La partie jaune représente la partie incertaine, la partie bleu représente la partie admissible.

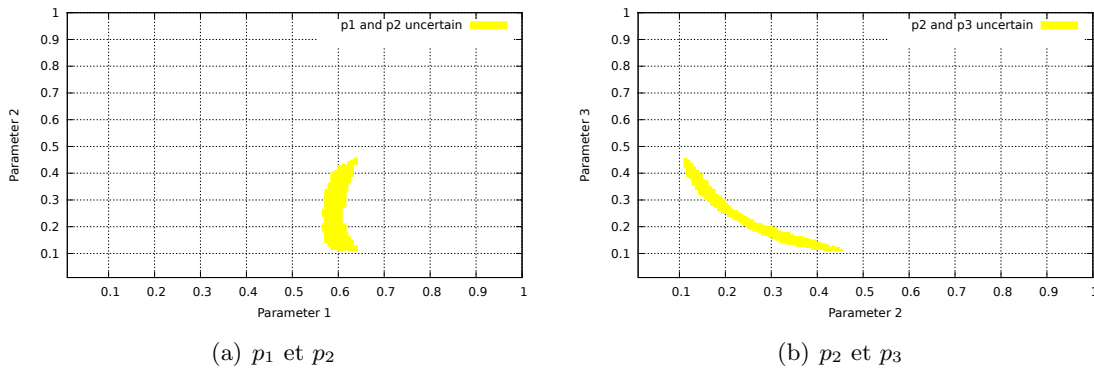
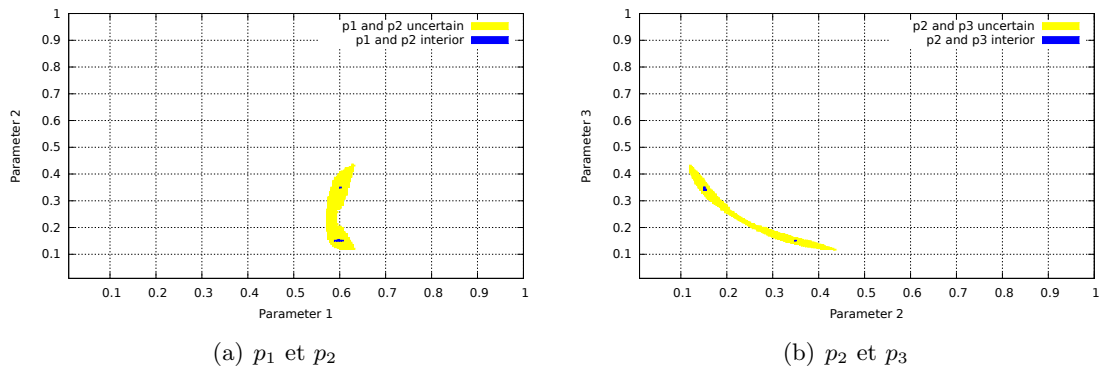
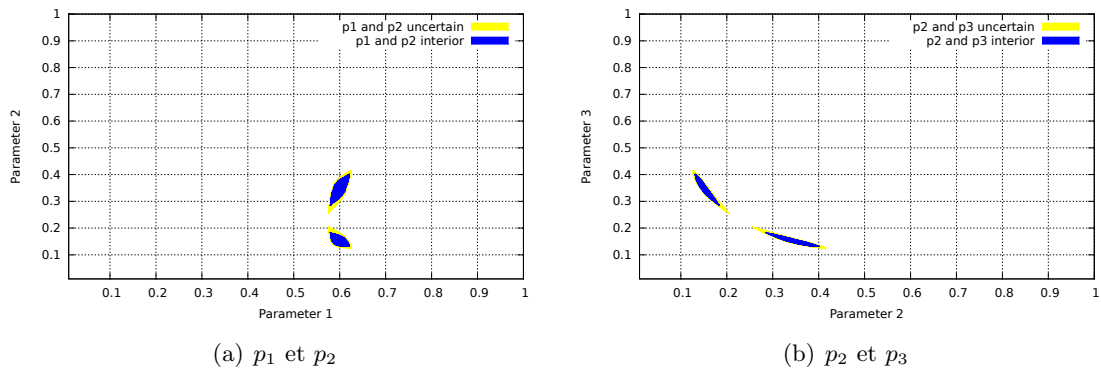
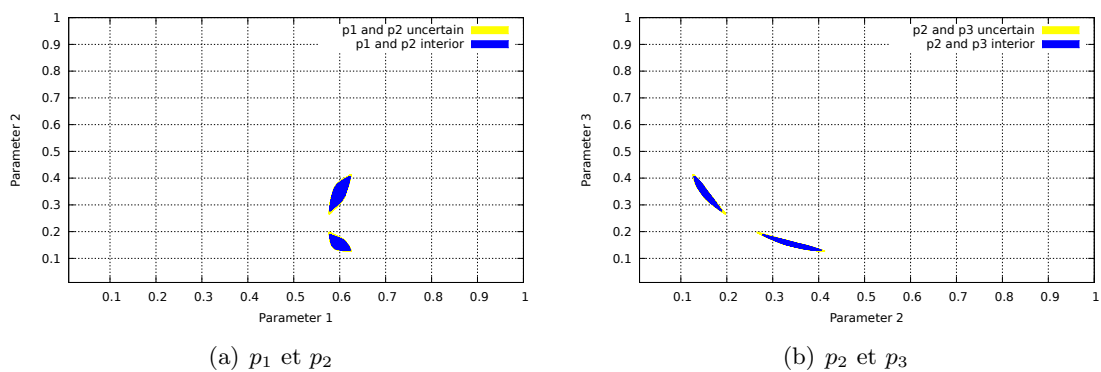


FIGURE 3.9 – Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite), $\varepsilon = 0.01$.

FIGURE 3.10 – Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite), $\varepsilon = 0.005$.FIGURE 3.11 – Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite), $\varepsilon = 0.001$.FIGURE 3.12 – Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite), $\varepsilon = 0.0005$.

Ou bien nous pouvons trouver ces ensembles de paramètres via l'approche d'optimisation en utilisant l'algorithme 6. En choisissant que l'écart entre la borne supérieure et la borne inférieure de paramètres choisi ne dépasse pas $\varepsilon = 0.00001$, avec les mêmes configurations, nous obtenons les figures 3.13 en 4372 secondes :

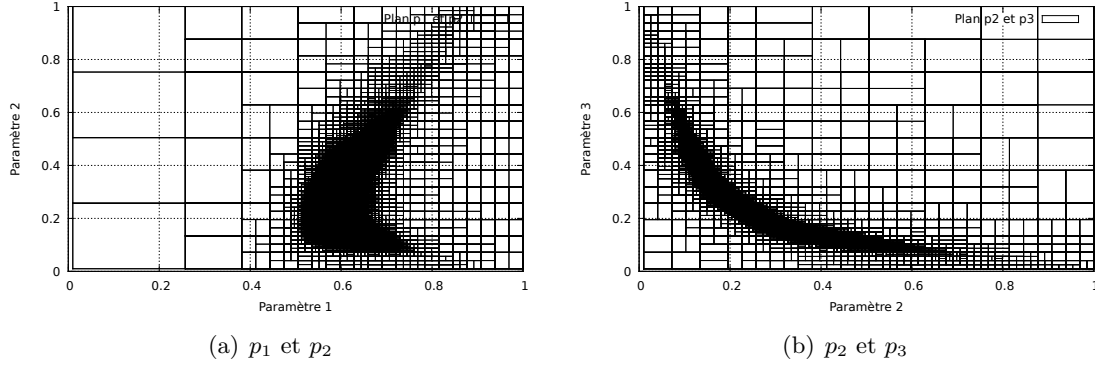


FIGURE 3.13 – Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite).

Nous voyons que ces résultats sont pire que ceux qui ont été obtenu par l'Algorithme 7 pour l'estimation de paramètres. L'approche d'optimisation a consommé plus de temps que l'inversion ensembliste avec les ensembles de solution plus important par rapport aux résultats obtenus par l'inversion ensembliste. Même si nous avons appliqué les algorithmes basiques de ces deux approches, l'inversion ensembliste semble plus intéressante pour nous.

Le problème avec des contraintes (les mesures ici) est un problème NP complet [Lin 2007], une méthode d'accélération afin de trouver plus rapidement l'ensemble de solution est souvent utilisée : le contracteur. Ici, nous pouvons utiliser un contracteur différentiel en utilisant la forme centrée pour enlever le domaine à rejeter et puis, nous prenons un processus d'inversion d'ensembliste afin de trouver les paramètres que nous nous intéressons.

Nous construisons le système de sensibilité avec le système d'état. C'est-à-dire un système de 11 équations :

$$\left\{ \begin{array}{l} \dot{x}_1 = -(p_1 + p_3)x_1 + p_2x_2, \\ \dot{x}_2 = p_1x_1 - p_2x_2, \\ \dot{p}_1 = 0, \\ \dot{p}_2 = 0, \\ \dot{p}_3 = 0, \\ \dot{s}_{11} = -(p_1 + p_3)s_{11} + p_2s_{21} - x_1, \\ \dot{s}_{21} = p_1s_{11} - p_2s_{21} + x_1, \\ \dot{s}_{12} = -(p_1 + p_3)s_{12} + p_2s_{22} + x_2, \\ \dot{s}_{22} = p_1s_{12} + p_2s_{22} - x_2, \\ \dot{s}_{13} = -(p_1 + p_3)s_{13} + p_2s_{23} - x_1, \\ \dot{s}_{23} = -p_1s_{13} - p_2s_{23}. \end{array} \right. \quad (3.62)$$

Notons que trouver la solution pour ce système d'équation pourrait être très compliqué, nous intégrons chaque fois les sensibilités liés avec un seul paramètre au système d'état afin de réduire le temps de calcul et le pessimisme, les systèmes sont alors :

$$\left\{ \begin{array}{l} \dot{x}_1 = -(p_1 + p_3)x_1 + p_2x_2, \\ \dot{x}_2 = p_1x_1 - p_2x_2, \\ \dot{p}_1 = 0, \\ \dot{p}_2 = 0, \\ \dot{p}_3 = 0, \\ \dot{s}_{11} = -(p_1 + p_3)s_{11} + p_2s_{21} - x_1, \\ \dot{s}_{21} = p_1s_{11} - p_2s_{21} + x_1. \end{array} \right. \quad (3.63)$$

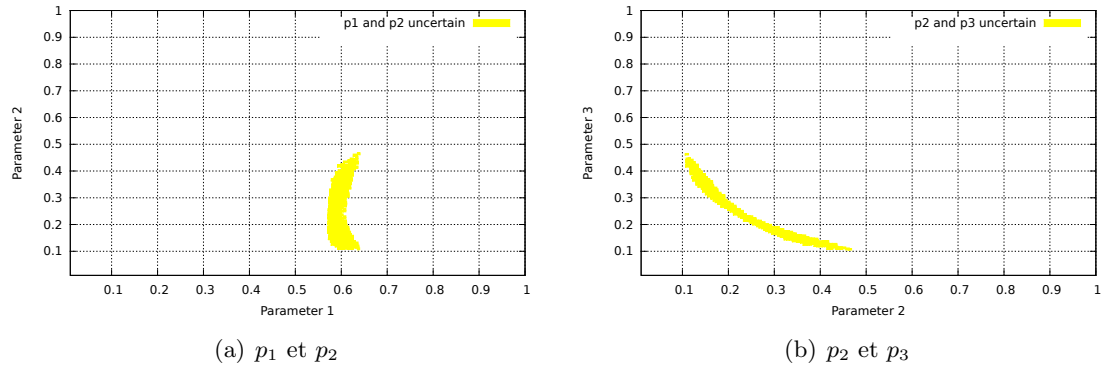
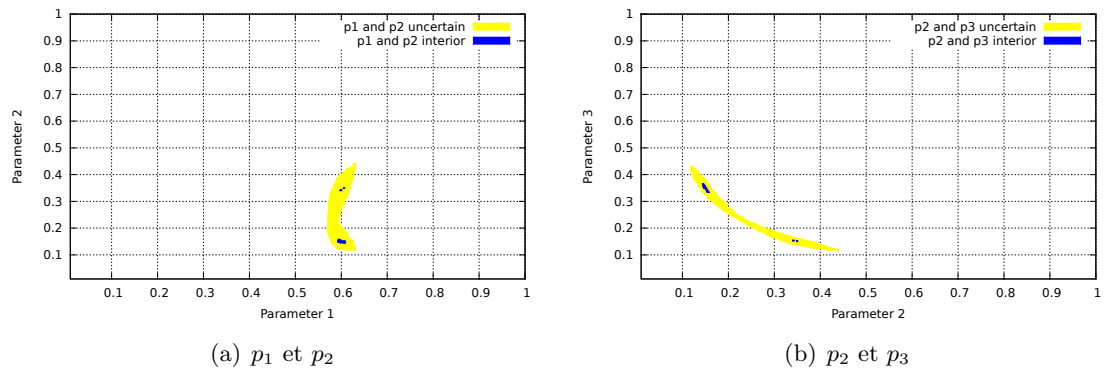
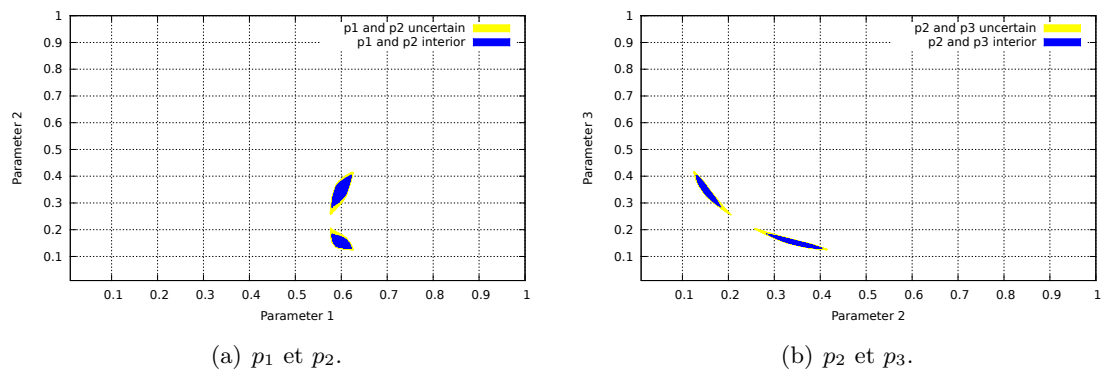
$$\left\{ \begin{array}{l} \dot{x}_1 = -(p_1 + p_3)x_1 + p_2x_2, \\ \dot{x}_2 = p_1x_1 - p_2x_2, \\ \dot{p}_1 = 0, \\ \dot{p}_2 = 0, \\ \dot{p}_3 = 0, \\ \dot{s}_{12} = -(p_1 + p_3)s_{12} + p_2s_{22} + x_2, \\ \dot{s}_{22} = p_1s_{12} + p_2s_{22} - x_2. \end{array} \right. \quad (3.64)$$

$$\left\{ \begin{array}{l} \dot{x}_1 = -(p_1 + p_3)x_1 + p_2x_2, \\ \dot{x}_2 = p_1x_1 - p_2x_2, \\ \dot{p}_1 = 0, \\ \dot{p}_2 = 0, \\ \dot{p}_3 = 0, \\ \dot{s}_{13} = -(p_1 + p_3)s_{13} + p_2s_{23} - x_1, \\ \dot{s}_{23} = -p_1s_{13} - p_2s_{23}. \end{array} \right. \quad (3.65)$$

Nous générons les mesures à chaque seconde jusqu'à $t = 15s$, les domaines a priori connus pour les paramètres sont :

$$[P_0] = [0.01, 1] \times [0.01, 1] \times [0.01, 1]. \quad (3.66)$$

Comme les paramètres sont constants, les conditions initiales des sensibilités sont nulles. Nous prenons d'abord $\varepsilon = 0.01$, et puis $\varepsilon = 0.005$, $\varepsilon = 0.001$, $\varepsilon = 0.0005$. Nous rappelons que cette valeur détermine la diamètre de bisection sur le domaine intéressé. Les figures 3.14, 3.15 et 3.16 sont des pavés de paramètres admissibles et incertains :

FIGURE 3.14 – Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite) $\varepsilon = 0.01$.FIGURE 3.15 – Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite) $\varepsilon = 0.005$.FIGURE 3.16 – Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite) $\varepsilon = 0.001$.

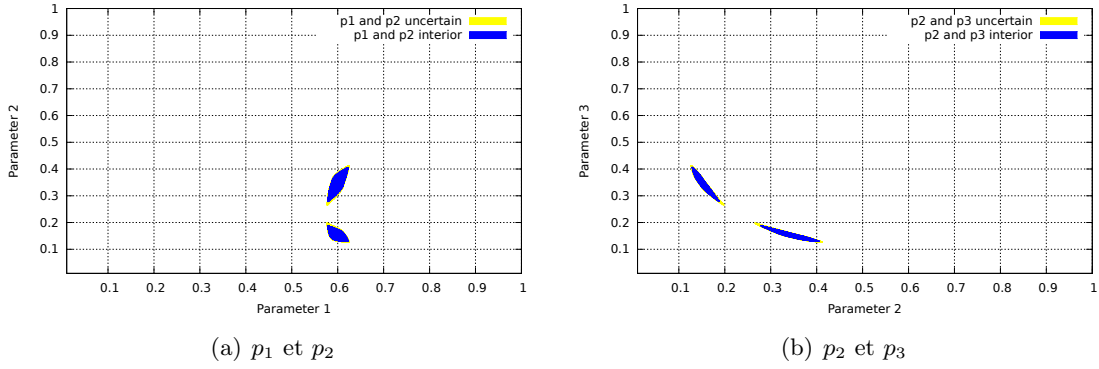


FIGURE 3.17 – Estimation paramétrique de p_1 et p_2 (gauche) et p_2 et p_3 (droite) $\varepsilon = 0.0005$.

Cette fois-ci, nous pouvons déduire des Figures 3.16(b) et 3.17(b) que les paramètres p_2 et p_3 ne sont que localement identifiable lorsque seul l'état x_2 est observé, parce que l'ensemble de solution est symétrique.

Nous pouvons le confirmer en résolvant le système d'état en échangeant les deux valeurs de paramètres : voir la Figure 3.18. Seul l'état x_1 est différent mais l'état x_2 reste le même. Cela est un problème d'identifiabilité dans le cas classique.

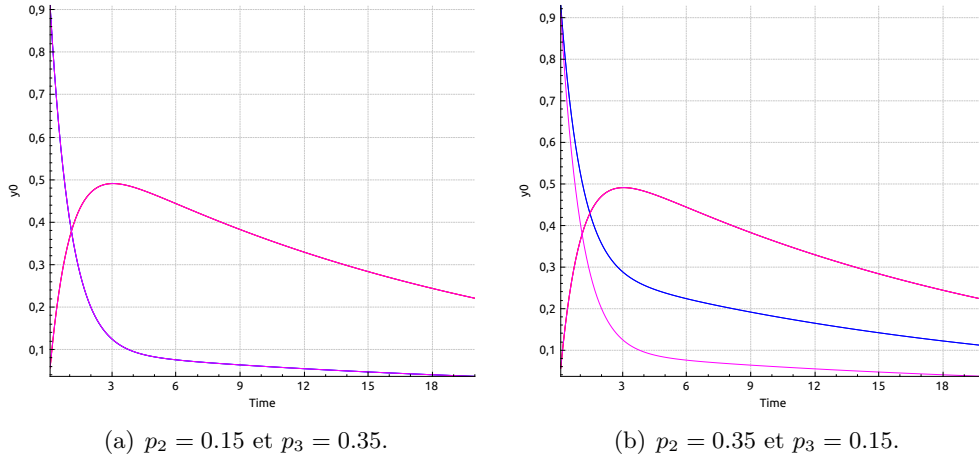


FIGURE 3.18 – Estimation d'état à partir de $p_2 = 0.35$ et $p_3 = 0.15$.

Ici, nous résumons le temps de calcul avec ou sans le contracteur. Le contracteur ¹ indique un système qui regroupe les équations d'état avec un système de sensibilité, le contracteur ² indique la technique proposée précédemment : construire plusieurs sous-systèmes 3.63, 3.64, 3.65.

Nous pouvons voir que la technique qui regroupe les équations de sensibilités avec le système d'état ne nous permet pas de réduire le temps de calcul. Cela pourrait être dû au fait que le temps passé sur VNODE-LP consacré à la calcul de l'ordre du série du Taylor pendant l'intégration garantie, chaque sous système doit être recalculé afin de générer les bonnes solutions a priori pour $[x]$, ce qui retarde le processus intégral.

D'autre part, quand la sensibilité contient zéro, une bisection doit être faite pour

TABLE 3.1 – Temps d'exécution du programme (s).

Temps d'exécution	$\varepsilon = 0.01$	$\varepsilon = 0.005$	$\varepsilon = 0.001$	$\varepsilon = 0.0005$
Sans contraction	188	341	3624	18067
Avec contraction 1	703	1880	25834	23332
Avec contraction 2	824	1900	33648	22181

générer les valeurs nécessaires. Ceci demande de recalculer les trois systèmes d'équation au lieu d'en traiter un.

3.5 Application dans le domaine pharmacocinétique

L'utilisation propre d'un contracteur accélère le processus d'estimation paramétrique. Nous illustrerons ceci par un exemple basé sur un système pharmacocinétique [Verdière 2005]. L'exemple illustratif mettant en œuvre l'algorithme 9 est le modèle suivant :

$$\begin{cases} \dot{x}_1 = k_{12}(x_2 - x_1) - k_v \frac{x_1}{1 + x_1}, & x_1(0) = x_{10}, \\ \dot{x}_2 = k_{21}(x_1 - x_2), & x_2(0) = 0, \\ y^m = x_1. \end{cases} \quad (3.67)$$

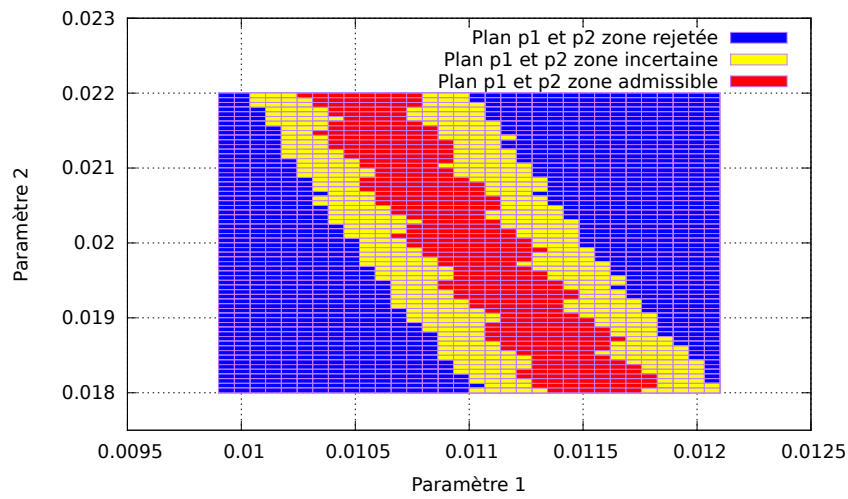
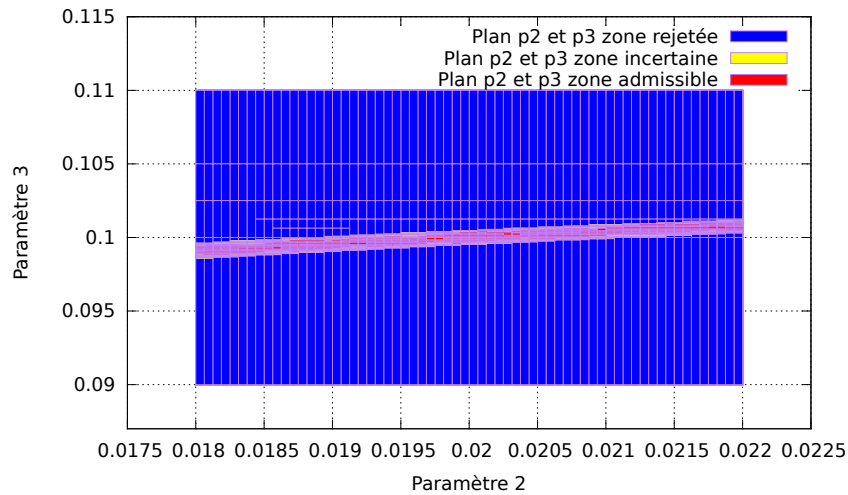
Les paramètres sont respectivement : $\theta = (k_{12}, k_{21}, k_v)$. Nous supposons la valeur des paramètres $\theta = (0.011, 0.02, 0.1)$ avec la condition initiale $x_{10} = 0.625$.

Pour tester ce système, l'essai est fait avec :

$$[\theta] = [0.0099, 0.0121] \times [0.018, 0.022] \times [0.09, 0.11], \quad (3.68)$$

en supposant que $x(0) = (x_1, x_2) = (0.625, 0)$. Le temps de simulation est 50s. les bruits sont supposés bornés dans un intervalle $e \in [-0.005, 0.005]$ sur x_1 . Les mesures sont disponibles à chaque seconde. Nous mettons d'abord $\varepsilon = 0.0001$ avec l'Algorithme 7 :

Les résultats de l'estimation de paramètres sur ce modèle sont des zones acceptées, rejetées et incertaines pour p_1, p_2, p_3 projetées dans le plan p_1 et p_2 , aussi dans le plan p_2 et p_3 présentés ci-dessous :

FIGURE 3.19 – Estimation de paramètre sur p_1 et p_2 .FIGURE 3.20 – Estimation de paramètre sur p_2 et p_3 .

Les parties bleues sont des boîtes rejetées, les boîtes jaunes sont des boîtes indéterminées, les boîtes rouges sont des boîtes acceptables selon l'Algorithme 7.

Selon la méthode proposée dans l'article [Kieffer 2011], nous pouvons utiliser ce système pour construire un contracteur afin de réduire le temps de calcul. Il propose d'utiliser la théorème de Müller en utilisant un intégrateur garanti pour trouver des solutions valides. Dans notre approche, nous avons appliqué un intégrateur garanti VNODE-LP directement sur notre système étudié pour tester sa méthode. Le système de sensibilité de ce système est écrit sous la forme suivante :

$$\left\{ \begin{array}{l} \dot{s}_{11} = \left(-\frac{k_v}{(x_1 + 1)^2} - k_{12}\right)s_{11} + k_{12}s_{21} + (x_2 - x_1), \\ \dot{s}_{21} = k_{21}s_{11} - k_{21}s_{21}, \\ \dot{s}_{12} = \left(-\frac{k_v}{(x_1 + 1)^2} - k_{12}\right)s_{12} + k_{12}s_{22}, \\ \dot{s}_{22} = k_{21}s_{12} - k_{21}s_{22} + (x_1 - x_2), \\ \dot{s}_{13} = \left(-\frac{k_v}{(x_1 + 1)^2} - k_{12}\right)s_{13} + k_{12}s_{23} - \frac{x_1}{1 + x_1}, \\ \dot{s}_{23} = k_{21}s_{13} - k_{21}s_{23}. \end{array} \right. \quad (3.69)$$

Le Tableau 3.2 a illustré les résultats de simulation, les volumes sont de l'ordre de e^{-10} qui est 1760. Le contracteur conditionnel est un algorithme de test, qui est activé si nous traitons des boîtes rejetées, si nous traitons d'autres boîtes, nous le désactivons.

TABLE 3.2 – Volume de $p_1p_2p_3$.

Volume (e^{-10})	Sans contracteur	Avec contracteur sans interruption	Avec contracteur conditionnel
Boîte acceptable	1.6952	1.9439	1.8246
Boîte incertaine	10.578	6.7395	8.0418
Boîte rejeté	1747.7	1.7821	65.741
Temps de calcul (<i>sec</i>)	532	843	460

Nous avons vu que le volume de boîte rejetée augmente quand nous appliquons le contracteur sous certaine condition. La bisection [Jaulin 1993] [Jaulin 2001a] est un traitement long par rapport au temps. Une décision avant la mise en marche du contracteur est un facteur important de réduction du temps de calcul. Le contracteur est d'abord un moyen de réduire les boîtes rejeteables ce qui entraîne que le nombre de fois où nous utilisons la bisection diminue. Un indice qui évite de rentrer dans la logique pour des boîtes incertaines et des boîtes rejeteables serait le bienvenu. L'indice volumique est un choix raisonnable. Il a des propriétés comme :

1. Quand les dimensions du vecteur d'intervalle sont larges, le contracteur peut réduire efficacement des boîtes inappropriée.
2. Quand les dimensions d'intervalle sont petits, le contracteur peut être mis en veille afin de ne pas ralentir le calcul de SIVIA.

Souvent, quand le diamètre de $[p]$ est large, il reste à déterminer s'il y a encore des boîtes rejeteables. Comme nous l'avons déjà souligné, quand le temps de calcul est long, les sorties de l'intégration garantie explosent, cela entraîne que ces solutions deviennent inutile. Alors un intervalle de paramètres avec des largeurs plus petites doit être choisi pour que ce calcul soit fait, ce qui demande alors une bisection. Selon le critère de bisection, les boîtes qui restent à la fin seront des boîtes petites, avec un diamètre de l'ordre de ε , $d[p] = C\varepsilon$, où $C \in [0.5, 2]$ à cause de l'effet bisection précédente.

Pour trois paramètres à estimer, cette constante peut être fixée comme $2^3 = 8$, mais, dans la pratique, il faut légèrement l'agrandir. Ici, dans la suite de l'exemple, nous prenons 10 comme constante de décision.

Le contracteur est utilisé uniquement lorsque la condition :

$$\prod_{i=1}^{n_p} rad([p_i]) > C \times \prod_{i=1}^{n_p} (\varepsilon), \quad (3.70)$$

est satisfaite avec $C = 10$. Sinon, seule l'estimation de paramètres utilisant l'inversion ensembliste va être appliquée [Li 2015d].

L'essai est fait avec $\varepsilon = 0.0001$:

TABLE 3.3 – Volume de $p_1 p_2 p_3$.

Volume (e^{-10})	Sans contracteur	Avec contracteur sans interruption	Avec contracteur volumique
Boite acceptable	1.6952	1.9439	1.3811
Boite incertaine	10.578	6.7395	9.8051
Boite rejetée	1747.7	1.7821	4.665
Temps de calcul (<i>sec</i>)	532	843	155

et la valeur de $\varepsilon = 0.00005$, nous obtenons le même résultat :

TABLE 3.4 – Volume de $p_1 p_2 p_3$.

Volume (e^{-10})	Sans contracteur	Avec contracteur sans interruption	Avec contracteur volumique
Boite acceptable	3.8286	4.0176	3.6808
Boite incertaine	5.2993	2.4375	4.7196
Boite rejetée	1750.9	1.8757	3.4242
Temps de calcul (<i>sec</i>)	2270	4143	731

Le temps de calcul a été divisé par trois par rapport au temps normal de l'estimation paramétrique et encore plus par rapport au contracteur sans arrêt. Grâce au fait que le nombre de calculs diminue, la précision numérique peut être améliorée.

3.6 Application dans le domaine aéronautique

Nous cherchons à estimer les états pour un modèle de comportement en vol longitudinal d'un avion à partir de données à erreurs bornées [Jaubert 2002] et puis à partir de ces états, la zone de paramètres admissibles sera trouvée via l'estimation de paramètres. L'approche de l'analyse par intervalles nous permet d'estimer les paramètres de façon sûre, trouver et valider à la fois le domaine de paramètres souhaité. Le cas de données aberrantes sera écarté. Notre simulation est effectuée pendant 1 seconde et les sorties sont mesurées toutes les 0.1 s. Nous prendrons donc le pas d'intégration $h = 0.1$ s pour pouvoir évaluer les sorties de notre modèle aux instants de mesures choisis.

3.6.1 Modèle du vol longitudinale d'un avion en atmosphère au repos

Le modèle étudié est celui du Dornier 28 établi par des essais en soufflerie [Coton 2001]. Il s'agit d'un modèle de comportement en vol longitudinal d'un avion dans une atmosphère en repos. Un vecteur d'état est : $x = (V, \alpha, q, \theta)^T$, où

- V : la vitesse de l'avion en mètres par seconde,
- α : l'incidence en degrés qui correspond à la direction du nez de l'avion par rapport au vecteur vitesse dans le plan vertical avec unité en degrés,
- q : la vitesse de tangage en degrés par seconde,
- θ : l'assiette longitudinale en degré.

La plupart des variables décrites précédemment peuvent être repérées sur le schéma d'avion de la Figure 3.21 :

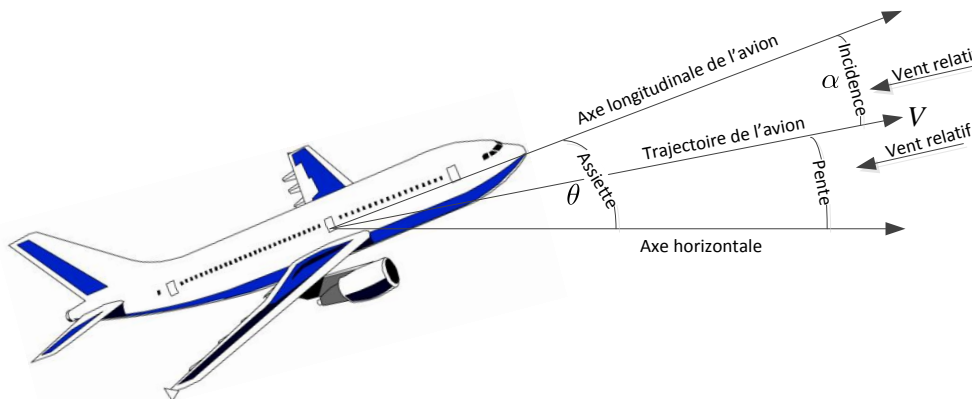


FIGURE 3.21 – Incidence et vecteur vitesse d'un avion.

Nous supposons que le pilote agit sur la commande de direction pour maintenir le virage nul. En outre, nous supposons que le pilote agit sur la commande de gauchissement pour maintenir la résultante aérodynamique dans un plan vertical, ceci signifie que la trajectoire de l'aéronef sera située dans un plan vertical puisqu'il n'y a pas d'accélération faisant sortir le vecteur vitesse de ce plan. Le modèle dynamique de comportement en vol du Dornier est non-linéaire contrôlé. Le mouvement longitudinal d'un avion peut être décrit par les équations suivantes :

$$\left\{ \begin{array}{l} \dot{V} = -g \sin(\theta - \alpha) - \frac{1}{2m} \rho S V^2 (C_x^0 + C_{x\alpha}(\alpha - \alpha_0) + C_{x\delta_m}(\delta_m - \delta_{m_0})), \\ \dot{\alpha} = \frac{2}{2mV + \rho S l V C_{z\dot{\alpha}}} \left\{ mVq + mg \frac{\cos(\theta - \alpha)}{V} - \frac{1}{2} \rho S V^2 (C_z^0 + C_{z\alpha}(\alpha - \alpha_0) \right. \\ \quad \left. + C_{zq} \frac{ql}{V} + C_{z\delta_m}(\delta_m - \delta_{m_0})) \right\}, \\ \dot{q} = \frac{1}{2B} \rho S l V^2 \left\{ C_m^0 + C_{m\alpha}(\alpha - \alpha_0) + C_{mq} \frac{ql}{V} + C_{m\dot{\alpha}} \frac{2l}{2mV^2 + \rho S l V^2 C_{z\dot{\alpha}}} \left[mVq \right. \right. \\ \quad \left. \left. + mg \frac{\cos(\theta - \alpha)}{V} - \frac{1}{2} \rho S V^2 (C_z^0 + C_{z\alpha}(\alpha - \alpha_0) + C_{zq} \frac{ql}{V} + C_{z\delta_m}(\delta_m - \delta_{m_0})) \right] \right. \\ \quad \left. + C_{m\delta_m}(\delta_m - \delta_{m_0}) \right\}, \\ \dot{\theta} = q. \end{array} \right. \quad (3.71)$$

où α_0 est la valeur d'équilibre de l'incidence α , nous prenons $\alpha_0 = 4.45^\circ$. L'entrée δ_m représente le braquage de gouverne de profondeur qui déplace un volet à l'arrière de l'avion créant ainsi un couple de tangage tendant à faire cabrer ou piquer l'avion. La valeur d'équilibre de l'entrée δ_m est notée δ_{m_0} . Dans la suite, nous notons $u = \delta_m - \delta_{m_0}$ comme l'entrée du modèle.

Les paramètres que nous allons estimer sont $p = (C_{z\dot{\alpha}}, C_{zq}, C_{m\dot{\alpha}}, C_{mq})$. Ce sont les dérivées de stabilité aérodynamique. Selon nos hypothèses, ces coefficients sont constants et nous supposons que leurs valeurs exactes sont respectivement : $p = (1.8, 5, -5, -22)$ qui reflètent les dynamiques de performance aéronautique.

Selon des contraintes de géométries et des contraintes de modèle, nous admettons que

$$\begin{aligned} |u(t) + 2.6| &\leq 1.6 \text{ deg}, (\delta_{m_0} = -2.6 \text{ deg}), \\ 2m &\leq z(t_f) \leq 3m, \end{aligned} \quad (3.72)$$

où $z(t_f)$ représente l'altitude du modèle vers la fin de vol (en unité mètre).

L'entrée utilisée dans notre modèle a pour le but de comparer les résultats de l'estimation paramétrique lorsque nous avons une entrée constante, une entrée classique et une entrée dite optimale (ce terme sera précisé dans un prochain chapitre) obtenue dans l'article [Jaubertie 2013].

Nous ne considérerons que le cas causal pour le problème d'estimation d'état et l'estimation de paramètres.

3.6.2 Estimation d'état du modèle

Dans notre cas, pour une raison de simplicité, le vecteur d'état est observé en totalité mais nous supposons que ces observations sont perturbées par des bruits de mesures inconnus $e(t_j, p)$ vérifiant l'équation suivante :

$$e(t_j, p) = y(t_j) - y^m(t_j, p), \quad \text{pour } j = 1, \dots, N, \quad (3.73)$$

où y représente le vecteur des mesures et y^m celui des observations à partir du modèle et où $e(t_j, p)$ correspond aux bruits de mesures inconnus mais dont les bornes inférieures

et supérieures sont connues ou admissibles compte tenu des caractéristiques du capteur correspondant. Ces bornes sont déterminées à partir d'informations relevées sur les instruments de mesure. Dans notre cas, elles sont supposées incluses dans $[E]$ où :

$$[E] = [-0.0447, 0.0447] \times [-0.0044, 0.0044] \times [-0.0044, 0.0044] \times [-0.0044, 0.0044]. \quad (3.74)$$

Les états initiaux sont inclus dans :

$$[X_0] = [28.48, 28.52] \times [0.1094, 0.1174] \times [-0.004, 0.004] \times [0.0384, 0.0464]. \quad (3.75)$$

Les paramètres sont donnés avec 5% d'erreur relative par rapport à leurs vraies valeurs. Ils sont considérés inclus dans :

$$[P_0] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-23.1, -20.9]. \quad (3.76)$$

Ensuite, nous faisons l'estimation d'état via le critère de démarrage en regardant si une des longueurs d'intervalles d'états estimés dépasse la longueur admissible (la longueur d'erreur admissible pour chaque état). L'entrée constante a été utilisée ici, voir la Figure 3.23(a) :

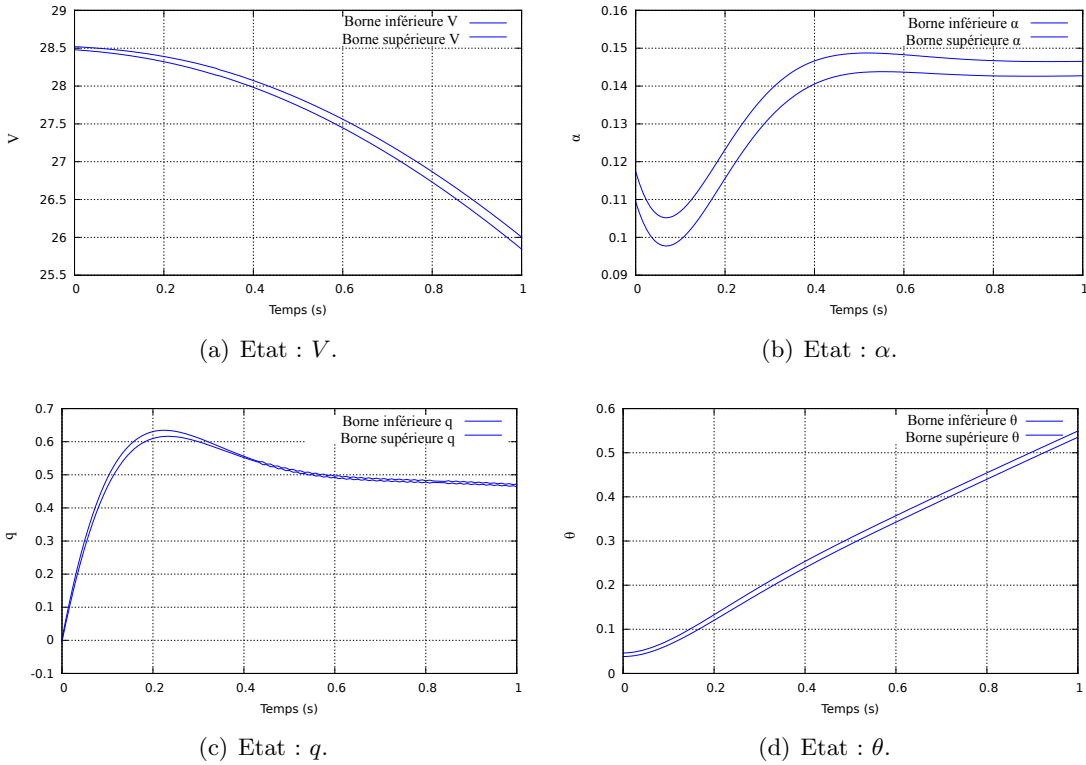


FIGURE 3.22 – L'estimation d'état du modèle d'avion.

3.6.3 Estimation de paramètres du modèle

Dans ce cas, les paramètres du modèle ne sont plus supposés connus mais encadrés par un domaine borné, nous gardons les suppositions sur les observations perturbées, et nous essayons de chercher le domaine de paramètres le plus petit possible à partir d'un ensemble de mesures données.

A partir de l'algorithme 7, nous vérifions toujours que les sorties du modèle sont incluses dans les mesures :

$$e_{inf} \leq y - y^m(t, p) \leq e_{sup}. \quad (3.77)$$

Si un pavé $[p]$ permet d'encadrer via le modèle du système les mesures, nous disons que ce pavé $[p]$ est admissible.

Selon les configuration réelles, les états initiaux sont considérés inclus dans le pavé :

$$[X_0] = [28.48, 28.52] \times [0.1094, 0.1174] \times [-0.004, 0.004] \times [0.0384, 0.0464]. \quad (3.78)$$

Nous générons nos mesures enveloppées par la matrice $[E]$. Ces valeurs sont prises à partir des caractéristiques des capteurs choisis autour des valeurs initiales d'états.

$$[E] = [-0.0447, 0.0447] \times [-0.0044, 0.0044] \times [-0.0044, 0.0044] \times [-0.0044, 0.0044]. \quad (3.79)$$

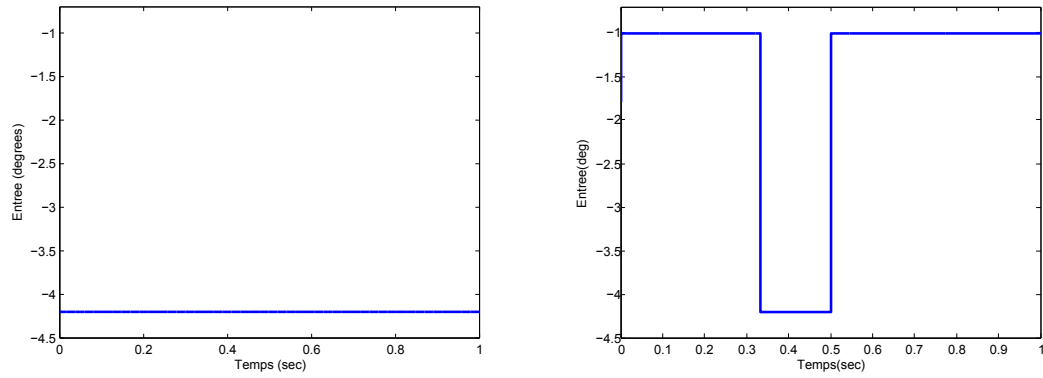
Le pavé initial pour les paramètres est supposé avec une incertitude de 5% autour de la valeur exacte :

$$[P_0] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-23.1, -20.9]. \quad (3.80)$$

La précision de découpage a été fixée à $\varepsilon = (0.01, 0.05, 0.05, 0.1)$. Les mesures sont disponibles à chaque $\frac{1}{12}$ de seconde pour la durée totale de 1 seconde avec 12 points de mesures, ils sont générés avec le vecteur de paramètres : $p = (C_{z\dot{\alpha}}, C_{zq}, C_{m\dot{\alpha}}, C_{mq}) = (1.8, 5, -5, -22)$.

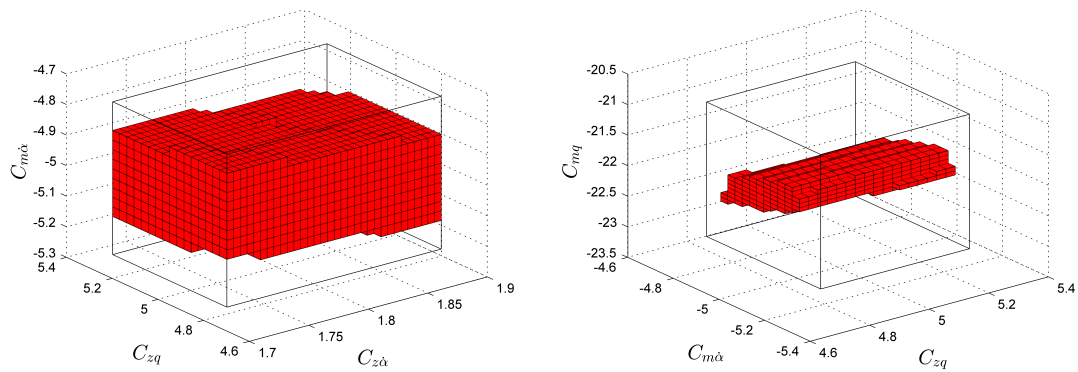
L'entrée u peut être représentée par la Figure 3.23(a) et la Figure 3.23(b).

Avec l'entrée constante 3.23(a), nous obtenons la Figure 3.24 qui représente les pavées de paramètres acceptables, la Figure 3.25 qui représente les pavées de paramètres incertains et la Figure 3.26(a) qui représente les pavées de paramètres rejetables. Ces résultats ont pris 21859 secondes (environ 4 h) du temps de calcul sur un ordinateur Intel i7, 3.2GHz sous Linux.



(a) Entrée constante pour le modèle d'avion. (b) Entrée pour le modèle d'avion du type classique.

FIGURE 3.23 – Les différentes entrées pour le modèle d'avion.



(a) Domaines admissibles de paramètre $C_{z\alpha}$, C_{zq} et (b) Domaines admissibles de paramètre C_{zq} , $C_{m\alpha}$ et C_{mq} .

FIGURE 3.24 – La représentation des domaines admissibles en 3D.

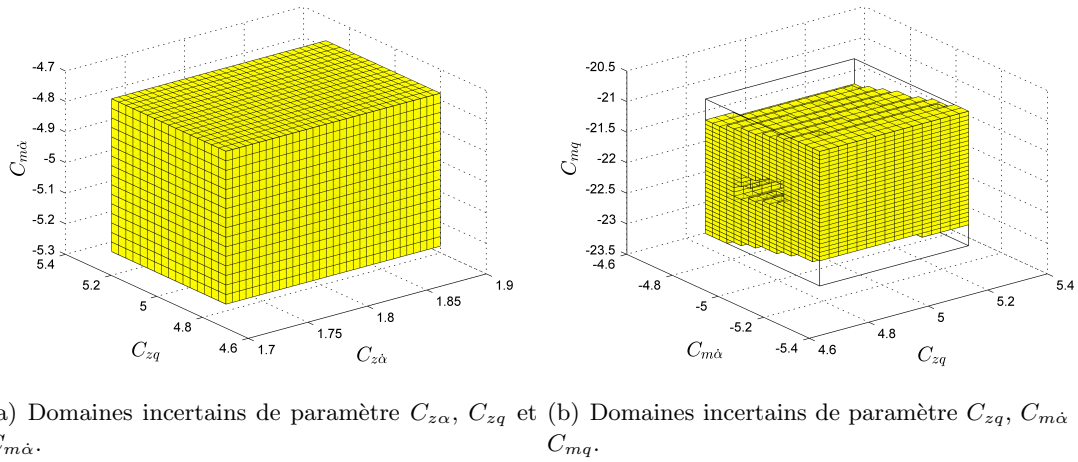


FIGURE 3.25 – La représentation des domaines incertains en 3D.

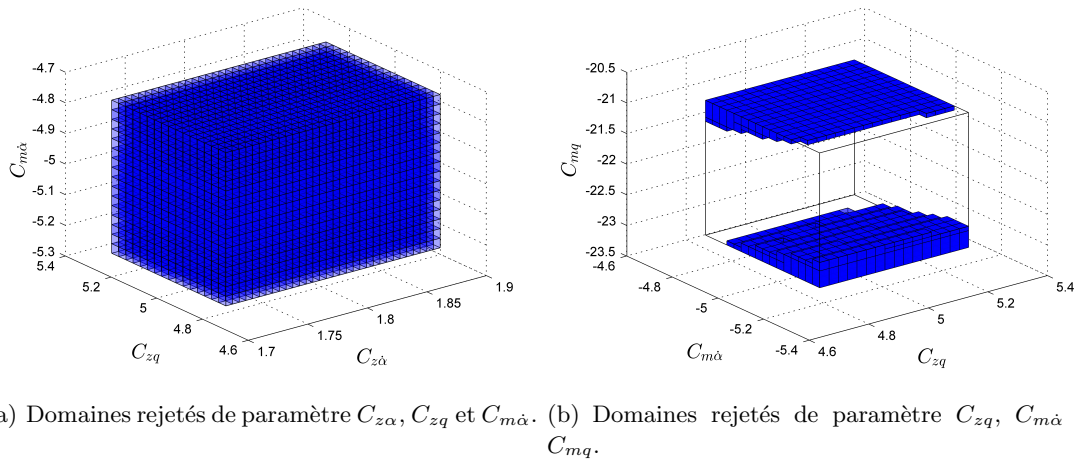
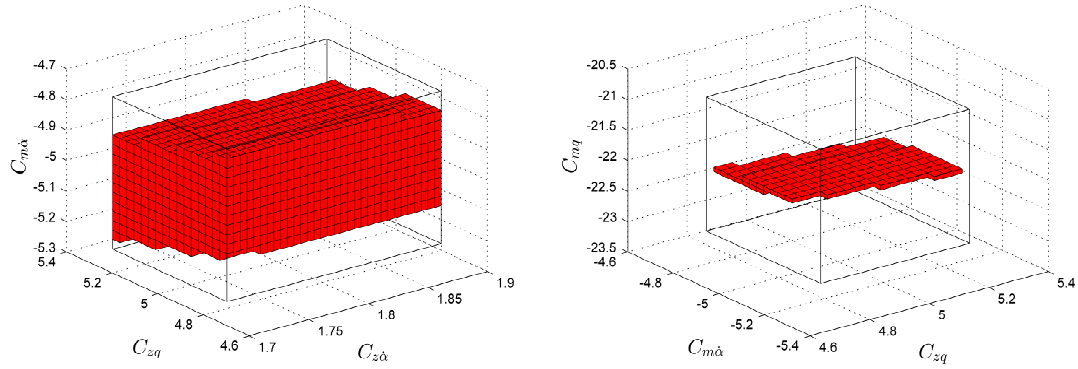


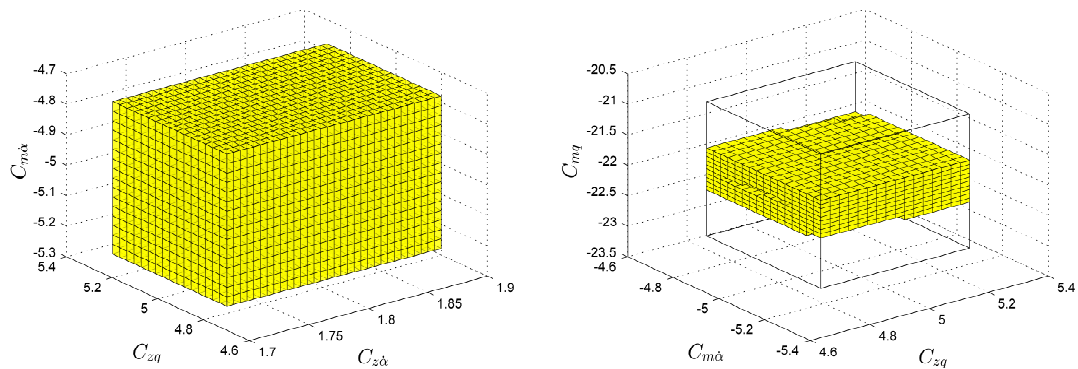
FIGURE 3.26 – La représentation des domaines rejetés en 3D.

Avec l'entrée classique 3.23(b), nous obtenons la Figure 3.27 qui représente les pavés de paramètres acceptables, la Figure 3.28 qui représente les pavés de paramètres incertains et la Figure 3.29 qui représente les pavés de paramètres rejetables. Ces résultats ont pris 13859 secondes (environ 4 h) du temps de calcul sur un ordinateur Intel i7, 3.2GHz de Linux :



(a) Domaines admissibles de paramètre $C_{z\alpha}$, C_{zq} et (b) Domaines admissibles de paramètre C_{zq} , $C_{m\alpha}$ et C_{mq} .

FIGURE 3.27 – La représentation des domaines admissibles en 3D.



(a) Domaines incertains de paramètre $C_{z\alpha}$, C_{zq} et (b) Domaines incertains de paramètre C_{zq} , $C_{m\alpha}$ et C_{mq} .

FIGURE 3.28 – La représentation des domaines incertains en 3D.

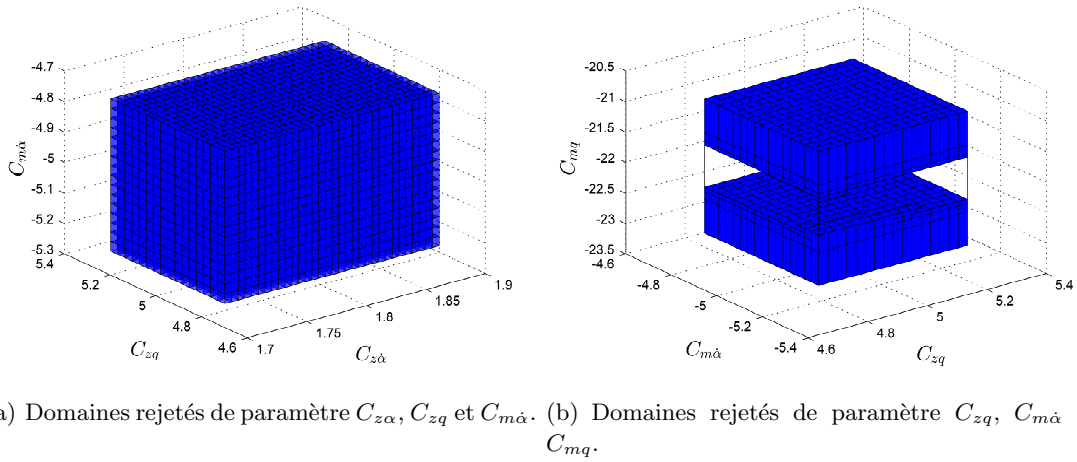


FIGURE 3.29 – La représentation des domaines rejetés en 3D.

Un autre essai est fait avec l'entrée représentée par la Figure 3.30, entrée proposée dans l'article [Jaubert [2013](#)]. Nous voulons mettre en évidence ici, l'impact du choix de l'entrée sur l'estimation dans ce modèle d'avion [Li [2015c](#)].

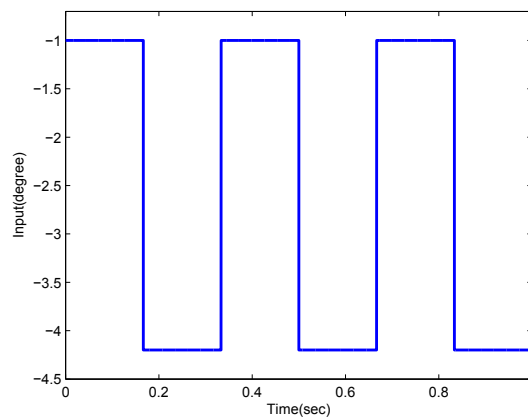
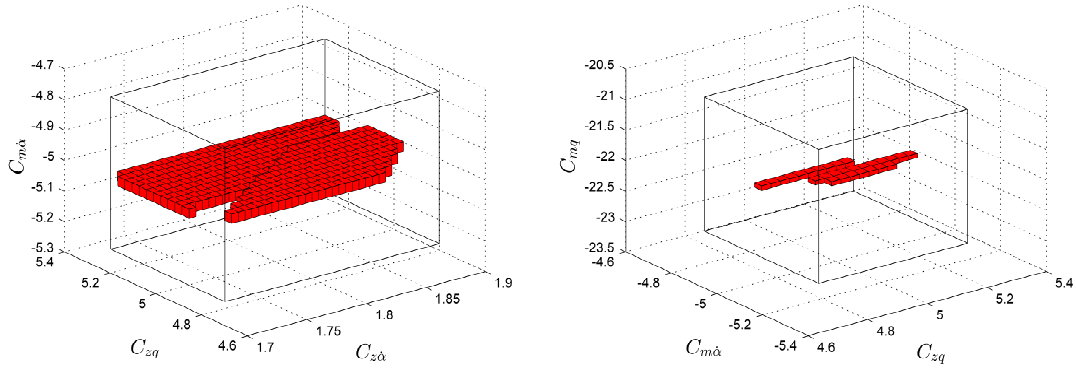


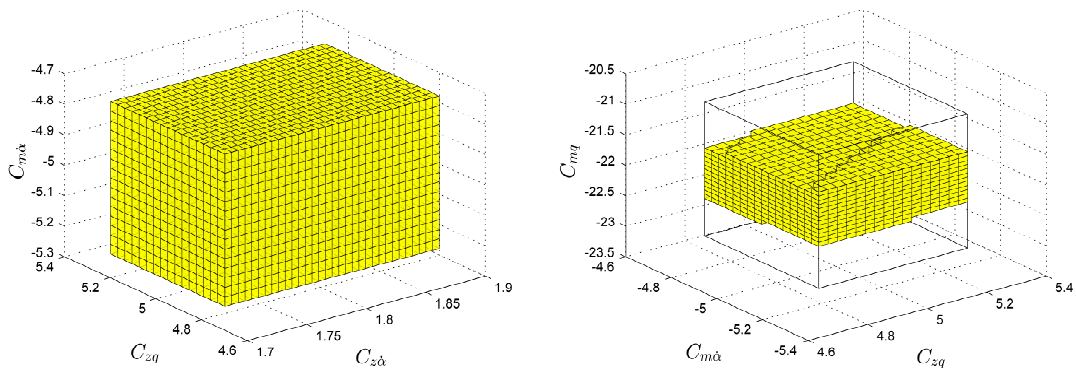
FIGURE 3.30 – Entrée pour le modèle d'avion dite optimale.

Nous reprenons le même processus de l'estimation de paramètres que précédemment, avec 19922 secondes de calcul, nous obtenons les Figures 3.31, 3.32 et 3.33 :



(a) Domaines admissibles de paramètre $C_{z\alpha}$, C_{zq} et $C_{m\alpha}$.
 (b) Domaines admissibles de paramètre C_{zq} , $C_{m\alpha}$ et C_{mq} .

FIGURE 3.31 – La représentation des domaines admissibles en 3D.



(a) Domaines admissibles de paramètre $C_{z\alpha}$, C_{zq} et $C_{m\alpha}$.
 (b) Domaines admissibles de paramètre C_{zq} , $C_{m\alpha}$ et C_{mq} .

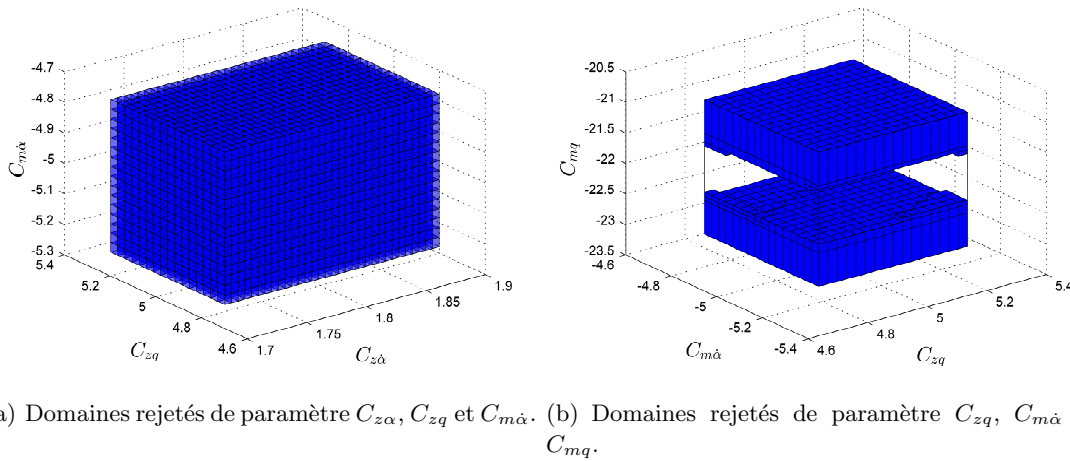
FIGURE 3.32 – La représentation des domaines incertains en 3D.

TABLE 3.5 – Volume de $p_1p_2p_3$.

Volume	Admissible	Incertain	Rejeté
Constante	0.1215	1.0651	0.1170
Non optimale	0.0386	0.4061	0.2693
Optimale	0.0028	0.5587	0.1882

TABLE 3.6 – Volume de $p_2p_3p_4$.

Volume	Admissible	Incertain	Rejeté
Constante	1.7325	13.0176	2.8499
Non optimale	0.4719	4.9641	12.1640
Optimale	0.0347	6.8286	10.7368



(a) Domaines rejetés de paramètre $C_{z\alpha}$, C_{zq} et $C_{m\alpha}$. (b) Domaines rejetés de paramètre C_{zq} , $C_{m\alpha}$ et $C_{m\gamma}$.

FIGURE 3.33 – La représentation des domaines rejeteables en 3D.

Nous résumons les résultats dans les Tableaux 3.5, 3.6.

Comme les domaines sont tous superposés, alors nous faisons un produit pour chaque groupe de paramètres obtenus dans le tableau 3.7.

TABLE 3.7 – Produit de $p_1p_2p_3p_4$.

Produit	Admissible	Incertain	Rejeté
Constante	0.0097	0.0732	0.0160
Classique	0.0027	0.0279	0.0684
Optimale	1.9501×10^{-4}	0.0384	0.0604

Nous voyons qu'avec une entrée optimale obtenue dans le sens classique [Jaubert [2002](#)], le volume du domaine admissible a été largement réduit. Mais le volume des boîtes rejetées est plus large que celui obtenu par l'entrée classique. L'entrée optimale obtenue dans le cas statistique ne rend pas des résultats satisfaisants par rapport aux entrées classiques. Une entrée optimale dans le contexte d'analyse par intervalles, va nous permettre d'obtenir les résultats premièrement avec plus de précision (la quantité

de boîte admissible et boîte incertaine) et deuxièmement en raccourcissant le temps de calcul, cela implique aussi qu'elle est capable d'enlever des quantités importantes sur les boîtes rejetables.

Dans ces figures, nous voyons quand même que le paramètre $C_{z\hat{\alpha}}$ reste encore mal estimé. Cela est peut être dû au fait que :

- Les erreurs sur les mesures sont grandes alors avec ces incertitudes, l'inversion ensembliste ne peut plus encadrer la bonne solution de manière précise et le nombre de domaines incertains sera plus important,
- L'entrée n'est pas idéale dans cette configuration pour estimer ce paramètre.

Dans le chapitre 4, nous allons ensuite proposer quelques critères afin de trouver une entrée plus adaptée à estimer les paramètres à la fois ou une entrée optimale à estimer certains paramètres. Cette méthodologie ouvre la porte de la planification d'expérience d'une entrée optimale, où les sensibilités seront très utiles.

Planification d'expérience pour les systèmes non-linéaires

Contents

4.1 Principe de la planification d'expériences	80
4.1.1 Critère de A-optimalité-intervalle	82
4.1.2 Critère M-optimalité-intervalle	87
4.1.3 Critère D-optimalité-intervalle	89
4.1.4 Critère Migmag-optimalité-intervalle	90
4.2 Application au modèle d'avion	92
4.2.1 Effet élémentaire des optimalités différentes	94
4.2.2 Résultat de l'estimation de paramètres	96
4.3 Conclusion	102

Lors de l'estimation de paramètres à partir de données expérimentales avec un modèle, il est important de sélectionner les conditions expérimentales les plus pertinentes, afin d'obtenir des résultats d'estimation les plus précis possibles. Pour cette estimation, nous obtiendrons ainsi l'information la plus pertinente. C'est la démarche de planification d'expérience ou d'optimisation d'essais. Elle permet de définir les conditions d'expériences les plus intéressantes parmi un ensemble admissible, cet ensemble traduisant les diverses contraintes (sorties, entrées, ...) que le protocole expérimental doit vérifier afin de pouvoir être effectivement mis en œuvre. La planification d'expériences constitue une étape essentielle car une expérience mal conçue peut rendre vaine toute tentative d'interprétation des données recueillies sur le système. Afin de résoudre le problème d'optimisation d'essais, il faut définir un critère d'optimalité (c'est-à-dire une grandeur scalaire) relié au but poursuivi qui est ici l'amélioration des mesures utilisées en estimation. Ce critère est ensuite optimisé par rapport à l'ensemble des variables disponibles à l'expérimentateur, tout en respectant l'ensemble des contraintes.

L'objectif de ce chapitre est de construire des outils numériques dans le cadre de l'analyse par intervalles permettant d'optimiser les essais d'un système décrit par des équations différentielles ordinaires. Nous ne nous intéressons pas à l'identification de la structure du modèle parmi un ensemble de structures mais à celle des paramètres de la structure retenue. Dans une première partie, nous parlerons du principe général de la planification d'expériences. Nous rappelons la notion de matrice d'information de Fisher. L'optimisation d'essais utilise cette matrice afin de trouver des conditions optimales qui maximisent ces variations en espérant avoir plus d'information. Dans le cadre de l'analyse par intervalles, nous montrerons un résultat similaire, sauf que la représentation de

l'information change. Nous proposons de nouveaux critères d'optimisation d'essais pour des modèles dynamiques non linéaires, dans le contexte de l'analyse par intervalles. Dans une deuxième partie, nous comparons ces différents critères sur un modèle d'avion qui a été déjà traité dans le cas stochastique [Jaubert [2002](#)].

4.1 Principe de la planification d'expériences

Considérons le système décrit par les équations d'état suivantes :

$$\begin{cases} \dot{x}(t) &= f(x(t), p, u(t)), \\ y^m(t) &= g(x(t), p), \\ x(t_0) &\in [X_0], \\ p &\in [P_0], \end{cases} \quad (4.1)$$

où $x(t) \in \mathbb{R}^{n_x}$ et $y^m(t) \in \mathbb{R}^{n_y}$ représentent respectivement les variables d'états et les sorties du modèle. $p \in \mathbb{R}^{n_p}$ et $u(t) \in U_{ad}$ représentent les variables de paramètres et l'entrée. L'entrée est continue par morceau. Les fonctions f et g sont non linéaires et continûment différentiables par rapport à x , p et u , $[X_0]$ est le domaine a priori connu pour la condition initiale $x(t_0)$ et $[P_0]$ est le domaine initial pour le vecteur de paramètre $p \in \mathbb{R}^{n_p}$. Les sorties mesurées $y(t_j)$ sont supposées disponibles à chaque instant $t_j \in \{t_1, \dots, t_N\}$.

Puisque nous sommes dans le contexte de l'estimation paramétrique à erreurs bornées, les valeurs admissibles des paramètres p peuvent être reliées avec les données expérimentales $y(t)$ de la façon suivante :

$$\underline{e}_i \leq y(t_i) - y^m(t_i, p, u) \leq \bar{e}_i, \quad (4.2)$$

où \underline{e}_i et \bar{e}_i sont considérés comme des bornes d'erreur admissible, u est l'entrée du système.

Nous supposons aussi qu'il existe des contraintes sur l'entrée $u(t)$ et éventuellement la sortie $y(t, p, u)$. Ces contraintes sont décrites sous la forme :

$$\begin{aligned} |u(t) - u_0| &\leq \mu \quad \forall t, \\ |y_k(t, p, u) - y_{k0}| &\leq \eta_k \quad \forall t, k = 1, \dots, n_y. \end{aligned} \quad (4.3)$$

Le but de la planification d'expérience pour l'estimation de paramètres est de trouver les entrées admissibles et optimales qui donnent plus de détails sur les résultats d'estimation : minimiser le domaine cherché des paramètres, augmenter la précision des paramètres, réduire le temps utilisé pour le processus d'estimation, etc. Selon les différents critères, nous pouvons trouver plein de conditions pour l'estimation de paramètre. Dans le contexte ensembliste exploitant les outils d'analyse par intervalles, comme les résultats de l'estimation paramétriques sont représentés sous forme d'intervalle, souvent par une forme géométrique, des boîtes ou des cubes, nous adoptons les techniques de planification pour réduire le domaine cherché ayant donné le pavé de paramètres recherchés.

Dans le cas classique stochastique les erreurs $e(t_i)$ sont assimilées à des variables aléatoires gaussiennes indépendantes de moyenne nulle, de variance égale à Σ et de matrice de variance covariance σ . La caractérisation de l'incertitude sur l'estimée des paramètres est alors souvent donnée par la matrice de Fisher suivante :

Définition 8. La matrice d'information de Fisher est donnée par :

$$F(p, u) = \sum_{i=1}^{N_t} \left(\frac{\partial y(t_i, p, u)}{\partial p} \right)^T \Sigma^{-1} \frac{\partial y(t_i, p, u)}{\partial p}, \quad (4.4)$$

où N_t est le nombre de mesure effectuées sur le système et Σ est la matrice de covariance des erreurs.

Il faut remarquer que la caractérisation par la matrice de Fisher dans le cas non linéaire suppose de nombreuses approximations. La dérivée sur la sortie du système par rapport aux paramètres influence la valeur de la matrice d'information de Fisher. La caractéristique de cette matrice décrit par exemple : A-optimalité, D-optimalité, etc, implique l'ensemble d'information apporté. A partir de la définition précédente, nous voyons que la valeur de $F(p, u)$ est influencée par les instants t_i , les paramètres p et l'entrée u . Dans le contexte de la planification d'expériences pour l'estimation paramétrique, nous pouvons nous intéresser à la recherche des instants et des entrées qui optimisent dans un certain sens une fonction de cette matrice. L'information obtenue aux instants où cette fonction est la plus grande est intuitivement plus pertinente qu'aux instants où la fonction est petite. Ceci nous conduit à la notion de planification d'expérience par rapport au temps, cela donnera les points optimaux pour faire l'échantillonnage. De même, par rapport à l'entrée u , nous obtenons des valeurs diverses et la planification d'expérience d'entrée nous donne l'entrée optimale pour estimer au mieux les paramètres. "Mieux" est dit dans le sens de précision. Avec cette idée nous proposons dans la suite une démarche de planification de l'expérience d'entrée dans le cas à erreur bornée, où apparaît cette dérivée sur la sortie du modèle par rapport aux paramètres. Notons que cela peut être utilisé facilement pour planifier les instants optimaux et les paramètres optimaux.

Notons que les termes $\frac{\partial y_i}{\partial p_j}(p, u)$ sont des solutions du système différentiel suivant, couplé avec l'équation 4.1 :

$$\begin{cases} \frac{d}{dt} \frac{\partial x}{\partial p_j} = \sum_{i=1}^{n_x} \left[\frac{\partial f}{\partial x_i} \frac{\partial x_i}{\partial p_j} \right] + \frac{\partial f}{\partial p_j}, \\ \frac{\partial y}{\partial p_j} = \sum_{i=1}^{n_x} \left[\frac{\partial g}{\partial x_i} \frac{\partial x_i}{\partial p_j} \right] + \frac{\partial g}{\partial p_j} \quad j = 1, \dots, n_p, \\ \frac{\partial x}{\partial p_j}(0) = 0. \end{cases} \quad (4.5)$$

La solution de ce système nécessite une augmentation de complexité de calcul d'EDO, mais il est incontournable pour le calcul des sensibilité des sorties par rapport aux paramètres.

Dans les sections qui suivent, nous allons mettre en évidence différents critères pour la minimisation du domaine des paramètres initialement cherché dans le contexte à erreur borné. Notons que ces critères peuvent non seulement être utilisés pour l'estimation de paramètres faite par l'inversion ensembliste, SIVIA par exemple, mais ils sont aussi efficaces pour l'estimation faite par l'approche d'optimisation. Les entrées optimales ne

sont pas le seul but de la planification, les conditions initiales et le nombre de points d'échantillonnage peuvent être aussi considérés par cette approche. Une analyse d'effet élémentaire sera appliquée sur différents critères choisis afin de réduire le nombre de points concernés.

4.1.1 Critère de A-optimalité-intervalle

Pour un système dynamique non linéaire donné, nous pouvons toujours écrire le système (4.1) sous forme d'intervalles. Nous pouvons trouver par exemple via la fonction d'inclusion de la forme centrée sur la fonction d'observation $[y](t, [p]) = [g]([x](t, [p]), [p])$, une relation déployée par rapport aux paramètres. Selon [Jaulin 2001a], la forme centrée donne souvent des intervalles plus petits quand les largeurs d'intervalles sont faibles, c'est celle ci que nous choisissons dans la suite. Supposons que $i = 1, \dots, n_x$ où n_x est la dimension de l'état, pour chaque sortie du système à l'instant t_k , nous cherchons les ensembles de paramètres qui satisfont la relation suivante :

$$y_i^m(m, u, t_k) + \sum_{j=1}^{n_p} ([p_j] - m_j) \left[\frac{\partial y_i^m}{\partial p_j} \right]([p], u, t_k) \subset [\underline{y}_i^k, \overline{y}_i^k], \quad (4.6)$$

où m_j est la valeur au milieu de $[p_j]$, m est le vecteur milieu des paramètres $[p]$, k est l'indice de temps avec $k = 1, \dots, n_t$. L'indice m pour y_i^m signifie la sortie du modèle dans l'équation et $[y_i^k] = [\underline{y}_i^k, \overline{y}_i^k]$ les mesures à erreurs bornées.

Avec les notations ci-dessus, nous notons $[S_{i,j}^k] = \left[\frac{\partial y_i^m}{\partial p_j} \right]([p], u, t_k)$, et la ligne $[S_i] \in \mathbb{R}^{1 \times n_p}$ formée par les entrées $[S_{i,j}^k] = ([S_{i,1}^k], \dots, [S_{i,n_p}^k])$ à chaque instant t_k avec le vecteur $[p] = ([p_1], \dots, [p_{n_p}])^T$ et $n_p = \dim([p])$. Nous rappelons que la matrice $[I]_\lambda$ est définie par :

$$[I]_\lambda = \begin{bmatrix} [1 - \lambda, 1 + \lambda] & \cdots & [0 - \lambda, 0 + \lambda] \\ \vdots & \ddots & \vdots \\ [0 - \lambda, 0 + \lambda] & \cdots & [1 - \lambda, 1 + \lambda] \end{bmatrix}. \quad (4.7)$$

Le terme $[1 - \lambda, 1 + \lambda]$ n'apparaît que dans la ligne diagonale, les autres éléments dans cette matrice sont tous sous forme $[0 - \lambda, 0 + \lambda]$ où $\lambda \in]0, 1[$. Nous utilisons $[v^k]$ pour représenter $[\underline{y}^k - y^m([p], u, t_k), \overline{y}^k - y^m([p], u, t_k)]$. Avec ces notations, nous avons la proposition suivante :

Proposition 5. Si $[S^k]^T [S^k]$ est inversible, alors nous avons :

$$[I]^k([p] - m) \subset ([S^k]^T [S^k])^{-1} \cdot [S^k]^T [v^k]. \quad (4.8)$$

où $[I]^k$ est une matrice "proche de la matrice identité intervalle" pour tout $k \in (1, \dots, n_t)$.

Démonstration. nous avons :

$$[S_i^k]([p] - m) + y_i^m(m, u, t_k) \subset [\underline{y}_i^k, \overline{y}_i^k]. \quad (4.9)$$

Comme cette relation doit être établie à chaque sortie et à l'instant t_k , nous noterons $[S^k] \in \mathbb{R}^{n_y \times n_p}$ la matrice dont les lignes sont les n_y lignes $[S_i^k]$. Nous avons donc :

$$[S^k]([p] - m) \subset [v^k]. \quad (4.10)$$

Nous multiplions cette relation par la matrice $[S^k]^T \in \mathbb{R}^{n_p \times n_y}$ transposée, nous obtenons :

$$[S^k]^T [S^k] \cdot ([p] - m) \subset [S^k]^T \cdot [v^k]. \quad (4.11)$$

Cette relation est valable pour tous les t_k où $k = 1, \dots, n_t$. Supposons que la matrice $[S^k]^T [S^k]$ soit inversible, alors nous multiplions par la matrice inverse de $[S^k]^T [S^k]$ comme nous l'avons définie dans la proposition 1 du chapitre 2.

Selon le lemme 3, nous multiplions une matrice notée $([S^k]^T [S^k])^{-1}$, alors nous avons :

$$([S^k]^T [S^k])^{-1} ([S^k]^T [S^k]) \cdot ([p] - m) \subset ([S^k]^T [S^k])^{-1} \cdot [S^k]^T [v^k], \quad (4.12)$$

et l'équation (4.12) peut être écrite :

$$[I^k]([p] - m) \subset ([S^k]^T [S^k])^{-1} \cdot [S^k]^T [v^k], \quad (4.13)$$

où $[I^k]$ représente l'inclusion $([S^k]^T [S^k])^{-1} ([S^k]^T [S^k])$.

Avec cette inclusion nous allons maintenant établir une première majoration de la norme du vecteur intervalle $[I^k]([p] - m)$. Cette majoration nécessitera l'introduction du lemme technique suivant.

Lemme 4. Soit $[A]$ une matrice intervalle, $[v^k]$ un vecteur intervalle et \tilde{v}^k le vecteur de composantes réelles égales à $(\text{mag}([v_j^k]))_j$ nous avons :

$$\|[A][v^k]\|^2 \leq \| [A] \|_F^2 \| [v^k] \|_2^2. \quad (4.14)$$

Démonstration. Nous notons : $[A] = ([A_{ij}])_{ij}$,

$$\|[A][v^k]\|^2 = \max_i \left| \sum_j [A_{ij}][v_j^k] \right|^2. \quad (4.15)$$

Nous avons $|[A_{ij}][v_j^k]| = |[A_{ij}]| |v_j^k|$ et $[A_{ij}] = [\underline{a}_{ij}, \bar{a}_{ij}]$.

En effet, pour la partie gauche :

$$\begin{aligned} |[A_{ij}][v_j^k]| &= |\min(\underline{a}_{ij}\underline{v}_j^k, \underline{a}_{ij}\bar{v}_j^k, \bar{a}_{ij}\underline{v}_j^k, \bar{a}_{ij}\bar{v}_j^k), \max(\underline{a}_{ij}\underline{v}_j^k, \underline{a}_{ij}\bar{v}_j^k, \bar{a}_{ij}\underline{v}_j^k, \bar{a}_{ij}\bar{v}_j^k)| \\ &= \max(|\underline{a}_{ij}\underline{v}_j^k|, |\underline{a}_{ij}\bar{v}_j^k|, |\bar{a}_{ij}\underline{v}_j^k|, |\bar{a}_{ij}\bar{v}_j^k|), \end{aligned} \quad (4.16)$$

puis la partie droite :

$$|[A_{ij}]| |v_j^k| = \max(|\underline{a}_{ij}\underline{v}_j^k|, |\underline{a}_{ij}\bar{v}_j^k|, |\bar{a}_{ij}\underline{v}_j^k|, |\bar{a}_{ij}\bar{v}_j^k|). \quad (4.17)$$

Alors :

$$\left| \sum_j [A_{ij}] [v_j^k] \right|^2 = \sum_j |[A_{ij}]| |[v_j^k]|^2. \quad (4.18)$$

Nous posons, $[A_i]$ le vecteur ligne de la matrice $[A]$ donc $[A_i] = ([A_i, j])_j$ et $A_i = (|[A_{ij}]|)_j$, vecteur à composantes réelles (positives) et $\tilde{v}^k = (|[v_j^k]|)_j$, élément à composantes réelles (positives). Ainsi le produit scalaire de A_i par \tilde{v} est :

$$\sum_j |[A_{ij}]| |[v_j^k]| = \langle A_i, \tilde{v}^k \rangle. \quad (4.19)$$

Avec l'inégalité de Cauchy Schwartz :

$$\langle A_i, \tilde{v}^k \rangle^2 = |\langle A_i, \tilde{v}^k \rangle|^2 \leq \|A_i\|_2^2 \|\tilde{v}^k\|_2^2. \quad (4.20)$$

Ainsi :

$$\|[A][v^k]\|^2 \leq \max_i (\|A_i\|_2^2 \|\tilde{v}^k\|_2^2) = (\max_i \|A_i\|_2^2) \|\tilde{v}^k\|_2^2. \quad (4.21)$$

Mais

$$\max_i (\|A_i\|_2^2) \leq \sum_i \|A_i\|_2^2 = \sum_{i,j} |[A_{ij}]|^2 = |\text{tr}([A][A]^\top)|. \quad (4.22)$$

Donc

$$\|[A][v^k]\|^2 \leq |\text{tr}([A][A]^\top)| \cdot \|\tilde{v}^k\|_2^2 = \|A\|_F^2 \cdot \|\tilde{v}^k\|_2^2. \quad (4.23)$$

Démonstration terminée. \square

Démonstration de la proposition terminée. \square

Ceci nous conduit à la proposition suivante :

Proposition 6. *Pour tout $k \in (1, \dots, n_t)$*

$$\|[I^k]([p] - m)\| \leq \|([S^k])^T [S^k]^{-1} \cdot ([S^k])^T \|_F \| [v^k] \|. \quad (4.24)$$

et

$$\|[I^k]([p] - m)\| \leq \sqrt[2]{|\text{tr}([I]([S^k])^T [S^k]^{-1})|} \| [v^k] \|. \quad (4.25)$$

Démonstration. Nous avons :

$$\|[I^k]([p] - m)\| \leq \|([S^k])^T [S^k]^{-1} \cdot ([S^k])^T \|_F \| [v^k] \|. \quad (4.26)$$

En introduisant la matrice d'intervalle $[A] = ([S^k])^T [S^k]^{-1} \cdot [S^k]^T$ et sachant que $([A][B])^T = [B]^T [A]^T$, nous avons :

$$[A][A]^T = ([S^k])^T [S^k]^{-1} ([S^k])^T [S^k]^{-1} = ([S^k])^T [S^k]^{-1} ([S^k])^T [S^k]^{-1}, \quad (4.27)$$

donc

$$[A][A]^T = [I^k] \cdot (([S^k]^T[S^k])^{-1})^T, \quad (4.28)$$

or $([A][A]^T)^T = [A] \cdot [A]^T$, donc

$$[A][A]^T = ([S^k]^T[S^k])^{-1} \times ([I^k])^T = ([S^k]^T[S^k])^{-1} \times [I^k]. \quad (4.29)$$

Nous avons la trace

$$tr([A][A]^T) = \sum_i \left(\sum_j [A_{ij}][A_{ji}]^T \right) = \sum_i \left(\sum_j [A_{ij}][A_{ij}] \right), \quad (4.30)$$

où les éléments de $[A]$ sont notés $[A_{ij}]$ et $[A_{ji}]^T = [A_{ij}]$, donc

$$tr([A][A]^T) = \sum_{i,j} [A_{ij}]^2, \quad (4.31)$$

qui est un intervalle. Nous utilisons la norme Frobenius sur une matrice d'intervalle définie par [Rump 2014] :

$$\|[A]\|_F = \sqrt{tr([A][A]^T)}, \quad (4.32)$$

où nous appelons que la magnitude d'un intervalle est définie par $|[J]| = \max(|\underline{J}|, |\overline{J}|)$. Notons que l'intervalle $[A_{ij}]^2$ a toujours ses bornes positives. Donc,

$$\overline{\sum_{i,j} [A_{ij}]^2} = \sum_{i,j} \overline{([A_{ij}]^2)} \Rightarrow \sum_{i,j} |[A_{ij}]^2| = |\sum_{i,j} [A_{ij}]^2|. \quad (4.33)$$

Donc nous avons posé $\|[A]\|_F = \sqrt{tr([A][A]^T)} = \sqrt{\sum_{i,j} |[A_{ij}]^2|}$.

Nous avons $[I^k]([p] - m) \subset [A][v^k]$. Nous savons que $\|[z]\| = \max_i |z_i|$. et donc $\|[z]\|^2 = \max_i |z_i|^2$.

Le lemme 4 permet de terminer la preuve de cette proposition. □

La proposition suivante va nous permettre d'isoler la trace afin d'extraire un critère.

Proposition 7. Pour tout $k \in \{1, \dots, n_t\}$

$|tr([I^k]([S^k]^T[S^k])^{-1})| \leq \lambda_{\max}(|[I]_\lambda|) |tr([S^k]^T[S^k])^{-1}|$ avec $\lambda_{\max}(|[I]_\lambda|) = 1 + n_p \lambda$ la valeur maximale de valeur propre de $|[I]_\lambda|$.

Démonstration. Pour tout $k \in 1, \dots, n_t$, il existe λ_k tel que $[I^k] \subset [I]_{\lambda_k} [I^k]([S^k]^T[S^k])^{-1} \subset [I]_{\lambda_k} ([S^k]^T[S^k])^{-1}$. Mettons $\lambda = \max_{k \in 1, \dots, n_t} \lambda_k$, nous avons $[I]_{\lambda_k} \subset [I]_\lambda$. Mettons $[C] = ([S^k]^T[S^k])^{-1}$ et $[c_{ij}]$ sont les éléments de la matrice $[C]$, nous obtenons l'inégalité suivante :

$$|tr([I^k][C])| \leq |tr([I]_\lambda [C])| \leq |tr(|[I]_\lambda| |[C]|)| \leq \lambda_{\max}(|[I]_\lambda|) |tr([C])|,$$

puisque les matrices $| [I]_\lambda |$ et $| [C] |$ sont symétriques [Mehra 1976]. Démonstration terminée. \square

Des propositions précédentes, nous déduisons le résultat suivant avec les notations déjà données :

Proposition 8. *Pour tout $k \in \{1, \dots, n_t\}$*

$$\| [I^k]([p] - m) \|^2 \leq \lambda_{\max}(| [I]_\lambda |) | \text{tr}([S^k]^T [S^k]^{-1}) | \| \tilde{v}^k \|^2. \quad (4.34)$$

Ainsi un critère possible pour l'entrée optimale dans le sens où le domaine de recherche est le plus contractant peut être défini par :

$$J_1(u) = \sum_k | \text{tr}([S^k]^T [S^k]^{-1}) |.$$

Remarquons que $[S^k][S^k]$ est similaire à la matrice d'information dans le cas classique [Friedrich 2006] [Saltelli 2008].

Parmi toutes les entrées possibles, une entrée optimale est recherchée pour pouvoir rendre le domaine de recherche des paramètres contractant (minimiser l'ensemble de domaine rejetable) et alors de plus réduire le temps de calcul au cours de l'inversion ensembliste. De l'étude qui précède, nous allons extraire dans un premier temps, deux critères possibles. Ces deux critères sont basés sur la trace. Nous notons $[S^k]$ la fonction d'inclusion de sensibilité, qui représente $[\frac{\partial y}{\partial p}](p, u, t_k)$ à l'instant t_k . Nous proposons les deux définitions suivantes.

Définition 9. *Nous appelons critère de A-optimalité-intervalle la fonction :*

$$J_1(u) = \sum_k (\text{mag}(\text{tr}([S^k]^T [S^k]^{-1}))). \quad (4.35)$$

Nous appelons critère de trace-intervalle la fonction :

$$J_2(u) = \sum_k (\text{mig}(\text{tr}([S^k]^T [S^k]))). \quad (4.36)$$

Définition 10. *Nous dirons qu'une entrée u^* est A-optimale-intervalle si elle vérifie :*

$$u^* = \underset{u \in U_{ad}}{\text{Argmin}} J_1(u), \quad (4.37)$$

nous dirons qu'une entrée u^ est trace-optimale-intervalle si elle vérifie :*

$$u^* = \underset{u \in U_{ad}}{\text{Argmax}} J_2(u). \quad (4.38)$$

Le deuxième critère est proposé pour éviter le calcul d'inverse d'une matrice d'intervalle, mais il faut vérifier que les matrices $([S^k]^T [S^k])$ sont inversibles. Pour cela nous pouvons calculer le déterminant de cette matrice qui est un intervalle et vérifier que 0 n'y appartient pas.

4.1.2 Critère M-optimalité-intervalle

Nous pouvons d'une part, chercher une entrée optimale à partir d'une matrice d'intervalle en appliquant une norme, d'autre part, nous pouvons aussi en chercher une à partir d'une matrice ponctuelle appartenant à la matrice intervalle $[S^k]$. Avec les mêmes notations que la section précédente, la matrice ponctuelle choisie est le milieu de la matrice $[S^k]$, en considérant toujours la matrice $[S^k] \in \mathbb{IR}^{n_y \times n_p}$. Supposons que $i = 1, \dots, n_x$ où n_x est la dimension de l'état, pour chaque sortie du système à l'instant t_k , nous avons :

$$y_i^m(m, u, t_k) + \text{mid}([S^k]) \cdot ([p] - m) \subset y_i^m(m, u, t_k) + [S^k]([p] - m) \subset [\underline{y}_i^k, \bar{y}_i^k]. \quad (4.39)$$

En reprenant les notations précédentes, nous avons :

$$\text{mid}([S^k]) \cdot ([p] - m) \subset ([v_1^k], \dots, [v_{n_y}^k])^T, \quad (4.40)$$

où $\text{mid}([S^k]) \in \mathbb{M}^{n_y \times n_p}$. Il faut remarquer qu'ici la matrice est une matrice à coefficients réels ce qui facilite les calculs matriciels. Alors nous avons la proposition suivante :

Proposition 9. *Si $\text{mid}([S^k])^T \text{mid}([S^k])$ est inversible, alors nous avons :*

$$[p] - m \subset ((\text{mid}([S^k]))^T \times (\text{mid}([S^k])))^{-1} (\text{mid}([S^k]))^T [v^k]. \quad (4.41)$$

Démonstration. nous multiplions les deux côtés de l'équation (4.40) par $(\text{mid}([S^k]))^T$ pour obtenir

$$(\text{mid}([S^k]))^T \cdot (\text{mid}([S^k]))([p] - m) \subset (\text{mid}([S^k]))^T [v^k], \quad (4.42)$$

et sous la condition que $(\text{mid}([S^k]))^T \text{mid}([S^k])$ soit inversible nous avons :

$$[p] - m \subset ((\text{mid}([S^k]))^T \times (\text{mid}([S^k])))^{-1} (\text{mid}([S^k]))^T [v^k]. \quad (4.43)$$

Démonstration terminée. □

Nous notons $A = [(\text{mid}([S^k]))^T \times (\text{mid}([S^k]))]^{-1} (\text{mid}([S^k]))^T$ qui est une matrice réelle, alors nous pouvons écrire :

$$[p] - m \subset A \cdot [v^k], \quad (4.44)$$

où $[p] \subset m + A \cdot [v^k]$. Pour que $[p] - m$ soit le plus petit possible, nous pouvons rendre $A \cdot [v^k]$ le plus petit possible. Nous connaissons $\| [v^k] \| = \max_i (|[v_i^k]|)$ et $\| [v_i^k] \| = \max(|[\underline{v}_i^k]|, |[\bar{v}_i^k]|)$.

Comme dans le paragraphe précédent nous avons le lemme qui cette fois porte sur une matrice réelle.

Lemme 5.

$$\| A[v^k] \| \leq \| A \|_F \cdot \| [v^k] \|_2. \quad (4.45)$$

Démonstration. Nous notons $A = (A_{ij})$,

$$A \cdot [v^k] = \sum_{j=1}^{n_p \times n_y} A_{ij} [v_j^k], \quad (4.46)$$

$$\|A \cdot [v^k]\| = \left\| \left(\sum_{j=1}^{n_p \times n_y} A_{ij} \cdot [v_j^k] \right) \right\| = \max_i \left(\left| \sum_{j=1}^{n_p \times n_y} A_{ij} \cdot [v_j^k] \right| \right). \quad (4.47)$$

Nous prenons le produit scalaire, alors :

$$\sum_j |A_{ij}| |[v_j^k]| = \langle A_i, \tilde{v}^k \rangle. \quad (4.48)$$

Rappelons que $[A_{i,j}]$ sont les éléments de la matrice $[A]$, le vecteur ligne de cette matrice est noté $[A_i] = [A_{i,j}]_j$ et nous notons A_i , le vecteur $(\text{mag}([A_{i,j}]))_j$ c'est un vecteur à composantes réelles (positives) et $\tilde{v}^k = (|[v_j^k]|)_j$, élément à composantes réelles (positives).

Avec l'inégalité de Cauchy Schwartz :

$$\langle A_i, [\tilde{v}^k] \rangle^2 = |\langle A_i, \tilde{v}^k \rangle|^2 \leq \|A_i\|_2^2 \|\tilde{v}^k\|_2^2, \quad (4.49)$$

$$\left| \sum_{j=1}^{n_t \times n_y} A_{ij} \cdot [v_j^k] \right|^2 \leq \sum_{j=1}^{n_t \times n_y} |A_{ij}|^2 \cdot |[v_j^k]|^2 \leq \left(\sum_{j=1}^{n_t \times n_y} |A_{ij}|^2 \right) \times \|[v^k]\|. \quad (4.50)$$

En conséquence,

$$\begin{aligned} \|A \cdot [v^k]\|^2 &\leq \max_i \left(\sum_{j=1}^{n_y} |A_{ij}|^2 \right) \times \|[v^k]\|^2, \\ &\leq \sum_{i=1}^{n_p} \sum_{j=1}^{n_y} |A_{ij}|^2 \times \|[v^k]\|^2, \\ &= \text{tr}(AA^T) \|[v^k]\|^2. \end{aligned} \quad (4.51)$$

Et puis, $\|A\|_F^2 = \text{tr}(A^T A)$. La démonstration terminée.

□

Ensuite, nous avons :

$$A^T = \text{mid}([S^k])[(\text{mid}([S^k]))^T (\text{mid}([S^k]))^{-1}]^T.$$

Donc,

$$\begin{aligned} AA^T &= [(\text{mid}([S^k]))^T \times \text{mid}([S^k])^{-1}] [(\text{mid}([S^k]))^T \times \text{mid}([S^k])] [(\text{mid}([S^k]))^T \times (\text{mid}([S^k]))^{-1}]^T, \\ &= I \times [(\text{mid}([S^k]))^T \times (\text{mid}([S^k]))^{-1}]^T = [(\text{mid}([S^k]))^T \times (\text{mid}([S^k]))^{-1}]^T, \end{aligned} \quad (4.52)$$

où I est la matrice identité.

Nous savons que

$$\text{tr}(AA^T) = \text{tr}(((\text{mid}([S^k]))^T \times \text{mid}([S^k]))^{-1}). \quad (4.53)$$

alors nous pouvons maximiser la trace de $A^T A$ ce qui est équivalent à maximiser $\text{tr}[(\text{mid}([S^k]))^T \times \text{mid}([S^k])]$ après avoir vérifié l'inversibilité des matrices $(\text{mid}([S^k]))^T (\text{mid}([S^k]))$ afin d'estimer de manière efficace les paramètres.

Avec le lemme 5, nous avons la proposition suivante :

Proposition 10. *Pour tout $k = \{1, \dots, n_t\}$,*

$$\| [p] - m \|^2 \leq \| A[v^k] \|^2 \leq |\text{tr}(((\text{mid}([S^k]))^T \times \text{mid}([S^k]))^{-1})| \| [v^k] \|^2. \quad (4.54)$$

Définition 11. *Nous appelons M-optimalité-intervalle la fonction :*

$$J_3(u) = \sum_k \text{tr}((\text{mid}([S^k]))^T (\text{mid}([S^k]))^{-1}). \quad (4.55)$$

Définition 12. *Nous dirons qu'une entrée u^* est M-optimale-intervalle si elle vérifie :*

$$u^* = \underset{u \in U_{ad}}{\text{Argmin}} J_3(u). \quad (4.56)$$

Dans la pratique les entrées trouvées, sur les exemples traités dans cette thèse, sont les mêmes qu'avec le critère précédent.

4.1.3 Critère D-optimalité-intervalle

Nous reprenons les notations des paragraphes précédents. Dans le cas classique, l'information à l'instant t_k est contenue dans la matrice formée des produits $\frac{\partial y^m(t_k)}{\partial p_i} \times \frac{\partial y^m(t_k)}{\partial p_j}$,

$i, j \in (1, \dots, n_p)$, c'est-à-dire la matrice $\frac{\partial y^m(t_k)}{\partial p} \frac{\partial y^m(t_k)}{\partial p}^T$.

Nous avons les définitions suivantes [Hazewinkel 2001] :

Définition 13. *Dans un espace euclidien la gramienne d'un système de vecteur v_1, \dots, v_n est la matrice formée par les produits scalaires de ces vecteurs. Le déterminant de cette matrice est le déterminant de Gram aussi appelé gramien. Ainsi l'information est contenue dans la gramienne du système formé par le système des vecteurs de sensibilité :*

$$\frac{\partial y_1^m(t_1)}{\partial p_i}, \dots, \frac{\partial y_n^m(t_1)}{\partial p_i}, \dots, \frac{\partial y_1^m(t_N)}{\partial p_i}, \dots, \frac{\partial y_{n_y}^m(t_N)}{\partial p_i}, \text{ pour } i = 1, \dots, n_p. \quad (4.57)$$

Si de plus cette matrice est inversible alors le Gramien représente le carré du volume du parallélotope formé par les vecteurs de sensibilité. La D-optimalité classique correspond à l'optimisation de ce volume. Ici nous étendons cette démarche dans le contexte intervalle. Nous noterons avec les notations des paragraphes précédents :

$$F([p], u) = \sum_{k=1}^{N_t} (([S^k])^T [S^k]). \quad (4.58)$$

Une forme avec la pondération est :

$$F([p], u) = \sum_{k=1}^{N_t} ([S^k])^T \sigma[S^k]. \quad (4.59)$$

Nous proposons le critère suivant :

Définition 14. Nous appelons critère de *D-optimalité-intervalle* la fonction :

$$J_4(u) = \det(\text{mig}(F([p], u))). \quad (4.60)$$

Définition 15. Nous dirons qu'une entrée u^* est *D-optimale-intervalle* si elle vérifie :

$$u^* = \underset{u \in U_{ad}}{\text{Argmax}} J_4(u). \quad (4.61)$$

Remarque dans [Jaubertie 2013] l'utilisation d'un déterminant intervalle est faite mais le critère utilisé est un peu différent, le contexte intervalle n'étant pas le même puisque pour le bruit de mesure des informations statistiques était pris en compte.

Pour trouver le déterminant, nous avons essayé différentes méthodes. La première tentative est d'utiliser la méthode proposée dans [Jaulin 2005] via l'approche inégalité matricielle linéaire (LMI en anglais). Mais, la borne inférieure reste toujours négative. L'autre approche est basée sur le calcul de la trace, sachant que les termes diagonaux sont toujours positifs et de plus le nombre de paramètres à estimer par l'approche ensembliste via analyse par intervalle ne doivent pas dépasser 5, alors selon le théorème de Cayley-Hamilton, pour des matrices de dimension 2×2 , 3×3 , 4×4 , nous pouvons toujours calculer le déterminant via la trace de matrice. Mais, cette méthode n'aboutit pas à un résultat satisfaisant. Enfin, nous utilisons le calcul classique du déterminant en prenant la valeur maximale pour le déterminant-intervalle.

4.1.4 Critère Migmag-optimalité-intervalle

Dans les paragraphes 4.1.1 et 4.1.2, nous avons pris le parti de contracter le vecteur intervalle $[p] - m$ en seul bloc. Dans ce qui suit nous allons travailler paramètre par paramètre. Nous en déduisons un critère d'optimalité : le Migmag-optimalité-intervalle. Ce critère sera utilisé pour améliorer la qualité de l'estimation des paramètres que l'on a eu plus de difficulté à estimer [Li 2015b].

Reprenons l'équation (4.6) à un instant t , nous pouvons tirer $[p_i]$:

$$[p_i] \subset \left(\frac{[y_k] - y_k^m(m, u(t)) - \sum_{j \neq i} ([p_j] - m_j) \left[\frac{\partial y_k^m}{\partial p_j} \right]([p], u)}{\left[\frac{\partial y_k^m}{\partial p_i} \right]([p], u)} + m_i \right),$$

ce qui introduit un nouveau domaine pour $[p_i]$. Mais en déplaçant m_i , nous avons aussi :

$$[p_i] - m_i \subset \left(\frac{[y_k] - y_k^m(m, u(t)) - \sum_{j \neq i} ([p_j] - m_j) \left[\frac{\partial y_k^m}{\partial p_j} \right]([p], u)}{\left[\frac{\partial y_k^m}{\partial p_i} \right]([p], u)} \right),$$

pour tout $i \in 1, \dots, n_p$, $[p_i] - m$ est inclus dans un intervalle symétrique.

Nous avons le résultat suivant :

Proposition 11. *Avec les notations qui précèdent nous avons pour tout $i \in 1, \dots, n_p$,*

$$[p_i] - m_i \subset \left(\frac{[y_k] - y_k^m(m, u(t)) - \sum_{j \neq i} ([p_j] - m_j) \text{mag}([\frac{\partial y_k^m}{\partial p_j}]([p], u))}{\text{mig}([\frac{\partial y_k^m}{\partial p_i}]([p], u))} \right).$$

Démonstration. Puisque $[p_j] - m_j$ est symétrique pour tout $j \in \{1, \dots, n_p\}$ avec le lemme 1 nous avons :

$$[p_i] - m_i \subset \left(\frac{[y_k] - y_k^m(m, u(t)) - \sum_{j \neq i} ([p_j] - m_j) \text{mag}([\frac{\partial y_k^m}{\partial p_j}]([p], u))}{[\frac{\partial y_k^m}{\partial p_i}]([p], u)} \right). \quad (4.62)$$

Intuitivement pour minimiser une fraction, il suffit de maximiser son dénominateur et minimiser son numérateur. C'est cette idée que nous utiliserons dans la suite. Notons :

$$[\frac{\partial y_k^m}{\partial p_i}]([p], u) = [S_{k,i}] = [\underline{S}_{k,i}, \overline{S}_{k,i}]. \quad (4.63)$$

Si $0 \notin [\underline{S}_{k,i}, \overline{S}_{k,i}]$, nous avons

$$[\underline{S}_{k,i}, \overline{S}_{k,i}]^{-1} = [\frac{1}{\overline{S}_{k,i}}, \frac{1}{\underline{S}_{k,i}}]. \quad (4.64)$$

Si $\underline{S}_{k,i} > 0$ et nous devons maximiser $|\underline{S}_{k,i}| = \text{mig}([S_{k,i}])$, c'est équivalent à minimiser $\frac{1}{|\underline{S}_{k,i}|}$.

Si $\overline{S}_{k,i} < 0$ et nous devons maximiser $|\overline{S}_{k,i}| = \text{mig}([S_{k,i}])$, c'est équivalent à minimiser $\frac{1}{|\overline{S}_{k,i}|}$.

Si $0 \in [\underline{S}_{k,i}, \overline{S}_{k,i}]$, aucune majoration sera faite, nous prendrons $\text{mig}([S_{k,i}]) = 0$.

Pour les éléments dont la somme est au numérateur, nous minimisons leurs magnitudes et puisque $[p_j] - m_j$ est symétrique, nous minimisons via le lemme 1 la magnitude de la sensibilité $[S_{j \neq i, i}]$. Ensuite pour le dénominateur nous maximisons les mignitudes des sensibilités $[S_{j, i}]$ selon le lemme 2 :

$$[p_i] - m_i \subset \left(\frac{[y_k] - y_k^m(m, u(t)) - \sum_{j \neq i} ([p_j] - m_j) \text{mag}([\frac{\partial y_k^m}{\partial p_j}]([p], u))}{\text{mig}([\frac{\partial y_k^m}{\partial p_i}]([p], u))} \right).$$

□

Rappelons que notre but est de minimiser la longueur de paramètre $[p_i]$. Mais, nous n'avons pas de connaissance sur les intervalles $[y_k] - y_k^m(m)$ obtenus, alors le seul facteur influant devient la valeur de $mig([S_{k,j}])$ et celle de $mag([S_{k,j}])$.

Définition 16. Nous appelons critère de Migmag-optimalité-intervalle la fonction suivante :

$$J_5(u) = \frac{\sum_{t=0}^{t_{max}} \sum_{k=1}^{n_y} \sum_{i=1}^{n_p} \sum_{j=1, j \neq i}^{n_p} mag([S_{k,j}](t))}{\sum_{t=1}^{t_{max}} \sum_{k=1}^{n_y} \sum_{j=1}^{n_p} mig([S_{k,j}](t))}, \quad (4.65)$$

où i, j variant de 1 à n_p , $[S](t)$ est la matrice de sensibilité à l'instant t avec $[S_{k,j}](t) = [\frac{\partial y_k^m}{\partial p_j}](t, [p], u)$.

Définition 17. Nous dirons qu'une entrée u^* est Migmag-optimale-intervalle lorsque :

$$u^* = \underset{u \in U_{ad}}{\operatorname{Argmin}} J_5(u). \quad (4.66)$$

Dans la section suivante, nous allons appliquer ces critères dans des configurations différentes sur le modèle d'avion afin d'illustrer les résultats.

4.2 Application au modèle d'avion

Dans cette section, le même système du modèle d'avion est considéré, voir la section 3.6.1. Les nouveaux critères proposés seront testés avec VSPODE [Lin 2007]. D'une part, c'est parce que ce modèle est plus complexe par rapport aux modèles que nous l'avons traités précédemment et d'une autre part, les résultats résolus par VNODE-LP et IOLAVABE ne sont pas satisfaisant en terme de pessimisme. Les conditions initiales pour les états sont supposés encadrés par :

$$[X_0] = [28.48, 28.52] \times [0.1094, 0.1174] \times [-0.004, 0.004] \times [0.0384, 0.0464]. \quad (4.67)$$

Les paramètres sont :

$$[P_0] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-23.1, -20.9]. \quad (4.68)$$

Pour le modèle d'avion, il existe des contraintes sur les vecteurs d'entrée et de sortie. Ces contraintes sont données sous la forme :

$$|u_j(t) + 2.6| \leq 1.6 \text{ deg}, \quad (4.69)$$

La programmation dynamique, introduite par R.E. Bellman vers 1957 [Bellman 1952], se prête bien à la résolution de ces types de problèmes. Elle permet la prise en compte de contraintes sur les entrées, sur les sorties, sur des éléments spécifiques (par exemple, pour un modèle d'avion, la dynamique des gouvernes) et a des aspects plus pratiques (comme la réalisation de braquage multi-gouvernes).

Dans cette partie, les contraintes (4.69) sont directement introduites. La durée de l'expérience est discrétisée en pas de longueurs égales, dits horizons. Sur chaque horizon (intervalle du temps), les composantes de l'entrée sont supposées constantes, et la résolution permet d'obtenir les trajectoires des sorties correspondantes. Les valeurs retenues pour l'entrée doivent vérifier (4.69). Chaque trajectoire obtenue à la fin d'un horizon est utilisée pour obtenir les conditions initiales de l'horizon suivant. L'espace des sorties admissibles, à un instant donné, peut être représenté par un hyperespace de dimension p dont les limites correspondent aux valeurs des contraintes de sorties. Cet espace est lui-même discrétisé en "boîte" ou "classe d'équivalence". Si des entrées différentes conduisent à des trajectoires "équivalentes", seule celle qui est optimale dans cette classe est retenue, les autres étant rejetées. Cette technique permet également de rejeter toute trajectoire ne satisfaisant pas les contraintes de type (4.69).

Sur la durée totale de l'essai, nous avons employé des entrées de type "créneaux". Nous testons un ensemble exhaustif d'entrées "créneaux" définies par :

$$u(t) = u_0 + \sum_{i=0}^r (a_i \varepsilon_i - a_{i-1} \varepsilon_{i-1}) h(t - t_i), \quad (4.70)$$

où h est la fonction de Heaviside, et u_0 une constante, éventuellement nulle. Nous définissons ε l'ensemble des paramètres de l'entrée "créneaux" par :

$$\varepsilon = \{\varepsilon_i \mid i = 0, \dots, r, \varepsilon_i \in \{-1, 0, +1\}\}, \quad (4.71)$$

où $\varepsilon_i \in \{-1, 0, +1\}$ car l'entrée, sur le i^{eme} horizon peut prendre trois valeurs $\{a_i, 0, -a_i\}$ avec $\varepsilon_{-1} = 0$, $i = 0, \dots, r$. Le paramètre a_i représente l'amplitude de chaque créneau, et les t_i sont les instants de changement de valeur de consigne pour l'entrée. Ils sont tels que $t_0 < t_1 < \dots < t_r$. Dans cette partie, ils représentent les instants de changement d'horizon et sont fixés. Les amplitudes a_i sont également fixées.

La durée du vol libre de la maquette dans le laboratoire B10 est de l'ordre de une seconde. La maquette utilisée pour les essais permet d'effectuer au plus quatre changements de position pour le braquage de gouverne en une seconde. Nous avons ainsi fixé le nombre d'horizons successivement égal à un, deux, trois, quatre et cinq. Nous testons également la méthode pour six horizons.

Le nombre de point $N = 120$ pendant $T = 1s$.

TABLE 4.1 – Valeur optimale des critères avec différents horizons.

Nb de horizon	Trace optimalité	M optimalité	D optimalité
2	3266.04	3723.93	79092122.79
3	3522.59	4111.19	123195123.60
4	3440.41	3981.55	100642580.56
5	3542.66	4125.19	176983565.88
6	3509.43	4111.07	147684446.66

Nous pouvons voir que la valeur maximale parmi ces horizons est toujours l'entrée avec 5 horizons. Selon les deux premiers critères, l'entrée en 6 horizons et l'entrée en 3

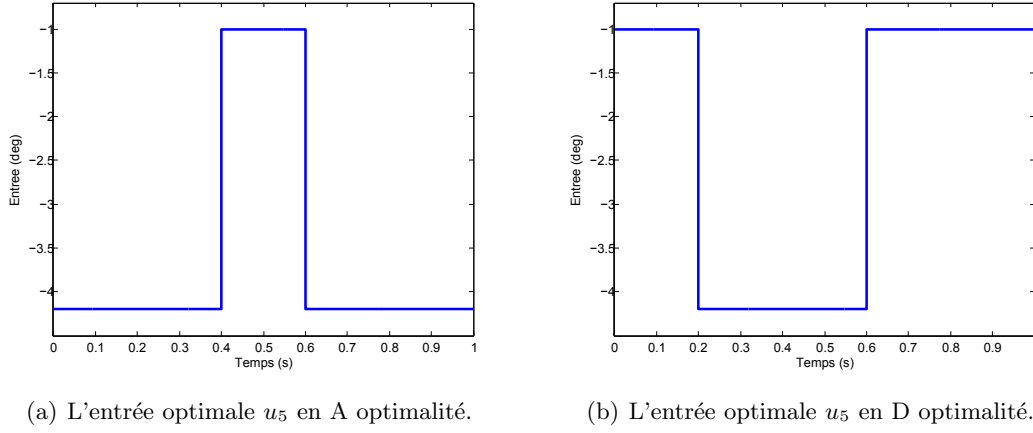


FIGURE 4.1 – Deux entrées optimales obtenues avec les critères A et D optimalité

horizons n'ont pas de grande différence, mais la valeur du critère de D optimalité varie beaucoup.

TABLE 4.2 – Type d'entrée obtenue.

Nb de horizon	Trace optimalité	D optimalité
2	9 [-4.2 -4.2]	7 [-4.2 -1]
3	21 [-4.2 -1 -4.2]	7 [-1 -4.2 -1]
4	75 [-4.2 -4.2 -1 -4.2]	25 [-1 -4.2 -4.2 -1]
5	225 [-4.2 -4.2 -1 -4.2 -4.2]	73 [-1 -4.2 -4.2 -1 -1]
6	657 [-4.2 -4.2 -1 -1 -4.2 -4.2]	218 [1 -4.2 -4.2 -1 -1 -2.6]

L'entrée optimale obtenue est montrée dans le tableau 4.2, les valeurs entre crochets du tableau 4.4 représente les valeurs obtenues sur chaque horizon, le nombre en avant est sa numérotation aux entrées possibles. Le type d'entrée obtenue par M optimalité n'est pas dans le tableau, car les résultats sont identiques à ceux obtenus avec le critère de trace-optimalité.

4.2.1 Effet élémentaire des optimalités différentes

SIVIA est un processus long quand le nombre de paramètres à estimer est nombreux (le nombre de paramètres supérieur à 5) [Raïssi 2004a] et aussi le domaine initial est large. Une analyse sur l'effet élémentaire, qui regarde les changements en variant les moments d'échantillonnage choisis est important pour minimiser le nombre de mesures. Sachant que le nombre de mesures s'élève proportionnellement avec le temps de calcul en utilisant l'approche SIVIA. Une étude sur la minimisation du nombre de points mesurés sans perdre beaucoup de précision en estimation de paramètres est nécessaire. L'analyse d'effet élémentaire nous rend cette possibilité.

Définition 18. *Considérons un modèle avec k entrées indépendantes x_i , $i = 1, \dots, k$, qui varie dans p niveaux sélectionnés d'un espace Ω . L'effet élémentaire pour l'entrée i*

est défini par :

$$EE_i = \frac{y(x_1, \dots, x_{i-1}, x_i + \Delta, \dots, x_k) - y(x_1, x_2, \dots, x_k)}{\Delta}. \quad (4.72)$$

où p est le nombre de niveau, Δ est la valeur de $\{\frac{1}{p-1}, \dots, 1 - \frac{1}{p-1}\}$,

Afin de mesurer la sensibilité de l' EE_i en variant p , [Morris 1991] a proposé d'utiliser la valeur moyenne de la distribution EE_i :

Définition 19. La mesure de sensibilité μ décrit la valeur moyenne de la distribution EE_i .

$$\mu = \frac{1}{n} \sum_{i=1}^n EE_i. \quad (4.73)$$

Ensuite, [Campolongo 1999] a proposé de remplacer la forme μ par μ^* qui est définie par la valeur moyenne des valeurs absolues des distributions EE_i .

Définition 20. La mesure de sensibilité μ^* décrit la valeur moyenne de la distribution $|EE_i|$.

$$\mu^* = \frac{1}{n} \sum_{i=1}^n |EE_i|. \quad (4.74)$$

En prenant les différents critères proposés précédemment, nous pouvons choisir le nombre de mesures en regardant la dégradation de μ^* .

4.2.1.1 Minimisation du nombre de point via la Trace optimalité

La minimisation du nombre de points mesurés sans introduire de l'effet négatif est le but de cette analyse. Le critère de trace est une somme de trace autour de points disponibles. Un ré-échantillonnage peut être fait sur le nombre de points proposés, ensuite, un tirage selon le nombre de points souhaités sera fait, la valeur de μ^* sera aussi indiquée. Nous négligeons l'effet d'interaction entre les sensibilités puisque ces paramètres sont indépendants, et aussi comme nous évaluons la variance de cette tendance, alors un indicateur par exemple le milieu de chaque EE_i sera pris en compte afin de simplifier la représentation. Par conséquent, l'effet élémentaire est calculé par :

$$EE_i = \frac{\sup(\text{trace}(S_{i+\Delta}^T S_{i+\Delta})) - \sup(\text{trace}(S_i^T S_i))}{\Delta}. \quad (4.75)$$

Une présentation graphique a été mise dans la Figure 4.2. La minimisation du nombre de point donne d'abord une augmentation de la valeur μ^* . Ensuite, après une augmentation, avec l'incrément de nombre de points par échantillonnage, la valeur μ^* diminue.

Sans perdre beaucoup dans la valeur moyenne μ^* , le nombre des points possibles sera choisi comme $N = 10$ dans la suite de l'estimation paramétrique.

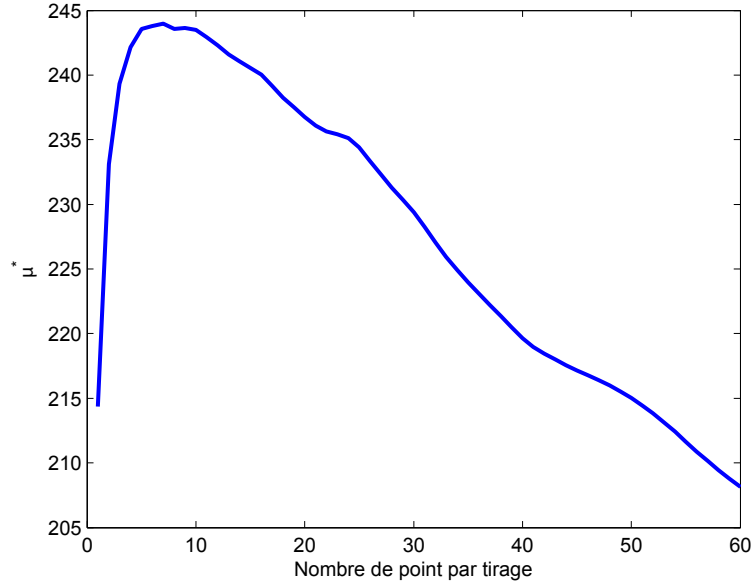
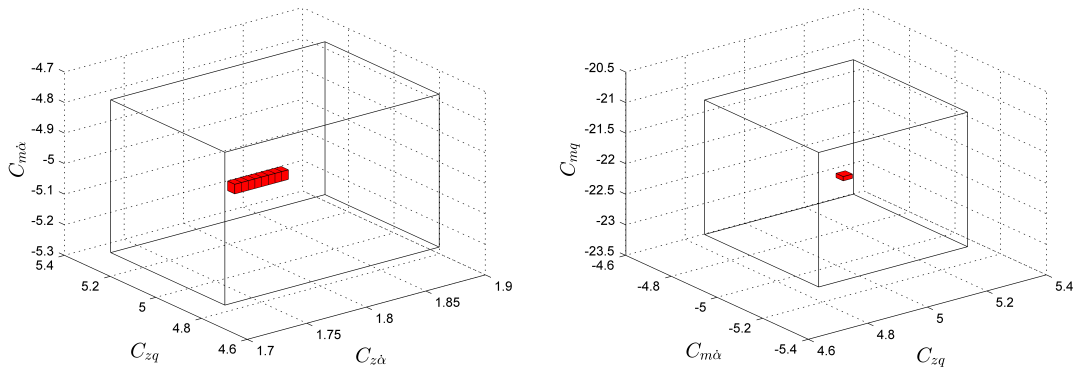


FIGURE 4.2 – La relation entre nombre de point par échantillonnage et la valeur μ^* .

4.2.2 Résultat de l'estimation de paramètres

Maintenant, nous allons appliquer ces deux entrées obtenues précédemment pour l'estimation paramétrique. Avec l'aide de l'analyse de l'effet élémentaire sur le nombre de point, les figures ci-dessous sont calculées via l'entrée obtenue par D-optimalité, le nombre de points d'échantillonnage est fixé $N = 10$, la précision de découpage est $\varepsilon = (0.1, 0.05, 0.05, 0.01)$ sur chaque paramètres respectivement :



(a) Domaines admissibles de paramètre $C_{z\alpha}$, C_{zq} et (b) Domaines admissibles de paramètre C_{zq} , $C_{m\dot{\alpha}}$ et C_{mq} .

FIGURE 4.3 – La représentation des domaines admissibles en 3D.

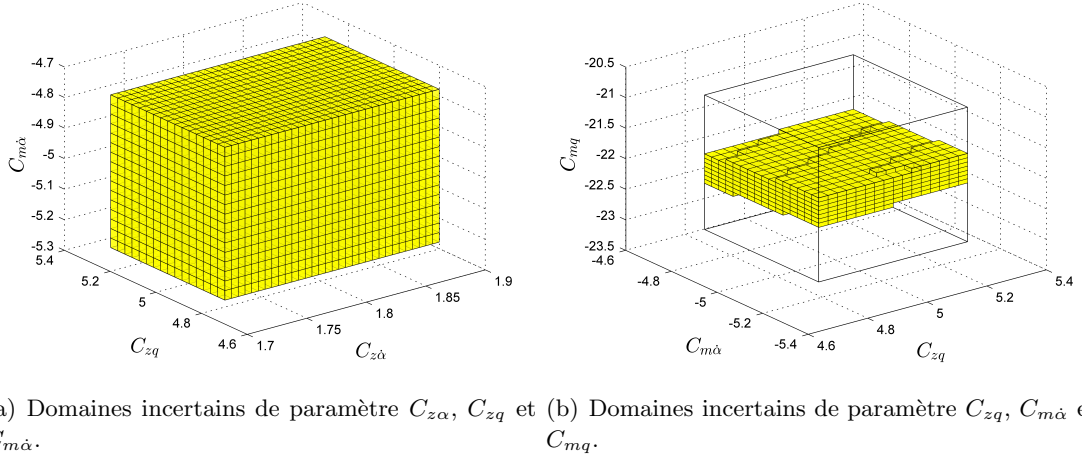


FIGURE 4.4 – La représentation des domaines incertains en 3D.

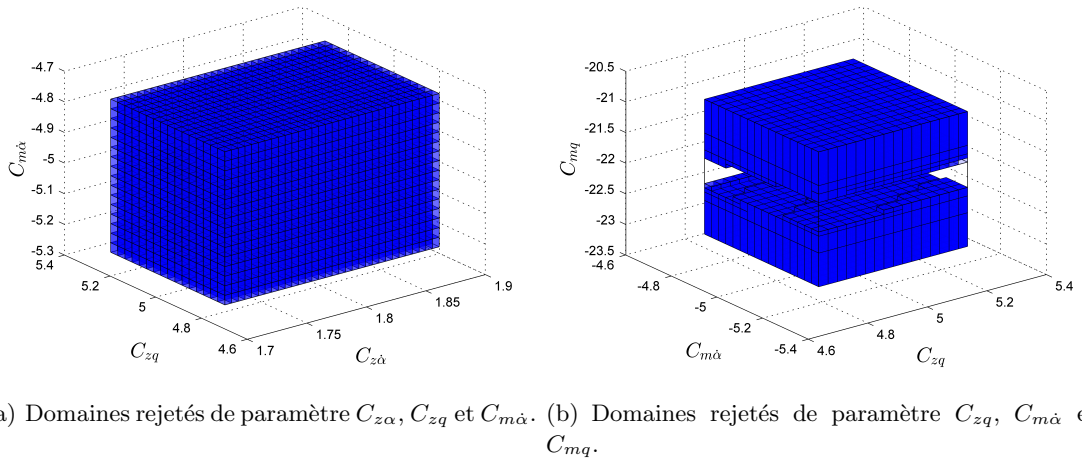
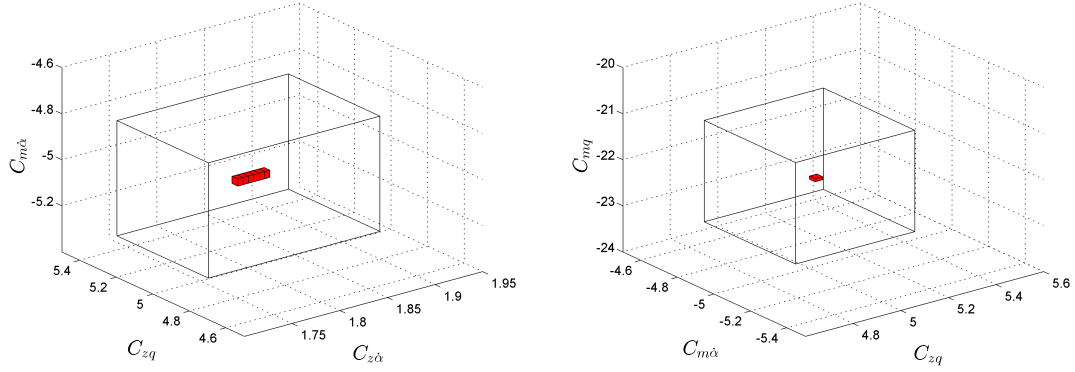


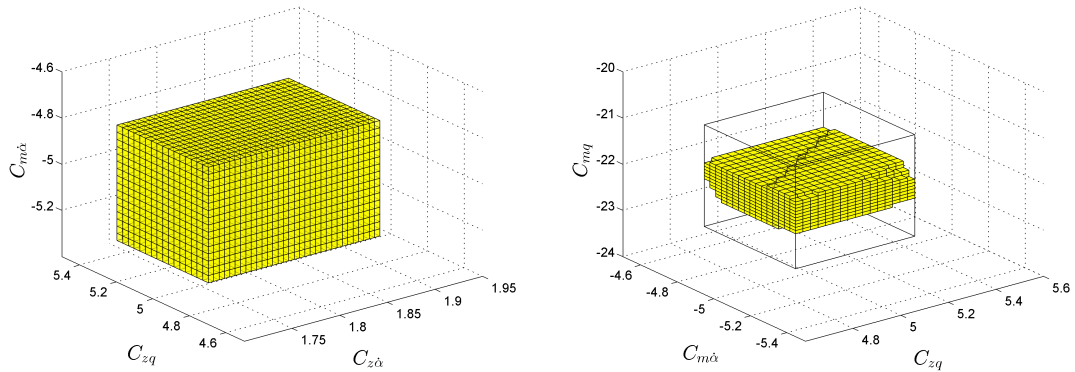
FIGURE 4.5 – La représentation des domaines rejetés en 3D.

Les figures ci-dessous sont calculés via l'entrée obtenue par trace-optimalité, le nombre d'échantillonnage est fixé $N = 10$:



(a) Domaines admissibles de paramètre $C_{z\alpha}$, C_{zq} et $C_{m\alpha}$.
 (b) Domaines admissibles de paramètre C_{zq} , $C_{m\alpha}$ et C_{mq} .

FIGURE 4.6 – La représentation des domaines admissibles en 3D.



(a) Domaines incertains de paramètre $C_{z\alpha}$, C_{zq} et $C_{m\alpha}$.
 (b) Domaines incertains de paramètre C_{zq} , $C_{m\alpha}$ et C_{mq} .

FIGURE 4.7 – La représentation des domaines incertains en 3D.

TABLE 4.3 – Volume de $p_1p_2p_3p_4$.

Volume	Admissible	Incertain	Rejeté
Constant	0.0097	0.0732	0.0160
Trace optimalité	2.2676e-06	0.0333	0.0657
D optimalité	3.0234e-06	0.0219	0.0771

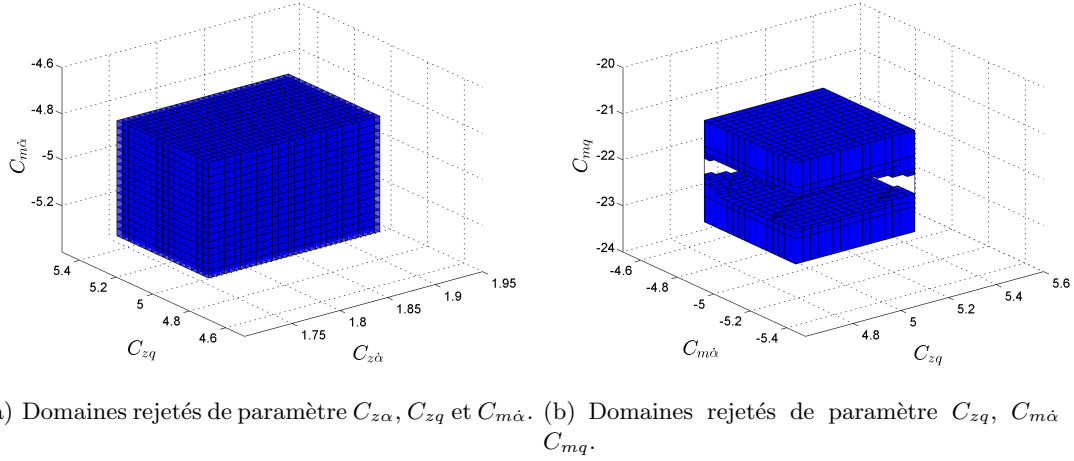


FIGURE 4.8 – La représentation des domaines rejetés en 3D.

Le volume sur ces ensembles est présentée dans le tableau 4.3.

Pour le critère MigMag d'optimalité, nous l'avons testé dans le cas où deux paramètres nécessitent d'améliorer l'estimation. Le premier paramètre est $C_{z\alpha} \in [1.71, 1.89]$ et le deuxième paramètre est $C_{zq} \in [4.75, 5.25]$. Après avoir fait la programmation dynamique, nous avons obtenu le Tableau 4.4 :

TABLE 4.4 – Valeur de la fonction coût $j(\Xi)$.

Nombre d'horizons	$j(\Xi)$	Entrée (degrés)
2	2.864	[-4.2 -4.2]
3	2.828	[-4.2 -2.6 -1]
4	2.809	[-4.2 -4.2 -2.6 -1]
5	2.791	[-4.2 -2.6 -2.6 -1 -1]
6	2.787	[-4.2 -1 -2.6 -1 -1 -1]

La dernière colonne représente les valeurs obtenues sur chaque horizon. Cette optimisation de $j(\Xi)$ conduit à une entrée optimale avec six horizons, qui est sous la forme :

$$\begin{aligned} \hat{u}(t) = & \delta_{m0} - aH(t - \tau_{0_6}) + 2aH(t - \tau_{1_6}) \\ & - aH(t - \tau_{2_6}) + aH(t - \tau_{3_6}), \end{aligned} \quad (4.76)$$

avec $a_{6_i} = 1.6$ degrés avec $i = 0, \dots, 5$ et, $t_{0_6} = 0s$, $t_{1_6} = 0.1667s$, $t_{2_6} = 0.3334s$, $t_{3_6} = 0.5001s$, $t_{4_6} = 0.6668s$, $t_{5_6} = 0.8335s$.

La fonction H est la fonction Heaviside.

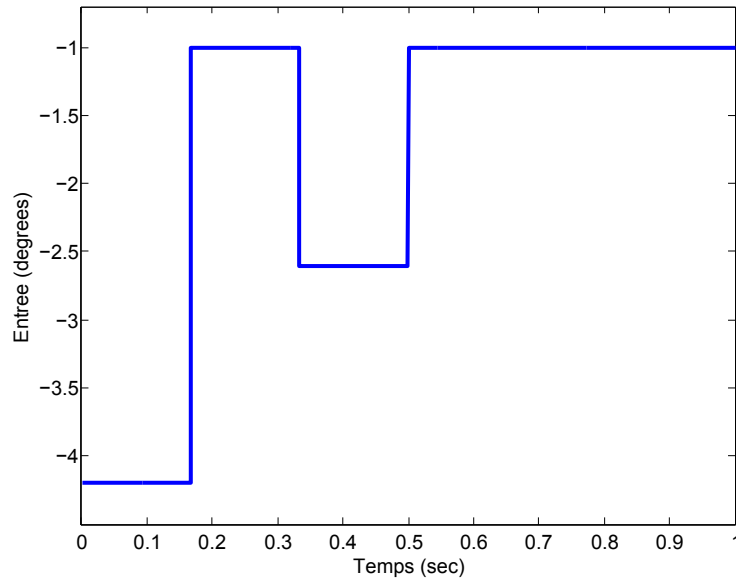


FIGURE 4.9 – Entrée optimale via le critère Migmag.

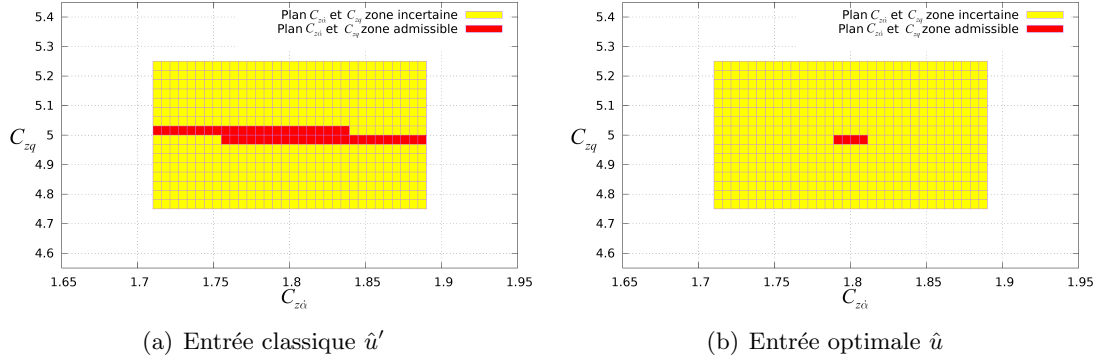
Le domaine obtenu via l'entrée optimisée est noté ci-dessous.

TABLE 4.5 – Résultat de l'estimation des paramètres en prenant ε_1 .

Paramètre	\hat{u}'	\hat{u}
$C_{z\dot{\alpha}}$	[1.71, 1.89]	[1.78875, 1.81125]
C_{zq}	[4.96875, 5.03125]	[4.96875, 5]

Dans le Tableau 4.5, l'entrée \hat{u}' est une entrée classique et l'entrée u est l'entrée obtenue précédemment.

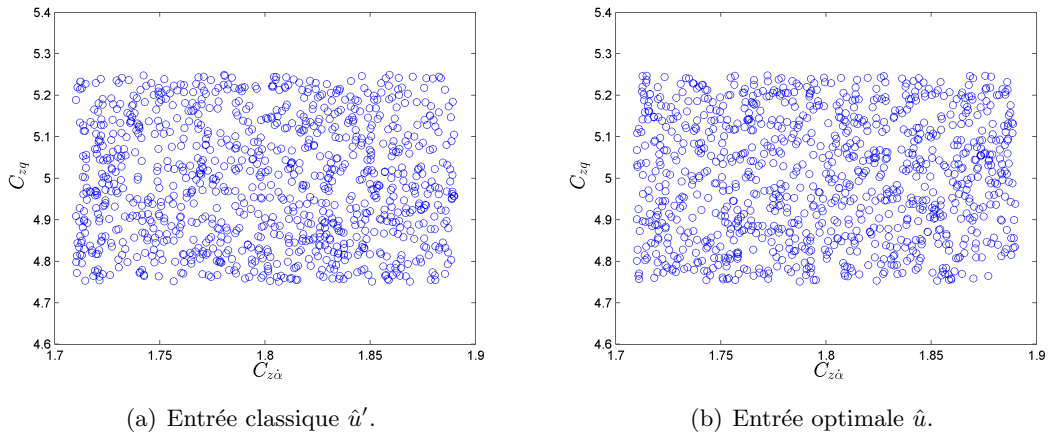
La Figure 4.11(a) est le résultat avec l'entrée non optimisée, le calcul dure 390s. La Figure 4.11(b) est le résultat de l'estimation des paramètres avec l'entrée optimale, le calcul dure 357s. Dans ces figures, les boîtes acceptables sont représentées par des blocs rouges, et les boîtes incertaines sont représentées par des blocs jaunes. Le contour est la boîte initiale $[p_0]$, la précision pour le découpage est $\varepsilon = (0.01, 0.05)$.

FIGURE 4.10 – Boîte acceptable et boîte incertaine avec l'entrée \hat{u}' et \hat{u} (ε_1).

Pour comparaison, nous avons aussi utilisé la méthode de Monte Carlo pour estimer les paramètres $C_{z\alpha}$ et C_{zq} . Nous essayons de trouver les ensembles de paramètres qui nous permettent de satisfaire la relation suivante :

$$\begin{aligned} y(t_i) - y^m(t_i, p) &< \bar{v}(t_i). \\ y(t_i) - y^m(t_i, p) &> \underline{v}(t_i). \end{aligned} \quad (4.77)$$

Après avoir fait 1000 fois de l'itération, nous avons obtenu la Figure 4.11 :

FIGURE 4.11 – Estimation de paramètres par Monte Carlo avec l'entrée \hat{u}' et \hat{u} .

La Figure 4.11 ne présente pas de meilleurs résultats par rapport à ceux obtenus par l'analyse par intervalles. La solution qui est la plus proche des mesures est cachée dans les paramètres passés par le test de Monte Carlo. Toutes les particules bleus sont considérées comme des points cohérents avec les mesures. Aucune information concernant la vraie solution sur le pavé initial de paramètres peut être obtenue. D'autre part, ces deux paramètres sont difficiles à estimer (aucune zone rejetée ou bien incohérence avec les mesures). L'analyse par intervalles peut fournir d'une façon sûre alors que le test de Monte Carlo a échoué pour distinguer les valeurs cohérentes avec les mesures. Les résultats donnés par l'analyse par intervalles font une preuve que, avec une entrée

Migmag-optimale-intervalle, le domaine admissible a été amélioré, le domaine admissible est encadré par le choix d'une entrée "optimale".

4.3 Conclusion

Dans ce chapitre, nous avons mis en évidence l'impact du choix d'une entrée sur l'estimation de paramètres. Nous avons proposé une procédure pour l'estimation de paramètres avec la planification d'une entrée optimale dans le contexte d'intervalle. Nous avons d'abord utilisé les critères de trace, M-optimalité et de D-optimalité. Nous avons testé sur l'exemple tiré du domaine aéronautique. Les critères de trace et M-optimalité donne des résultats identiques, ils permettent d'obtenir un meilleur encadrement des ensembles admissibles. Le critère D-optimalité encadre moins bien les ensembles admissibles, mais élimine plus d'ensemble à rejeter. Dans tous les cas, les entrées obtenues nous permettent d'obtenir un domaine de recherche plus contractant pour les paramètres. Une augmentation de précision sur les paramètres estimés et une réduction du temps de calcul sont aussi obtenues avec ces entrées optimales.

Ensuite, le critère Migmag mis en évidence a été testé sur un modèle complexe non linéaire d'avion et les résultats sont prometteurs. Le coefficient $C_{z\dot{\alpha}}$ est difficile à encadrer et nous avons travaillé à des stratégies mettant en œuvre des entrées optimales judicieusement choisies. Le critère Migmag a été capable de rendre le domaine admissible de ce paramètre plus contractant. Via ce critère, toutes les largeurs de paramètres sont diminuées. Les résultats par la méthode Monte Carlo valident que aucune boîte rejetable existe dans l'espace de paramètres a priori. Toutes les boîtes sont considérées comme admissibles.

Dans le prochain chapitre, nous allons utiliser l'entrée obtenue par la D-optimalité au cas du diagnostic de fautes basé sur le modèle du système.

Diagnostic par analyse par intervalles

Contents

5.1 Diagnostic dans le contexte à erreurs bornées	103
5.1.1 Approche par estimation d'état	105
5.1.2 Approche par estimation de paramètres	106
5.2 Diagnostic pour le modèle d'avion	107
5.2.1 Diagnostic par estimation d'état	107
5.2.2 Diagnostic par estimation de paramètres	108
5.3 Diagnostic actif pour le modèle d'avion	109
5.3.1 Cas de faute simple	110

5.1 Diagnostic dans le contexte à erreurs bornées

Il est souvent demandé aux systèmes actuels de développer des capacités d'autonomie qui leur permettent de résister à l'apparition d'une faute ou toutes autres perturbations, tout en restant fonctionnels avec un service éventuellement dégradé mais sûr. Ils doivent donc intégrer des mécanismes leur permettant de percevoir un mode opératoire dégradé, sans intervention humaine, puis de réaliser les corrections nécessaires à la restauration d'un mode opératoire normal, ou d'un mode dégradé mais sûr. Ces opérations sont basées notamment sur une bonne modélisation du système étudié.

Pour cela, la mise en place d'une méthode de diagnostic est importante, elle doit permettre de détecter, localiser et identifier les fautes affectant le système. Le diagnostic consiste tout d'abord à surveiller les entrées et sorties du système afin de décider si une faute est présente. Si une faute est détectée, il s'agit de la localiser et d'en identifier les caractéristiques [Isermann 1997]. De nombreux travaux existent pour le diagnostic actif : citons par exemple [Zhang 1989], [Nikoukhah 1998], [Nikoukhah 2000], [Nikoukhah 2006], [Gholami 2011] ou [Tabatabaeipour 2009].

Dans ce travail, nous considérerons qu'une *faute* est détectée lorsqu'un paramètre de l'entité à diagnostiquer n'a pas sa valeur nominale et que cette différence est suffisante pour provoquer un dysfonctionnement dans des circonstances données [Isermann 2005].

Les fautes apparaissant sur un système peuvent être classées en trois catégories : les fautes actionneurs, les fautes capteurs et les fautes des composants. Concernant la détection, différentes approches existent :

- redondance matérielle : cette technique est assez intuitive, elle propose d'utiliser plusieurs capteurs en rajoutant un système de décision. La décision est prise selon les états des capteurs. Mais, la complexité et le coût matériel sont souvent élevés,
- redondance analytique : cette technique requiert la connaissance d'un modèle mathématique. Elle collecte les mesures à partir du système réel et les valeurs calculées par le système du modèle. Ensuite, elle étudie la différence de ces deux variables et en tire des conclusions.

Dans ce travail, nous nous concentrons sur le diagnostic à base de modèles. Nous considérons cette approche en contexte ensembliste.

Dans la littérature, l'idée de redondance analytique est premièrement proposée par [Willsky 1976]. Ensuite, [Frank 1990], [Frank 1997], [Patton 1997], [Isermann 1992], [Isermann 2005] présentent leurs travaux en utilisant les techniques de détection de faute. Dans le livre de [Chen 2012], la redondance analytique a été détaillée et implémentée à la détection de faute et au diagnostic sur les systèmes dynamiques. Les approches par estimation de paramètres, à base de modèle d'état et de relation de parité ont été développées dans ce livre.

Les méthodes analytiques se basent sur la connaissance du système à surveiller et ce par le biais d'un modèle de comportement. Ces techniques comparent les informations issues du système réel en fonctionnement avec celles calculées à partir du modèle. Cette opération aboutit à la génération de variables appelées résidus. La génération de résidus est une étape d'extraction et de concentration d'information : les résidus sont censés contenir toute l'information utile pour le diagnostic. Cette étape est fondamentale car elle conditionne la qualité de la détection. Cette méthode basée sur la génération de résidus sera celle développée par la suite car elle répond à nos objectifs : détection, localisation et identification.

L'idée de base de la redondance analytique est le calcul de résidus comme indiqué dans la Figure 5.1. Le résidu r peut être défini par la différence entre les mesures à partir du système réel et les sorties estimées par le modèle mathématique du système :

$$r(t) = y(t) - y^m(t). \quad (5.1)$$

Dans le contexte à erreurs bornées, l'équation (5.1) peut être considérée comme l'erreur d'incertitude. Une erreur au delà d'un seuil défini ultérieurement implique la présence d'une faute potentielle. Souvent, le système réel subit de nombreux types de fautes, *l'isolation de faute* est la capacité du processus de diagnostic à les distinguer.

Nous supposons que les sorties du système sont disponibles à chaque instant t . Une sortie du système est notée d'une manière générale Y et $Y = (y(t_1), \dots, y(t_N))^T$. Nous noterons $T = (t_{n_1}, \dots, t_{n_k})$ un ensemble d'instant de mesures, $Y_i(T) = (y_i(t_{n_1}), \dots, y_i(t_{n_k}))^T$ la sortie qui subit la faute f_i .

Pour que la faute f_i soit détectable, il suffit qu'il existe un instant $t_{n_{ji}}$ tel que la différence entre $y_i(t_{n_{ji}})$ et le mode normal en $t_{n_{ji}}$ noté $y_0(t_{n_{ji}})$ soit significative [Isermann 2005] :

$$|y_0(t_{n_{ji}}) - y_i(t_{n_{ji}})| \geq d \text{ pour tout } i \in \{1, \dots, n_f\}, \quad (5.2)$$

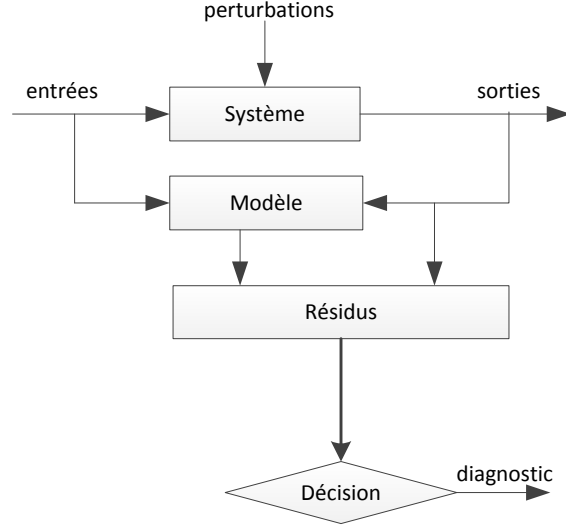


FIGURE 5.1 – Approche du diagnostic par redondance analytique

où $d > 0$ et n_f représente le nombre de fautes. $t_{n_{ji}}$ est un élément de $(y_i(t_{n_{1i}}), \dots, y_i(t_{n_{ki}}))$.

Dans le contexte à erreurs bornées en utilisant l'analyse par intervalles, il existe de nombreuses méthodes de diagnostic dans la littérature, par exemple [Adrot 2002], [Raïssi 2004b], [Bedoui 2014] ou encore [Drevelle 2015]. Ces approches fournissent des résultats garantis. La détection de fautes, dans un contexte à erreurs bornées, est effectuée par un test d'appartenance. Nous détectons une faute si l'estimation à partir de modèle à l'instant t_k est inconsistante avec les mesures à l'instant t_k :

$$[y^m](t_k) \not\subseteq [\underline{y}(t_k), \overline{y}(t_k)]. \quad (5.3)$$

Ceci peut être réécrit sous la forme suivante :

$$\begin{aligned} & 0 \notin [\underline{y}(t_k), \overline{y}(t_k)] - [y^m](t_k), \\ \Rightarrow & 0 \notin [\underline{y}(t_k) - y^m(t_k), \overline{y}(t_k) - y^m(t_k)], \\ \Rightarrow & 0 \notin [\underline{r}(t_k), \overline{r}(t_k)], \end{aligned} \quad (5.4)$$

où $[r(t_k)]$ est le résidu à l'instant t_k .

Dans les sections qui suivent, nous présentons les approches par estimation d'état et par estimation de paramètres pour le diagnostic à base de modèles dans un contexte à erreurs bornées.

5.1.1 Approche par estimation d'état

La détection de fautes par estimation d'état consiste à vérifier les incohérences entre les sorties prédites du modèle et les mesures. Toutes les incertitudes sont considérées sous

forme d'intervalles [Stancu 2005].

[Marx 2010] a proposé d'utiliser un mode actif afin de trouver un seuil adaptatif pour ne pas perdre la sensibilité à la faute. [Frank 1997] introduit l'estimation par intervalles en la combinant avec un seuil en logique floue. Ces seuils sont le moins serrés possible afin de ne pas perdre la qualité de la détection de faute. Des applications peuvent être trouvées dans [Alhaj-Dibo 2004] [Karim 2008] [Blesa 2014].

La détection de faute peut être décrite de la façon suivante :

Algorithme 13 Diagnostic via estimation d'état ($[x]$)

Input: $[x](0)$, $[p](0)$;

Output: $[x]$;

```

1: initialization :  $[x]_e(0) := ([x](0), [p](0))$  ;
2:  $i := 1$  ;
3: fault := false ;
4: while  $i \leq N$  do
5:    $[x]_e^+(i) := ODE([x]_e(i-1))$  ;
6:   if  $w([g][x](i)) > w([e(i)])$  and fault = false then
7:      $[x]_e(i) := [x]_e^+(i) \cap [g]^{-1}([y_j])$ 
8:   if  $[x]_e^+(i) \cap [g]^{-1}([y_j]) = \emptyset$  then
9:     fault := true ;
10:   $i := i + 1$  ;
```

Cet algorithme sera testé dans les sections suivantes.

5.1.2 Approche par estimation de paramètres

Quand le modèle est linéaire, il est possible d'utiliser par exemple l'approche ellipsoïdale [Milanese 1991] [Reppa 2011] ou encore l'approche par zonotopes [Ingimundarson 2008]. Nous présentons dans cette section une approche basée sur l'estimation de paramètres en utilisant l'analyse par intervalles.

Lors de la détection de dysfonctionnement par estimation de paramètres, les fautes possibles du système à surveiller sont supposées pouvoir être associées aux paramètres du modèle et aux états spécifiques du modèle mathématique du système. Ainsi, partant de l'estimation des paramètres du modèle à partir des mesures des sorties y et des entrées u , une faute est détectée s'il existe des variations sur les paramètres à partir des seuils de tolérance préalablement définis.

Le vecteur des résidus $[r]$ est alors obtenu en faisant la différence entre les paramètres estimés $[\hat{p}]$ et les valeurs nominales $[p]$:

$$[r] = [p] - [\hat{p}], \quad (5.5)$$

Une faute est détectée si 0 n'appartient pas à au moins une composante de $[p] - [\hat{p}]$. Un vecteur de pondération w peut être aussi considéré afin d'évaluer les différentes réactions sur la déviation de paramètres avec sa valeur nominale.

$$[r_w] = w([p] - [\hat{p}]). \quad (5.6)$$

Une approche similaire peut être trouvée dans [Li 2015a].

Dans la section suivante, nous appliquons les deux approches proposées de détection/diagnostic au modèle d'avion et comparons les résultats obtenus.

5.2 Diagnostic pour le modèle d'avion

Nous rappelons que le système (3.71) représente le comportement en vol longitudinal d'un avion en atmosphère au repos.

5.2.1 Diagnostic par estimation d'état

Nous rappelons que pour estimer les états, il y a deux étapes : prédiction et correction. L'étape de correction s'effectue lorsque les résultats d'estimation dépassent certains critères prédéfinis.

Les résultats proposés sont effectués en simulation. Les mesures sont générées via le modèle d'avion en prenant $p = (1.8, 5, -5, -15)$ à partir de $t = 0$, notons que le dernier paramètre $C_{mq} = -22$ en situation nominale. Nous avons ainsi injecté une faute dans le système. Les états initiaux sont supposés satisfaire :

$$[X_0] = [28.48, 28.52] \times [0.1094, 0.1174] \times [-0.004, 0.004] \times [0.0384, 0.0464]. \quad (5.7)$$

Nous supposons que l'état x est entièrement mesuré et les bruits de mesures sont supposés additifs :

$$[E] = [-0.0447, 0.0447] \times [-0.0044, 0.0044] \times [-0.0044, 0.0044] \times [-0.0044, 0.0044]. \quad (5.8)$$

Les états sont générés via l'outil de calcul garanti VNODE-LP. Le domaine initial de paramètres $[P_0]$ est donné par :

$$[P_0] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-23.1, -20.9]. \quad (5.9)$$

Dans les figures ci-dessous, les courbes de couleur rouge représentent les mesures et les courbes de couleur bleu représentent les sorties du modèle.

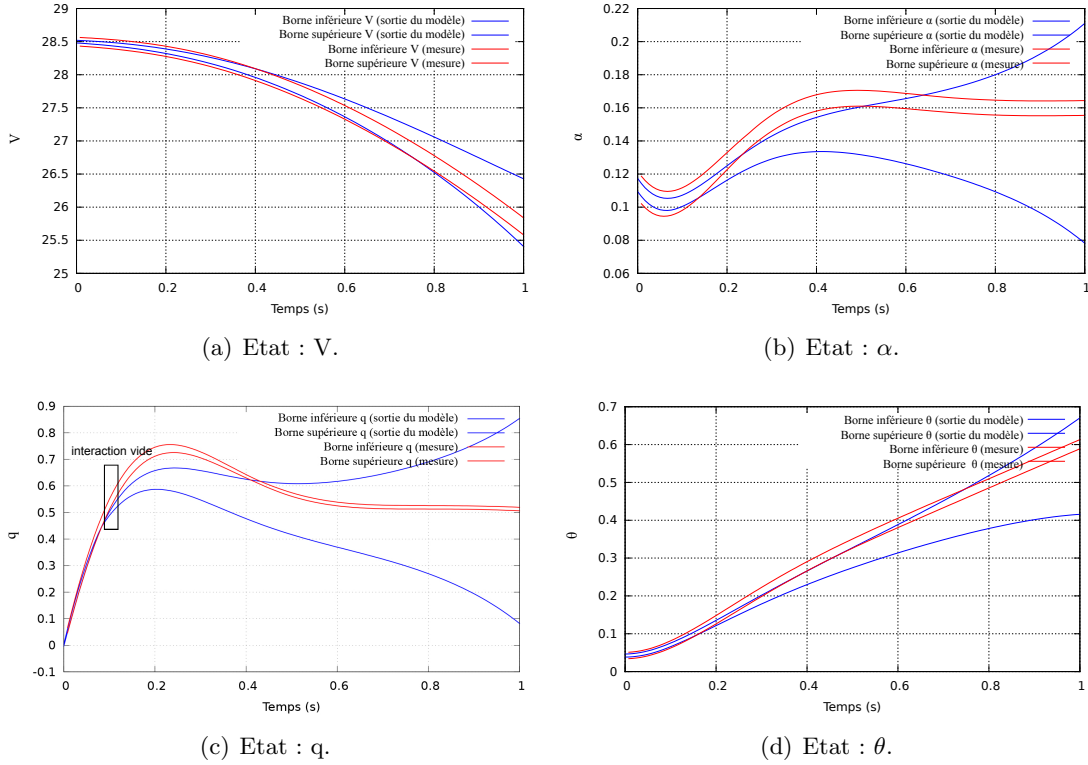


FIGURE 5.2 – L'estimation d'état du modèle d'avion

Nous voyons que la faute est détectée à $t = 0.091s$. Mais, le pessimisme apparaît dès que nous n'avons plus de mesures à corriger ainsi quelques instants plus tard la solution du modèle couvre de nouveau les mesures.

Nous présentons donc dans la section suivante, le diagnostic sur le modèle d'avion par estimation de paramètres.

5.2.2 Diagnostic par estimation de paramètres

La méthodologie d'estimation de paramètres présentée dans les précédents chapitres peut être utilisée pour le diagnostic de faute.

Comme dans la section sur le détection de faute du modèle d'avion par estimation d'état, nous travaillerons en simulation. Les états initiaux $[X_0]$ et les erreurs de mesures E sont les même que précédemment. Nous ne considérons, pour commencer, qu'une faute sur le système. Ainsi, les paramètres en défauts sont donnés par $p = (1.8, 5, -5, -15)$. Le domaine initial des paramètres $[P_0]$ n'est le même que précédemment, il est ici donné par :

$$[P_0] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-23.1, -11.5]. \quad (5.10)$$

Nous mettons un large intervalle pour le domaine de C_{mq} . Ceci augmente considérablement le temps de calcul. Afin de réduire le temps de calcul, il serait intéressant d'utiliser un nombre minimal de points afin de réduire le nombre de bisection lors de l'utilisation de SIVIA.

La précision de découpage lors de la phase de bisection est $\varepsilon = (0.05, 0.25, 0.25, 0.5)$. L'entrée est considérée constante $\delta = -4.2$ deg pendant la période de simulation de 1 s. Le solveur utilisé est VSPODE, le contracteur a été appliqué en choisissant $C_i = 2$ (voir l'Algorithme 8). Le temps total d'exécution de l'algorithme d'estimation de paramètres est de 328 secondes. Aucune boîte admissible n'a été trouvée. L'enveloppe de la zone incertaine obtenue est :

$$[p_{hull}] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-15.9084, -14.1344]. \quad (5.11)$$

Nous pouvons dire que la faute a été bien détectée après 328 secondes d'exécution. Cette faute se situe sur le paramètre C_{mq} .

Dans ce test, l'utilisation du contracteur avec critère volumique a permis de réduire le volume de boîtes rejetables, ce qui accélère le processus d'estimation de paramètres. Comme aucune boîte admissible n'a été trouvée, afin d'avoir plus de précision sur l'ensemble admissible, nous pouvons choisir une valeur plus petite comme précision de découpage. Dans un autre test, nous avons injecté deux fautes sur les paramètres $C_{zq} = 4.8$ et $C_{mq} = -15$, dans ce cas, l'estimation de paramètres n'a pas permis de mettre en évidence les paramètres en situation de fautes. En prenant $\varepsilon = (0.1, 0.5, 0.5, 1)$, le domaine incertain suivant :

$$[p_{hull}] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-15.85, -14.4]. \quad (5.12)$$

a été obtenu en 7 secondes. Comme nous avons utilisé une entrée constante, l'utilisation d'une entrée optimale devrait permettre de raffiner l'estimation de paramètres et donc ici le diagnostic.

Un dernier test est effectué en utilisant le domaine initial suivant pour les paramètres :

$$[P_0] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-23.1, -20.9]. \quad (5.13)$$

Remarquons qu'ici, nous mettons un intervalle "sans défaut" pour le domaine de C_{mq} . Nous utilisons les mêmes configurations (états initiaux, précision de découpage, etc.) que précédemment. Le temps de l'estimation de paramètres est de 16 secondes. Tous les paramètres initiaux sont considérés comme des boîtes rejetables. Une faute a été détectée à 16 secondes. Mais nous sommes incapables de dire où se trouve cette faute.

5.3 Diagnostic actif pour le modèle d'avion

Nous voyons que les résultats obtenus pour les entrées constantes ne sont pas satisfaisants en ce qui concerne la détection de faute. Quand les fautes ne sont pas de grandes amplitudes, l'encadrement de paramètres admissibles via l'entrée constante ne nous permet pas de les détecter. Donc une optimisation sur le choix de l'entrée nous semble pertinent.

Pour le diagnostic actif via l'estimation d'état, des algorithmes similaires peuvent être trouvés dans [Frank 1987] [Maillet 2013].

Nous n'évoquerons dans cette section que l'approche par estimation de paramètres. Pour rechercher l'entrée optimale, nous utilisons le critère de trace optimalité ensembliste sur 5 horizons. L'entrée retenue est $u(t) = (-4.2, -4.2, -1, -4.2, -4.2)$.

5.3.1 Cas de faute simple

Les états initiaux sont supposés donnés par :

$$[X_0] = [28.48, 28.52] \times [0.1094, 0.1174] \times [-0.004, 0.004] \times [0.0384, 0.0464]. \quad (5.14)$$

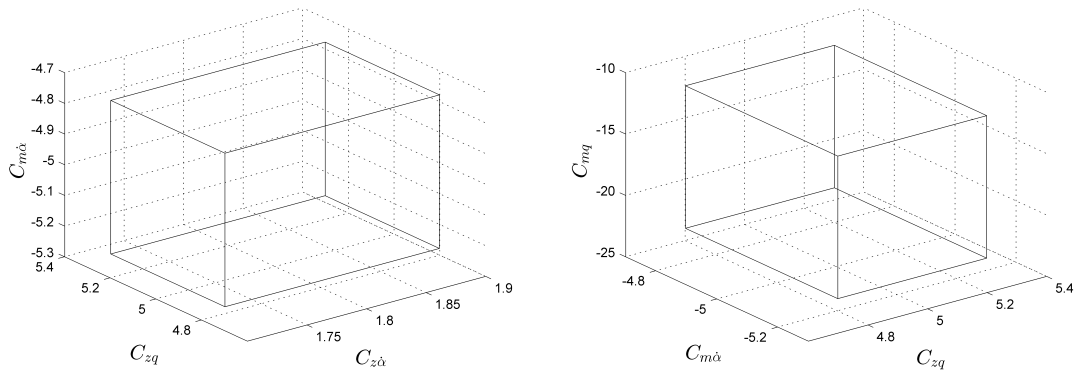
Nous considérons une faute sur le dernier paramètre $C_{mq} = -15$. Les états sont générés via l'outil de calcul garantie VSPODE. Les erreurs sont supposées additives sur chaque état :

$$[E] = [-0.0447, 0.0447] \times [-0.0044, 0.0044] \times [-0.0044, 0.0044] \times [-0.0044, 0.0044]. \quad (5.15)$$

Le domaine initial est donné par :

$$[P_0] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-23.1, -11.5]. \quad (5.16)$$

Les Figures 5.3, 5.4, 5.5 suivantes représentent les résultats d'estimation de paramètres.



(a) Domaines admissibles de paramètre $C_{z\dot{\alpha}}$, C_{zq} et $C_{m\dot{\alpha}}$.
(b) Domaines admissibles de paramètre C_{zq} , $C_{m\dot{\alpha}}$ et C_{mq} .

FIGURE 5.3 – La représentation des domaines admissibles en 3D

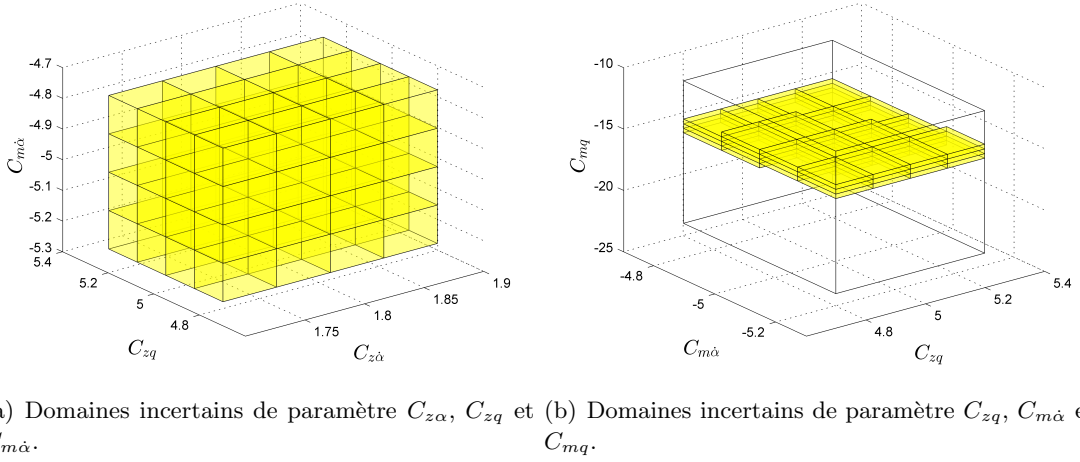


FIGURE 5.4 – La représentation des domaines incertains en 3D.

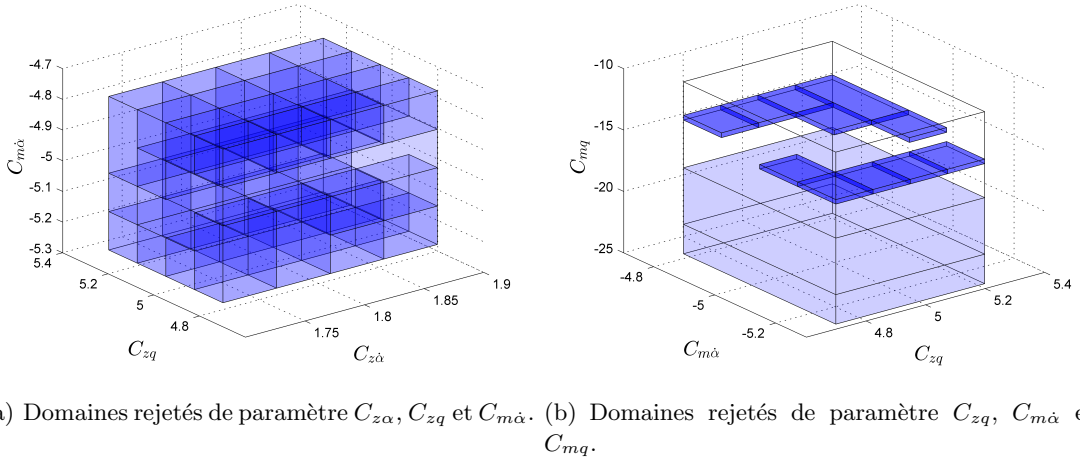


FIGURE 5.5 – La représentation des domaines rejetés en 3D.

Le temps d'exécution est de 318 secondes, ce qui est plus petit que celui avec l'entrée constante. Nous encadrons les ensembles incertains par l'enveloppe $[p_{hull}]$:

$$[p_{hull}] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-15.7306, -14.2281]. \quad (5.17)$$

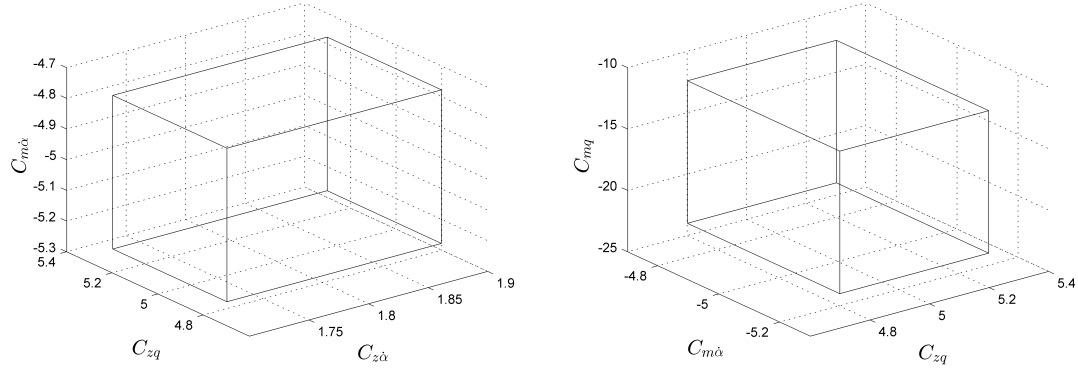
La faute sur C_{mq} a bien été détectée. Les autres paramètres sont également bien estimés.

Nous gardons les configurations de l'expérience précédente. Nous supposons cependant que deux paramètres sont en défauts ; p est alors donné par $(1.8, 5, -4.8, -15)$.

Après avoir fait tourner notre processus d'estimation, nous obtenons :

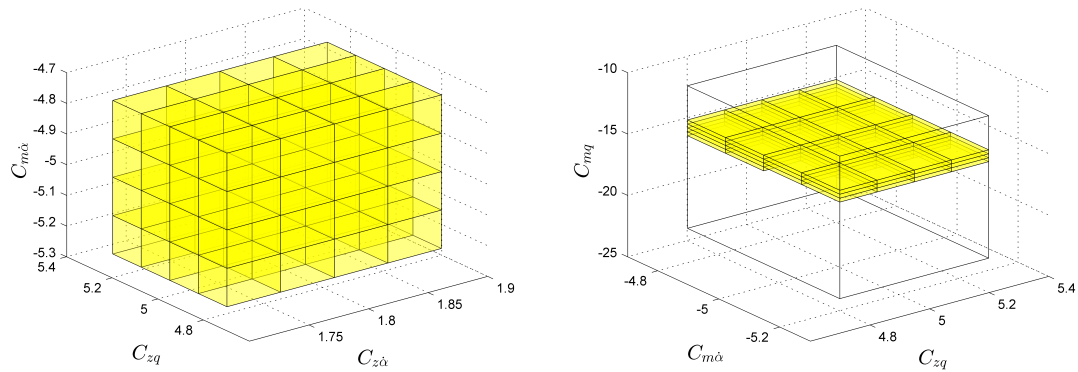
$$[p_{hull}] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-15.5625, -14.2811]. \quad (5.18)$$

La faute sur C_{mq} est bien détectée par contre, celle sur $C_{m\dot{\alpha}}$ ne l'est pas car l'intervalle estimé pour $C_{m\dot{\alpha}}$ contient la valeur nominale et valeur de la faute



(a) Domaines admissibles de paramètre $C_{z\dot{\alpha}}$, C_{zq} et $C_{m\dot{\alpha}}$.
(b) Domaines admissibles de paramètre C_{zq} , $C_{m\dot{\alpha}}$ et C_{mq} .

FIGURE 5.6 – La représentation des domaines admissibles en 3D.



(a) Domaines incertains de paramètre $C_{z\dot{\alpha}}$, C_{zq} et $C_{m\dot{\alpha}}$.
(b) Domaines incertains de paramètre C_{zq} , $C_{m\dot{\alpha}}$ et C_{mq} .

FIGURE 5.7 – La représentation des domaines incertains en 3D.

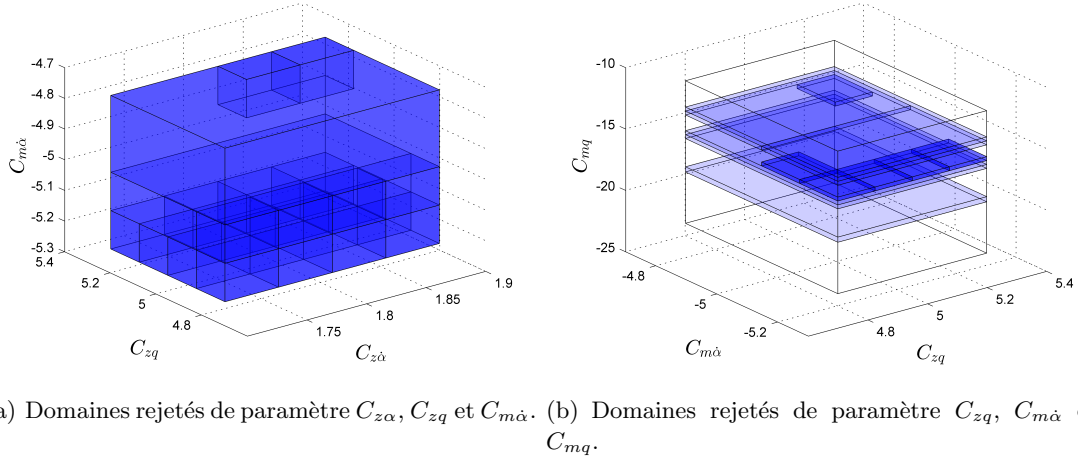


FIGURE 5.8 – La représentation des domaines rejetés en 3D.

Afin d'améliorer l'estimation du paramètre $C_{m\alpha}$, nous choisissons une précision plus fine. Cette fois, nous choisissons $\varepsilon = (0.01, 0.05, 0.05, 0.1)$ et $C_i = 2$ pour $i = 1, 2, 3, 4$. Notons que $[P_0] = [1.71, 1.89] \times [4.75, 5.25] \times [-5.25, -4.75] \times [-23.1, -11.5]$.

Après avoir fait tourner notre processus d'estimation, nous obtenons :

$$[p_{hull}] = [1.71, 1.89] \times [4.9063, 5.0625] \times [-4.8438, -4.75] \times [-15.27, -14.5371]. \quad (5.19)$$

avec une entrée optimale. Les fautes sont détectés à 421s.

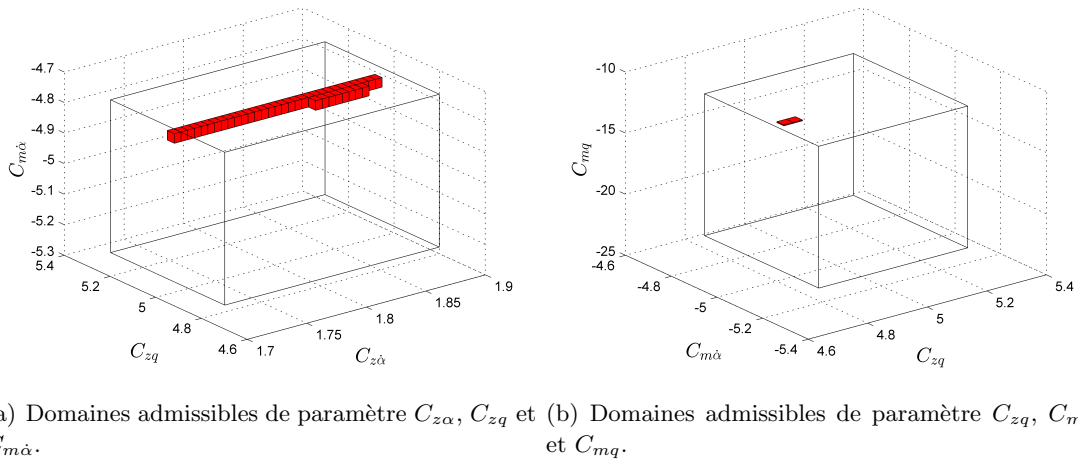


FIGURE 5.9 – La représentation des domaines admissibles en 3D.

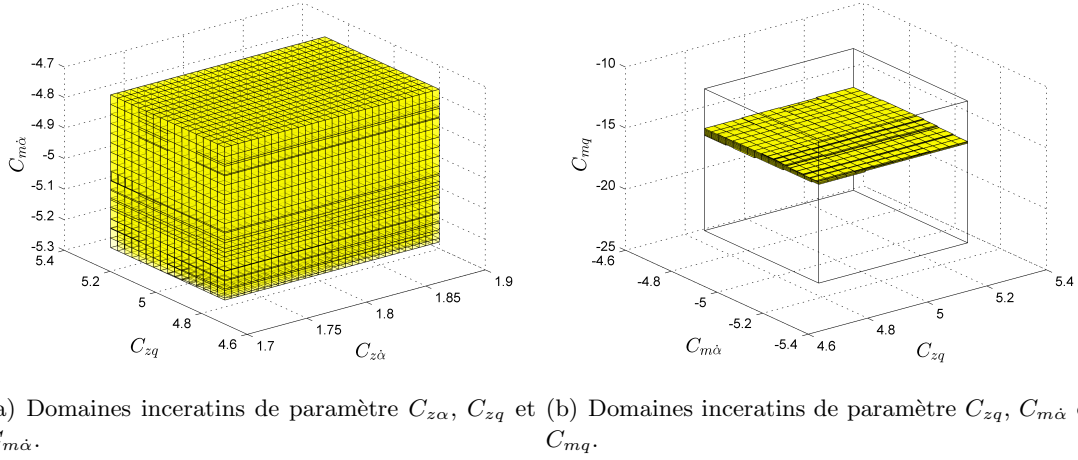


FIGURE 5.10 – La représentation des domaines incertains en 3D.

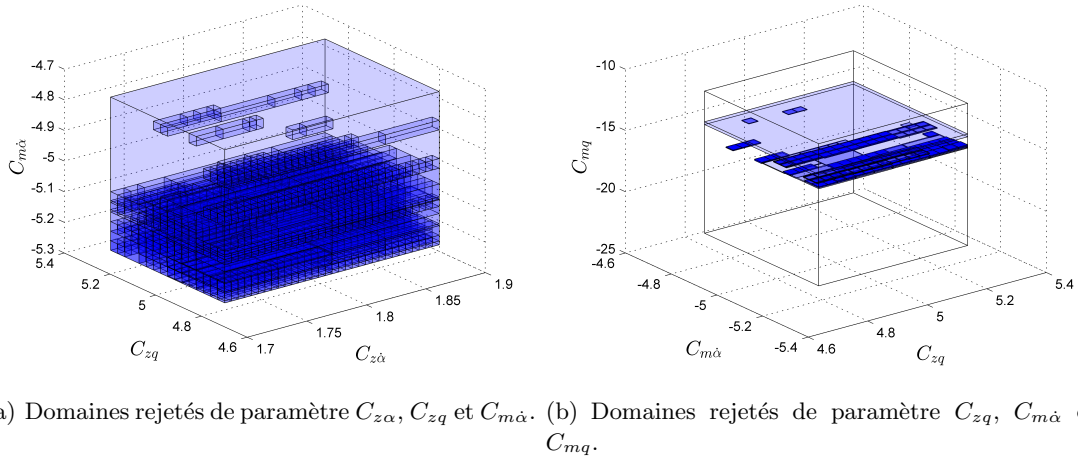
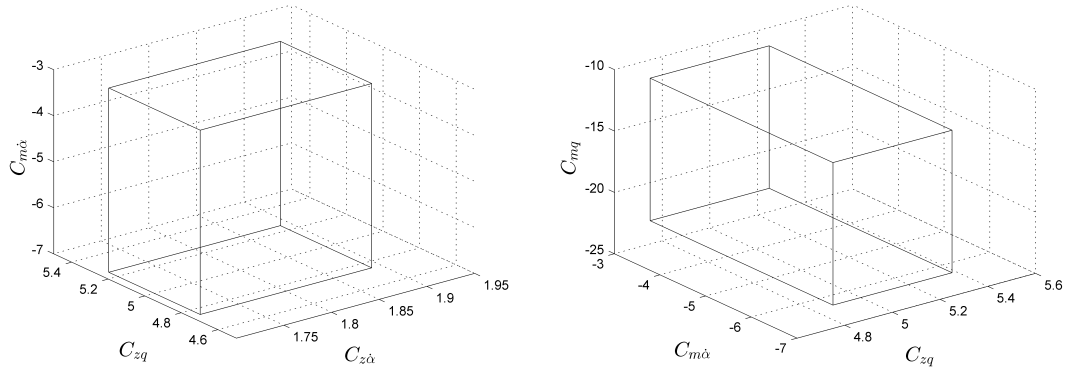


FIGURE 5.11 – La représentation des domaines rejetés en 3D.

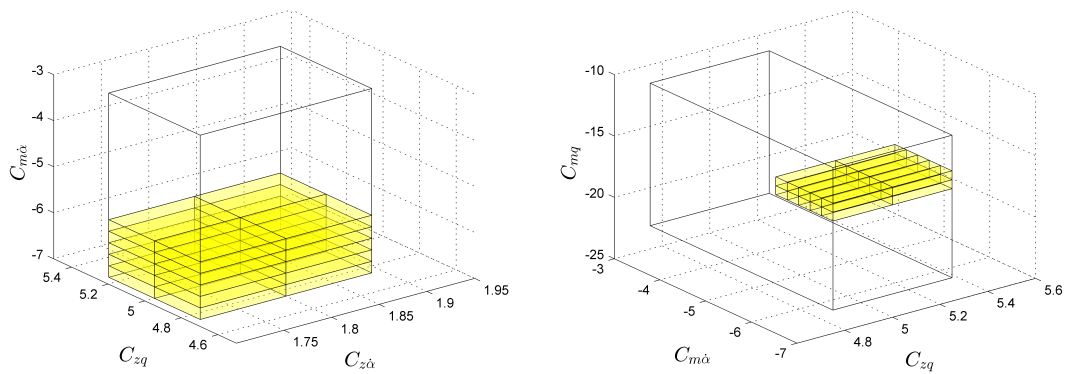
Pour réduire le temps de détection, nous élargissons l'intervalle de recherche pour $[P_0]$, la valeur d'épsilon a été augmentée à $\varepsilon = (0.1, 0.5, 0.5, 1)$, la valeur de la faute sur $C_{m\alpha}$ est considérée plus grande et valant -6.5 . Notons que $[P_0] = [1.71, 1.89] \times [4.75, 5.25] \times [-7, -3] \times [-23.1, -11.5]$.

Nous faisons cette estimation en utilisant l'entrée optimale, les résultats suivants sont obtenus dans 2 secondes.



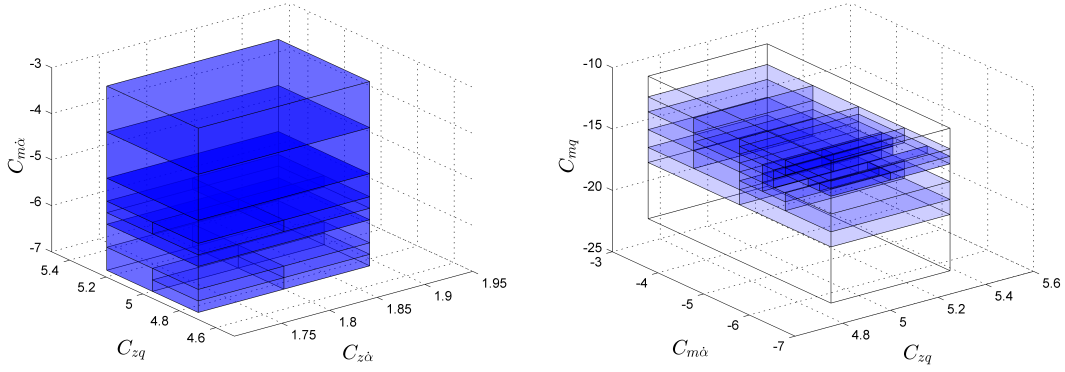
(a) Domaines admissibles de paramètre $C_{z\alpha}$, C_{zq} et (b) Domaines admissibles de paramètre C_{zq} , $C_{m\alpha}$ et C_{mq} .

FIGURE 5.12 – La représentation des domaines admissibles en 3D.



(a) Domaines incertains de paramètre $C_{z\alpha}$, C_{zq} et (b) Domaines incertains de paramètre C_{zq} , $C_{m\alpha}$ et C_{mq} .

FIGURE 5.13 – La représentation des domaines incertains en 3D.



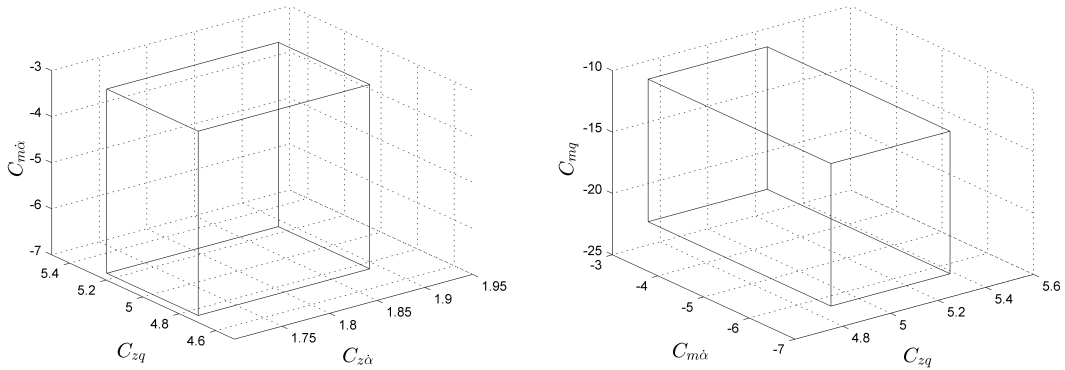
(a) Domaines rejetés de paramètre $C_{z\alpha}$, C_{zq} et $C_{m\dot{\alpha}}$. (b) Domaines rejetés de paramètre C_{zq} , $C_{m\dot{\alpha}}$ et C_{mq} .

FIGURE 5.14 – La représentation des domaines rejetés en 3D.

Avec l'entrée optimale, nous avons obtenu les paramètres :

$$[p_{hull}] = [1.71, 1.89] \times [4.75, 5.25] \times [-7, -5.75] \times [-15.85, -14.40]. \quad (5.20)$$

Nous voyons que cette fois, les fautes sont détectées et localisées. Si nous faisons cette expérience avec l'entrée constante :



(a) Domaines admissibles de paramètre $C_{z\alpha}$, C_{zq} et $C_{m\dot{\alpha}}$. (b) Domaines admissibles de paramètre C_{zq} , $C_{m\dot{\alpha}}$ et C_{mq} .

FIGURE 5.15 – La représentation des domaines admissibles en 3D.

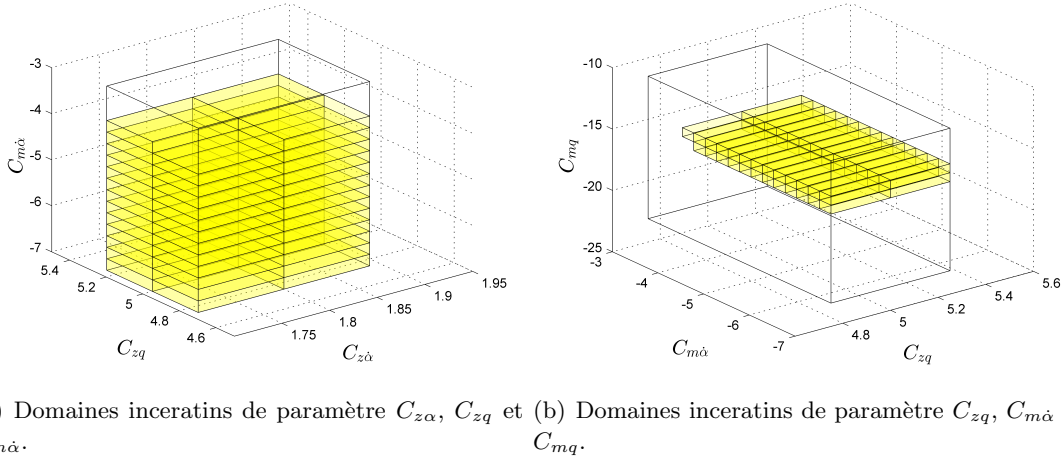


FIGURE 5.16 – La représentation des domaines incertains en 3D.

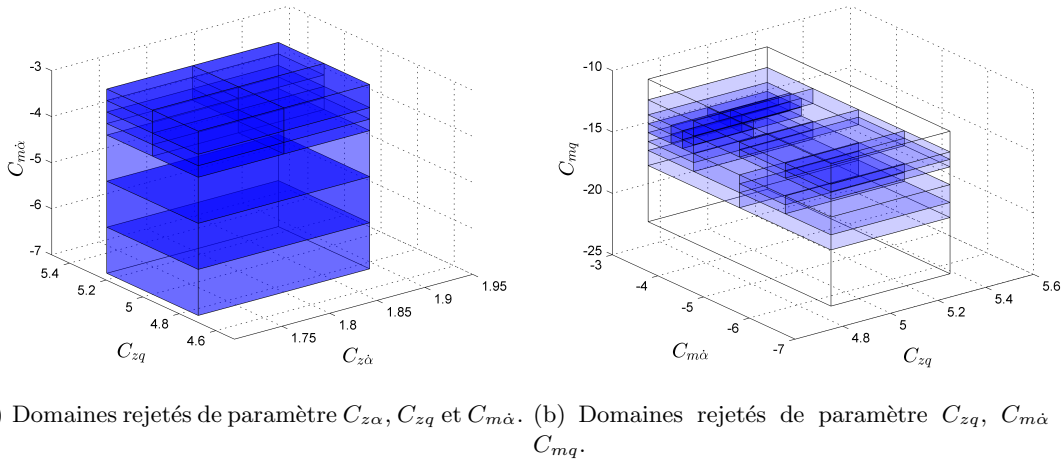


FIGURE 5.17 – La représentation des domaines rejetés en 3D.

Les résultats sont obtenus dans 5 secondes.

$$[p_{hull}] = [1.71, 1.89] \times [4.75, 5.25] \times [-7, -4.75] \times [-15.85, -14.40]. \quad (5.21)$$

La faute sur C_{mq} a été détectée mais aucune décision peut être faite au $C_{m\alpha}$.

L'avantage de l'approche basée sur l'analyse par intervalles est d'estimer d'une façon sûre les paramètres sans introduire d'hypothèse stochastique, les solutions du problème d'estimation sont garanties. Mais le conservatisme nous empêche de diagnostiquer les fautes de petites amplitudes. Une méthode possible est d'utiliser le diagnostic actif en utilisant une entrée plus informative. Les résultats trouvés sont alors plus sensibles aux fautes. Nous avons alors obtenu plus d'information sur les paramètres au travers d'une seule expérience. Cependant, lorsque les fautes sont multiples, il y a un effet de dégradation sur les résultats du diagnostic. Une solution possible est d'exécuter plusieurs

expériences en se focalisant sur un seul paramètre. Pour effectuer cela, le critère comme Migmag-ensembliste pourrait être utile.

Dans ce chapitre, nous avons utilisé deux méthodes de diagnostic à base de modèle : à base d'estimation d'état et à base d'estimation de paramètres.

Les résultats obtenus sur l'exemple d'avion, par estimation d'état, ne nous permettent pas sans autre considération de localiser les fautes. Ensuite, nous avons utilisé le critère trace-optimalité-ensembliste présenté dans le chapitre 4 pour l'estimation de paramètres, d'autres critères auraient pu être utilisés. Les résultats obtenus en utilisant ce critère sont très encourageants.

Conclusion

Contents

6.1 Conclusion	119
6.2 Perspective	120

6.1 Conclusion

L'analyse par intervalle est une méthode polyvalente qui nous permet d'évaluer une fonction avec des résultats garantis, de résoudre des problèmes inverses de manière sûre. La transformation naturelle et simple sur les erreurs à partir de mesure donne des applications diverses qui peuvent être trouvées dans le domaine de la robotique, de l'aéronautique, de l'automobile, etc. De plus dans le domaine d'ingénierie, la simplicité de manipulation et la facilité de mise en œuvre du calcul d'intervalle ont fait la popularité dans le domaine physique, mathématique, etc.

Dans cette thèse, notre contribution principale est de fournir une méthodologie complète concernant l'estimation d'état et l'estimation de paramètres des systèmes dynamiques non linéaires avec la planification d'entrée optimale. Différents critères d'optimisation ont été proposés, dédiés à la minimisation du domaine de recherche sur les paramètres admissibles, l'efficacité d'utilisation de ces entrées planifiées a été démontrée avec une application tirée du domaine de l'aéronautique.

Nous avons aussi amené notre étude dans le domaine du diagnostic ainsi que le diagnostic actif, qui est plus performant pour détecter les fautes non facilement détectables. Ensuite, nous donnons des applications de ces méthodologies en considérant par exemple le modèle d'avion. Pour la détection de faute, nous pouvons détecter la faute à l'instant d'apparition, mais sans précision sur sa localisation. Si nous voulons des informations précises sur la faute, sa localisation, le type de faute, etc, alors nous devrions prendre plus de temps afin de sortir ces informations. Malgré tout, le résultat de l'estimation est donné de manière sûre sans les étapes de validation en comparant avec d'autres méthodes stochastiques. Le principal inconvénient de notre méthode porte sur le nombre de variables à estimer qui doit être petit. En effet, la complexité de calcul s'élève exponentiellement avec le nombre de paramètres en cas d'estimation de paramètre et la surestimation devient plus importante en cas d'estimation d'état. Néanmoins, une façon pour contourner ce problème peut être l'utilisation d'un contracteur.

6.2 Perspective

L'estimation de paramètres pour les systèmes non linéaires reste un sujet difficile quelque soit son contexte : l'analyse par intervalles ou par analyse stochastique. La difficulté se trouve sur la complexité du temps de calcul et aussi la technique pour générer les solutions garanties à partir de l'équation différentielle. Le temps d'intégration garanti reste petit, car le pessimisme augmente sévèrement au cours de calcul, ce qui rend les solutions d'intégration inutiles. L'analyse par intervalles peut être utile quand les comportements du système peuvent être calculés sans trop de pessimisme (difficile à distinguer les uns des autres) ou bien certains systèmes apportent quelques caractéristiques de monotonie. D'autre part, l'analyse par intervalle peut être aussi utile quand nous avons des redondances sur les informations reçues, dans ce cas, cet outil sera un outil performant pour donner une décision (solution) de manière garantie sans faire intervenir de supposition statistique.

Une piste de recherche intéressante est la fusion de donnée ou bien la propagation de contraintes en combinant les données sous formes intervalles et probabiliste. Sachant que tous les résultats obtenus par les intervalles restent sous une forme intervalle, des méthodologies sur la décision d'intervalle peuvent être aussi intéressantes. La technique de combinaison entre l'intervalle et la lois de distribution est à élaborer, car parfois les résultats d'intervalle restent trop pessimiste, un raffinement sur ces résultats utilisant des statistiques serait utile.

Aussi, nous pouvons utiliser les optimalités proposées sur les instants, essayons de sortir les instants le plus sensibilisants afin de réduire finalement le temps de calcul durant l'estimation de paramètres.

Annexe

A.1 Annexe pour l'algorithme de la recherche de l'entrée

Nous présentons ici une version simplifiée de l'algorithme d'optimisation d'essais développé qui utilise les fonctions suivantes :

eval-critère évalue le critère à optimiser.

eval-ODE évalue les sorties du système par résolution de (3.3).

eval-sensibilité calcule les sensibilités des sorties par rapport aux paramètres par résolution de l'équation (4.5).

Algorithme 14 Search for the optimal input (u_{opt})

Input: $[f], [x_0], [p_0], r, t;$

Output: $u(t);$

```

1: initialisation,  $[x]_e(0) = ([x_0], [p_0]), \mathbb{L} := \mathbb{L} \cup [x]_e$ 
2: for  $i := 1 : r$  do
3:   for  $j := 1 : t_r$  do
4:      $[x]_e(j) := pop(\mathbb{L})$ 
5:     for  $k := 1 : l+1$  do
6:        $u(k) := \text{input-signal}(k);$ 
7:        $[x]_e(j+1) := \text{eval-ODE}(u(k), [x]_e(j));$ 
8:       if contrainte outputs are not satisfying then
9:         reject the input sequence
10:      if equivalent outputs then
11:        select the optimal sequence within the equivalent outputs, reject the others;
12:         $\mathbb{L} := \mathbb{L} \cup [x]_e(j+1);$ 
13:         $\text{sensibilité} := \text{eval-sensibilité}([x]_e(j), u(k));$ 
14:         $\text{cout-intermédiaire} := \text{eval-critère}(\text{sensibilité});$ 
15:         $\text{cout} := \text{cout} \cup (\text{cout-intermédiaire}, u(k));$ 
16:  $\text{parcours-cout}(\text{cout});$ 
17: retourne( $u_{opt}$ )
```

où r représente le nombre d'horizon, t représente le nombre d'échantillonnage, l représente le nombre de contraintes. $[x_0]$ représente la condition initiale d'état et $[p_0]$ représente le domaine initial de paramètres.

Remarque Dans la ligne 1, l'étape initialisation concerne la préparation du calcul, y compris l'initialisation des états initiaux et des quelques constantes.

Dans la ligne 10, 11, c'est l'étape de la sélection de la séquence optimale parmi les trajectoires équivalentes. Des nombreuses évaluations de sensibilité sont faites à cette étape afin de sortir la séquence optimale.

La fonction parcours-cout cherche le point optimal utilisant un arbre d'exploration. La fonction retourne donne la séquence d'entrée la plus sensibilisante.

Bibliographie

- [Adrot 2002] O. Adrot, H. Janati-Idrissi and D. Maquin. *Fault detection based on interval analysis*. In 15th IFAC World Congress on Automatic Control, volume 15, pages 760–760, 2002.
- [Alessi 2007] E.M. Alessi, A. Farré, Á. Jorba, C. Simo and A. Vieiro. *Efficient usage of self validated integrators for space application*. In Technical report. Tech. rep, 2007.
- [Alhaj-Dibo 2004] M. Alhaj-Dibo, D. Maquin and J. Ragot. *Data validation of uncertain dynamic systems*. In 15th International Workshop on Principles of Diagnosis, DX’04, Carcassonne, France, 2004.
- [Banks 2009] H.T. Banks, M. Davidian, J.R. Samuels Jr and K.L. Sutton. An inverse problem statistical methodology summary. Springer, 2009.
- [Bedoui 2014] H. Bedoui, A. Kedher and K.B. Othman. *Interval analysis for the sensors faults detection of a nonlinear system*. In International conference on control, engineering and information technology, pages 180–184. IPCO, 2014.
- [Bellman 1952] R.E. Bellman. *On the theory of dynamic programming*. Proceedings of the National Academy of Sciences of the United States of America, vol. 38, no. 8, page 716, 1952.
- [Blesa 2014] J. Blesa, D. Rotondo, V. Puig and F. Nejjari. *FDI and FTC of wind turbines using the interval observer approach and virtual actuators/sensors*. Control Engineering Practice, vol. 24, pages 138 – 155, 2014.
- [Bokanowski 2010] O. Bokanowski, N. Forcadel and H. Zidani. *Reachability and minimal times for state constrained nonlinear problems without any controllability assumption*. SIAM Journal on Control and Optimization, vol. 48, no. 7, pages 4292–4316, 2010.
- [Burns 2007] J.A. Burns, E.M. Cliff and S.E. Doughty. *Sensitivity analysis and parameter estimation for a model of Chlamydia Trachomatis infection*. Journal of Inverse and Ill-posed Problems, vol. 15, no. 3, pages 243–256, 2007.
- [Campolongo 1999] F Campolongo, S Tarantola and A Saltelli. *Tackling quantitatively large dimensionality problems*. Computer physics communications, vol. 117, no. 1, pages 75–85, 1999.
- [Casella 2002] G. Casella and R.L. Berger. Statistical inference, volume 2. Duxbury Pacific Grove, CA, 2002.
- [Chabert 2014] G. Chabert. *IBEX : C++ library for constraint processing over real numbers*. <https://github.com/ibex-team/ibex-lib/>, 2014.
- [Chen 2012] J. Chen and R.J. Patton. Robust model-based fault diagnosis for dynamic systems, volume 3. Springer Science & Business Media, 2012.
- [Coton 2001] P. Coton, A. Bucharles, C. Jauberthie, Th. Lemoing and L. Planckaert. *CAIRE identification des dérivées de stabilités dynamiques, phase 2*, 2001.

- [Darboux 1876] G. Darboux. *Sur les développements en série des fonctions d'une seule variable*. Journal de Mathématiques pures et appliquées, pages 291–312, 1876.
- [de Figueiredo 2004] L.H. de Figueiredo and J. Stolfi. *Affine Arithmetic : Concepts and Applications*. Numerical algorithms, vol. 37, no. 1-4, 2004.
- [Drevelle 2015] V. Drevelle and P. Bonifait. *Interval-based fast fault detection and identification applied to radio-navigation multipath*. International Journal of Adaptive Control and Signal Processing, pages 1085–1090, 2015.
- [Eggers 2012] A. Eggers, N. Ramdani, N.S. Nedialkov and M. Fränzle. *Improving the SAT modulo ODE approach to hybrid systems analysis by combining different enclosure methods*. Software & Systems Modeling, pages 1–28, 2012.
- [Frank 1987] P.M. Frank. *Fault diagnosis in dynamic systems via state estimation a survey*. In System fault diagnostics, reliability and related knowledge-based approaches, pages 35–98. Springer, 1987.
- [Frank 1990] P.M. Frank. *Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy : A survey and some new results*. Automatica, vol. 26, no. 3, pages 459–474, 1990.
- [Frank 1997] P.M. Frank and X. Ding. *Survey of robust residual generation and evaluation methods in observer-based fault detection systems*. Journal of process control, vol. 7, no. 6, pages 403–424, 1997.
- [Friedrich 2006] P. Friedrich. Optimal design of experiments. SIAM, 2006.
- [Gaol 2013] Gaol. *Gaol :Not just another interval library*. 2013.
- [Gay 2006] O. Gay, D. Coeurjolly and N. Hurst. *Libaffa- C++ affine arithmetic library for GNU/Linux*. 2006.
- [Gertler 1998] J. Gertler. Fault detection and diagnosis in engineering systems. CRC press, 1998.
- [Gholami 2011] M. Gholami, H. Schioler and T. Bak. *Active fault diagnosis for hybrid systems based on sensitivity analysis and ekf*. In American Control Conference (ACC), 2011, pages 244–249. IEEE, 2011.
- [Hazewinkel 2001] M. Hazewinkel. Encyclopaedia of mathematics. Springer, 2001.
- [Hermite 1874] Ch Hermite. *Extrait d'une lettre de M. Ch. Hermite à M. Borchardt sur la transformation des formes quadratiques ternaires en elles-mêmes*. Journal für die reine und angewandte Mathematik, vol. 78, pages 325–328, 1874.
- [Houska 2013] B. Houska, M.E. Villanueva and B. Chachuat. *A validated integration algorithm for nonlinear ODEs using Taylor models and ellipsoidal calculus*. In Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, pages 484–489, Dec 2013.
- [Ingimundarson 2008] A. Ingimundarson, J.M. Bravo, V. Puig, T. Alamo and P. Guerra. *Robust fault detection using zonotope-based set-membership consistency test*. International journal of adaptive control and signal processing, vol. 23, 2008.

- [Isermann 1992] R. Isermann. *Estimation of physical parameters for dynamic processes with application to an industrial robot*. International Journal of Control, vol. 55, no. 6, pages 1287–1298, 1992.
- [Isermann 1993] R. Isermann. *Fault diagnosis of machines via parameter estimation and knowledge processing Tutorial paper*. Automatica, vol. 29, no. 4, pages 815 – 835, 1993.
- [Isermann 1997] R. Isermann. *Supervision, fault detection and fault diagnosis methods, an introduction*. Control engineering practice, vol. 5, no. 5, pages 639–652, 1997.
- [Isermann 2005] R. Isermann. *Model-based fault-detection and diagnosis—status and applications*. Annual Reviews in control, vol. 29, no. 1, pages 71–85, 2005.
- [Jauberthie 2002] C. Jauberthie. *Méthodologies de planification d’expériences pour systèmes dynamiques*. PhD thesis, Université de technologie de Compiègne, 2002.
- [Jauberthie 2013] C. Jauberthie and E. Chantery. *Optimal input design for a nonlinear dynamical uncertain aerospace system*. In IFAC Symposium on Nonlinear Control Systems, pages 469 – 474, Toulouse, France, 2013. IFAC.
- [Jaulin 1993] L. Jaulin and E. Walter. *Set inversion via interval analysis for nonlinear bounded-error estimation*. Automatica, vol. 29, 1993.
- [Jaulin 2000] L. Jaulin. *Le calcul ensembliste par analyse par intervalles et ses applications*. Habitation à la dirige de la recherche, Université d’Angers, 2000.
- [Jaulin 2001a] L. Jaulin, M. Kieffer, D. Olivier and W. Eric. Applied interval analysis, with exaples in parameter and state estimation, robust control and robotics. Springer Verlag London, London, 2001.
- [Jaulin 2001b] L. Jaulin and E. Walter. *Nonlinear bounded-error parameter estimation using interval computation*. In Granular computing, pages 58–71. Springer, 2001.
- [Jaulin 2005] L. Jaulin and D. Henrion. *Contracting optimally an interval matrix without loosing any positive semi-definite matrix is a tractable problem*. Reliable Computing, vol. 11, no. 1, pages 1–17, 2005.
- [Karim 2008] J. Karim, C. Jauberthie and M. Combacau. *Model-based fault detection method using interval analysis : application to an aeronautic test bench*. In 19th International Workshop on Principles of Diagnosis, DX’04, pages 269–273, 2008.
- [Keil 2009] C. Keil. *Profil/Bias*. 2009.
- [Kieffer 1998] M. Kieffer, L. Jaulin and E. Walter. *Guaranteed recursive nonlinear state estimation using interval analysis*. In 37th IEEE conference on decision and control, 1998.
- [Kieffer 1999] M. Kieffer. *Estimation ensembliste par analyse par intervalles Application à la localisation d’un véhicule*. PhD thesis, Université Paris Sud-Paris XI, 1999.
- [Kieffer 2003] M. Kieffer and E. Walter. *Guaranteed parameter estimation for cooperative models*. In Positive Systems, pages 103–110. Springer, 2003.
- [Kieffer 2005] M. Kieffer and E. Walter. *Interval analysis for guaranteed non-linear parameter and state estimation*. Mathematical and Computer Modelling of Dynamical Systems, vol. 11, no. 2, pages 171–181, 2005.

- [Kieffer 2011] M. Kieffer and E. Walter. *Guaranteed estimation of the parameters of nonlinear continuous-time models : Contributions of interval analysis*. International Journal of Adaptive Control and Signal Processing, vol. 25, no. 3, pages 191–207, 2011.
- [Kosut 1992] R.L. Kosut, M.K. Lau and S.P. Boyd. *Set-membership identification of systems with parametric and nonparametric uncertainty*. Automatic Control, IEEE Transactions on, vol. 37, no. 7, pages 929–941, 1992.
- [Lerch 2006] M. Lerch, G. Tischler, J.W.V. Gudenberg, W. Hofschuster and W. Krämer. *Filib++, a fast interval library supporting containment computations*. ACM Transactions on Mathematical Software (TOMS), vol. 32, no. 2, pages 299–324, 2006.
- [Lewis 1986] F.L. Lewis. Optimal estimation. Wiley, New york, 1986.
- [Li 2014] Q. Li, C. Jaubertie, L. Denis-vidal and Z. Cherfi. *Guaranteed state and parameter estimation for nonlinear dynamical aerospace models*. In Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on, volume 1, pages 519–527. IEEE, 2014.
- [Li 2015a] Q. Li, C. Jaubertie, L. Denis-vidal and Z. Cherfi. *A diagnosis scheme for dynamical systems : Approach by guaranteed parameter estimation*. In Informatics in Control, Automation and Robotics (ICINCO), 2015 12th International Conference on, volume 1, pages 330–335. IEEE, 2015.
- [Li 2015b] Q. Li, C. Jaubertie, L. Denis-vidal and Z. Cherfi. *Entrée optimale pour l'estimation de paramètre des systèmes dynamiques non linéaires avec application en aéronautique*. In Journées Nationales et doctorales de Modélisation, Analyse et Conduite des Systèmes dynamiques, JDMAC2015. JDMAC, Bourge, France, 2015.
- [Li 2015c] Q. Li, C. Jaubertie, L. Denis-vidal and Z. Cherfi. *Impact of optimized input on guaranteed parameter and state estimation in a bounded error context for nonlinear dynamical aerospace models*. Journal of Multidisciplinary Engineering Science and Technology (JMEST), vol. 2, pages 216 – 224, 2015.
- [Li 2015d] Q. Li, C. Jaubertie, L. Denis-vidal and Z. Cherfi. *Optimal initial state for fast parameter estimation in nonlinear dynamical systems*. In 9th IFAC Symposium on Biological and Medical Systems, BMS, pages 556 – 562. Berlin, Germany, 2015.
- [Lin 2007] Y. Lin and M.A. Stadtherr. *Validated solutions of initial value problems for parametric ODEs*. Applied Numerical Mathematics, vol. 57, no. 10, pages 1145–1162, 2007.
- [Lohner 1988] R.J. Lohner. *Einschließung der Lösung gewöhnlicher Anfangs-und Randwertaufgaben und Anwendungen*. PhD thesis, Universität Karlsruhe, 1988.
- [Magnus 2000] N. Magnus, K. P. Niels and R. Ole. *New developments in state estimation for nonlinear systems*. Automatica, vol. 36, no. 11, pages 1627 – 1638, 2000.
- [Maillet 2013] Q. Maillet, H. Xu, N. Ozay and R.M. Murray. *Dynamic state estimation in distributed aircraft electric control systems via adaptive submodularity*. In Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, pages 5497–5503. IEEE, 2013.

- [Makino 1996] M. Makino K. and Berz. *Remainder differential algebras and their applications*. Computational differentiation : techniques, applications and tools, pages 63–74, 1996.
- [Makino 2005] K. Makino and M. Berz. *Suppression of the wrapping effect by Taylor model-based verified integrators : Long-term stabilization by preconditioning*. In International Journal of Differential Equations and Applications, vol. 10, no. 4, pages 385–403, 2005.
- [Marx 2010] B. Marx, D. Maquin and J. Ragot. *State estimation and fault detection of uncertain systems based on an interval approach*. In Control and Fault-Tolerant Systems (SysTol), 2010 Conference on, pages 720–725, Nice, France, 2010.
- [Mehra 1976] R.K. Mehra. *Synthesis of optimal inputs for multiinput Multioutput (MIMO) systems*. In System identification : advances on case studies, volume 126, pages 211–229. Academic press, 1976.
- [Milanese 1991] M. Milanese and A. Vicino. *Estimation theory for nonlinear models and set membership uncertainty*. Automatica, vol. 27, no. 2, pages 403–408, 1991.
- [Moore 1966] R.E. Moore. Interval analysis. Prentice Hall, New Jersey, 1966.
- [Moore 1979] R.E. Moore and F. Bierbaum. Methods and applications of interval analysis. SIAM Studies in applied mathematics, 1979.
- [Moore 2009] R.E. Moore, R.B. Kearfott and M.J. Cloud. Introduction to interval analysis. SIAM, Philadelphia, 2009.
- [Morris 1991] M.D. Morris. *Factorial sampling plans for preliminary computational experiments*. Technometrics, vol. 33, no. 2, pages 161–174, 1991.
- [Müller 1927] M. Müller. *über das fundamentaltheorem in der theorie der gewöhnlichen differentialgleichungen*. Mathematische Zeitschrift, vol. 26, pages 619–645, 1927.
- [Nedialkov 1999a] N.S. Nedialkov and K.R. Jackson. Ode software that computes guaranteed bounds on the solution, volume 10, pages 197–224. Springer-Verlag, 1999.
- [Nedialkov 1999b] N.S. Nedialkov, K.R. Jackson and G.F. Corliss. *Validated solutions of initial value problems for ordinary differential equations*. Applied mathematics and computation, Elsevier, 1999.
- [Nedialkov 2001] N.S. Nedialkov, R.J. Kenneth and D.P. John. *An effective High-Order interval method for validating existence and uniqueness of the solution of an IVP for an ODE*. Reliable computing, vol. 7, pages 449–465, 2001.
- [Neumaier 2003] A. Neumaier. *Taylor forms-use and limits*. Reliable Computing, vol. 9, no. 1, pages 43–79, 2003.
- [Neumaier 2011] A. Neumaier. *Improving interval enclosures*, 2011. unpublished.
- [Nikoukhah 1998] R. Nikoukhah. *Guaranteed active failure detection and isolation for linear dynamical systems*. Automatica, vol. 34, no. 11, pages 1345 – 1358, 1998.
- [Nikoukhah 2000] R. Nikoukhah, S. L. Campbell and F. Delebecque. *Detection signal design for failure detection : a robust approach*. International Journal of Adaptive Control and Signal Processing, vol. 14, no. 7, pages 701–724, 2000.

- [Nikoukhah 2006] R. Nikoukhah and S.L. Campbell. *Auxiliary signal design for active failure detection in uncertain linear systems with a priori information*. Automatica, vol. 42, no. 2, pages 219 – 228, 2006.
- [Pasca 2010] I. Pasca. *Formally Verified Conditions for Regularity of Interval Matrices*. In Symposium on the Integration of Symbolic Computation and Mechanised Reasoning, Calculemus, volume 6167 of *LNAI*, pages 219–233, Paris, France, 2010. Springer.
- [Patton 1997] R.J. Patton and J. Chen. *Observer-based fault detection and isolation : robustness and applications*. Control Engineering Practice, vol. 5, no. 5, pages 671–682, 1997.
- [Pons 2008] R. Pons, C. Jaubertie, L. Trave-Massuyes and P. Goupil. *Interval analysis based learning for fault model identification. Application to control surfaces oscillatory failures*. In 22nd International Workshop on Qualitative Reasoning. Boulder, USA, 2008.
- [Raena 2011] F. Raena, R. Jiri and L. Taher. *Norms of interval matrices*, 2011.
- [Raïssi 2004a] T. Raïssi. *Méthodes ensemblistes pour l'estimation d'état et de paramètres*. PhD thesis, Université Paris XII Val de Marne, 2004.
- [Raïssi 2004b] T. Raïssi, N. Ramdani and Y. Candau. *Set membership state and parameter estimation for systems described by nonlinear differential equations*. Automatica, vol. 40, no. 10, pages 1771–1777, 2004.
- [Reppa 2011] V. Reppa and A. Tzes. *Fault detection and diagnosis based on parameter set estimation*. IET control theory & applications, vol. 5, no. 1, pages 69–83, 2011.
- [Rihm 1994] R. Rihm. *Interval methods for initial value problems in ODEs*. In in topics in validated computations : proceedings of the IMACS-GAMM international workshop on validated computations. University of Oldenburg, J. Herzberger, ed. Elsevier Studies in Computational Mathematics, Elsevier, Amsterdam, 1994.
- [Rump 2014] S.M. Rump. *Verified bounds for the p-norm condition number*. Reliable Computing, vol. 20, pages 45–52, 2014.
- [Saltelli 2008] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana and S. Tarantola. *Global sensitivity analysis : the primer*. Wiley, 2008.
- [Sandretto 2015] J.A.D. Sandretto and C. Alexandre. *Validated solution of initial value problem for ordinary differential equations based on explicit and implicit Runge-Kutta schemes*, 2015.
- [Skelboe 1974] S. Skelboe. *Computation of rational interval functions*. BIT Numerical Mathematics, vol. 14, no. 1, pages 87–95, 1974.
- [Stancu 2005] A. Stancu, V. Puig, P. Cugueró and J. Quevedo. *Benchmarking on Approaches to Interval Observation Applied to Robust Fault Detection*. In C. Jerermann, A. Neumaier and D. Sam, éditeurs, Global Optimization and Constraint Satisfaction, volume 3478 of *Lecture Notes in Computer Science*, pages 171–191. Springer Berlin Heidelberg, 2005.

- [Stol 1997] J. Stol and L.H. De Figueiredo. *Self validated numerical methods and applications*. In Monograph for 21st Brazilian Mathematics Colloquium, IMPA, Rio de Janeiro. Citeseer, 1997.
- [Stolfi 2003] J. Stolfi and L.H. de Figueiredo. *An introduction to affine arithmetic*. Trends in Applied and Computational Mathematics, vol. 4, no. 3, pages 297–312, 2003.
- [Tabatabaeipour 2009] S. Tabatabaeipour, A.P. Ravn, R. Izadi-zamabadi and T. Bak. *Active diagnosis of hybrid systems A model predictive approach*. In Control and Automation, 2009. ICCA 2009. IEEE International Conference on, pages 465–470, Christchurch, New zealand, 2009.
- [Tomas 2008] J. Tomas and T. Warwick. *Rigorous parameter reconstruction for differential equations with noisy data*. Automatica, vol. 44, no. 9, pages 2422 – 2426, 2008.
- [Verdière 2005] N. Verdière, L. Denis-Vidal, G. Joly-Blanchard and D. Domurado. *Identifiability and estimation of pharmacokinetic parameters for the ligands of the macrophage mannose receptor*. International Journal of Applied Mathematics and Computer Science, vol. 15, pages 517–526, 2005.
- [Willsky 1976] A.S. Willsky. *A survey of design methods for failure detection in dynamic systems*. Automatica, vol. 12, no. 6, pages 601–611, 1976.
- [Xue 2007] W. Xue, Y. Guo and X. Zhang. *A bank of Kalman filters and a robust Kalman filter applied in fault diagnosis of aircraft engine sensor/actuator*. In Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on, pages 10–10, Kumamoto, Japan, 2007.
- [Zhang 1989] X. Zhang. *Auxiliary signal design in fault detection and diagnosis*. Springer, 1989.
- [Zilinskas 2005] J. Zilinskas. In Comparison of packages for interval arithmetic, volume 16, pages 145–154. Information, 2005.

Contributions to the design of experiment, the estimation and active diagnosis for nonlinear dynamical systems with aeronautical application

Abstract : In this work, we will study the uncertainty integration problem in a bounded error context for the dynamic systems, whose input and the initial state have to be optimised so that some other operation could be more easily and better obtained.

This work is consisted of 6 chapters : the chapter 1 is an introduction to the general subject which we will discuss about. The chapter 2 represents the basic tools of interval analysis.

The chapter 3 est dedicated to state estimation and parameter estimation. We explain at the first, how to solve the ordinary differential equation using interval analysis, which will be the basic tool for the state estimation problem given the initial condition of studied systems. On the other side, we will look into the parameter estimation problem using interval analysis too. Based on a simple hypothesis over the uncertain variable, we calculate the system's parameter in a bounded error form, considering the operation of intervals as the operation of sets. Guaranteed results are the advantage of interval analysis, but the big time consumption is still a problem for its popularization in many non linear estimation field. We presents our founding techniques to accelerate this time consuming processes, which are called contractor in constraint propagation field. At the end of this chapter, different examples will be the test proof for our proposed methods.

Chapter 4 presents the searching for optimal input in the context of interval analysis, which is an original approach. We have constructed several new criteria allow such searching. Some of them are intuitive, the other need a theoretical proof. These criteria have been used for the search of optimal initial states and le better parameter estimation results. The comparisons are done by using multiple applications and the efficiency is proved by evidence.

In chapter 5, we applied the approaches proposed above in diagnosis by state estimation and parameter estimation. We have developed a complete procedure for the diagnosis. The optimal input design has been reconsidered in an active diagnosis context. Both state and parameter estimation are implemented using an aeronautical application in literature.

The last chapter gives a brief summary over the realized subject, some further research directions are given in the perspective section.

All the algorithms are written in C/C++ on a Linux based operation system.

Keywords : Interval analysis, optimal input design, active diagnosis, parameter estimation
