



HAL
open science

Analyse de sûreté de procédés multi-modes par des méthodes à base d'intervalles

Kyarash Shahriari

► **To cite this version:**

Kyarash Shahriari. Analyse de sûreté de procédés multi-modes par des méthodes à base d'intervalles. Sciences de l'ingénieur [physics]. Université Joseph-Fourier - Grenoble I, 2007. Français. NNT : . tel-00579496

HAL Id: tel-00579496

<https://theses.hal.science/tel-00579496v1>

Submitted on 24 Mar 2011

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.

Université Joseph Fourier

No. attribué par la bibliothèque

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

THESE

pour obtenir le grade de

DOCTEUR DE L'UJF

Spécialité : AUTOMATIQUE-PRODUCTIQUE

préparée au Laboratoire des Sciences pour la Conception, l'Optimisation et la Production

dans le cadre de l'École Doctorale :

Électronique, Électrotechnique, Automatique, Traitement du Signal

présentée et soutenue publiquement

par

Kyarash SHAHRIARI

le 3 Mai 2007

Titre :

**Analyse de Sûreté de Procédés Multi-Modes par des Méthodes
à base d'Intervalles**

Directeurs de thèse :

M. Jean-Marie Flaus (G-SCOP - UJF)

M. Olivier Adrot (G-SCOP - UJF)

JURY :

M. Yannick Frein Président

M. José Ragot Rapporteur

M. Lionel Boillereaux Rapporteur

M. Christophe Combastel Examineur

M. Jean-Marie Flaus Directeur de thèse

M. Olivier Adrot Co-encadrant

*à ma mère, mon père, ma famille
et à mon amour, Mojgan*

La théorie, c'est quand on sait tout et que rien ne fonctionne. La pratique, c'est quand tout fonctionne et que personne ne sait pourquoi. Ici, nous avons réuni théorie et pratique : rien ne fonctionne... et personne ne sait pourquoi !

Albert Einstein

Remerciements

Je tiens tout d'abord à exprimer mes remerciements envers mon père, ma mère, mes frères Keyhan et Soroosh et ma soeur Mana qui m'ont toujours soutenu tout au long de mon parcours depuis le premier jour.

Je tiens à remercier Christian Commault (responsable du Master EEATS) la première personne qui m'a accueilli au sein du Laboratoire d'Automatique de Grenoble comme étudiant de Master de Recherche ainsi que mes professeurs pendant mon Master : C. Commault, P. Ladet, S. Gentil, J.M. Flaus, M.L. Espinouse et M. Jacomino.

Mes remerciements s'adressent aussi à Jean-Marie Flaus et Olivier Adrot pour leurs conseils et leurs esprits critiques durant ma thèse. Merci également à mes collaborateurs pendant mes encadrements de TP et de TD : S. Ploix, J.F. Béteau et collègues du labo Melha, Salvador, Fermi, Sameh, Emmanuel, Laurent, Samir, Cindy, Mathieu, Abed, Alexandru, Zara et Shadi.

Je tiens à remercier l'équipe technique et administrative du laboratoire GIPSA (l'ancien LAG), Daniel, Marie-Thérèse, Virginie et Patricia. Je remercie également tous les gens du nouveau laboratoire G-SCOP que j'ai rejoint depuis le premier janvier 2007.

Pour finir, je remercie Mojgan, la dernière arrivée, qui a supporté ma mauvaise humeur pendant ces derniers mois.

Merci à tous et à toutes. Je ne vous oublierai jamais.

Table des matières

1	Introduction	1
1.1	Problématique	1
1.2	Modèle intervalle	3
1.3	Contribution de ce travail de thèse	6
2	État de l’art	9
2.1	Introduction	9
2.2	Systèmes et Modèles	9
2.3	Modèles Mathématiques	11
2.3.1	Modèle Déterministe	12
2.3.2	Modèle Probabiliste	13
2.3.3	Modèle Ensembliste	14
2.3.4	Conclusion	19
2.4	Simulation des Modèles	20
2.4.1	Simulateur Quantitatif	20
2.4.2	Simulateur Qualitatif	21
2.4.3	Simulateur Semi-Quantitatif	21
2.4.4	Simulateur d’un modèle intervalle	21
2.4.5	Conclusion	24
2.5	Modélisation Non-Linéaire	24
2.5.1	Modèles Non-Linéaires	25
2.5.2	Réseaux de Neurones	25
2.5.3	Modèles Multi-Experts	26
2.5.4	Modèles Flous	26
2.5.5	Modèles Hybrides : Approche Multi-Mode	26
2.5.6	Conclusion	30
2.6	Modèles Multi-Modes Intervalle	31
2.7	Conclusion	31
3	Modèle Intervalle	33
3.1	Introduction	33

3.2	Outil Intervalle	34
3.2.1	Pourquoi un intervalle?	34
3.2.2	Extension intervalle d'une fonction réelle	37
3.2.3	Forme alternative pour représenter un intervalle	40
3.2.4	Effet d'enveloppement dans le calcul par intervalle	42
3.3	Identification d'un Modèle Intervalle	44
3.3.1	Jeu de données expérimental	45
3.3.2	Structure de modèle	46
3.3.3	Sémantique d'identification	56
3.3.4	Critère d'optimisation	59
3.3.5	Calcul des paramètres d'un modèle intervalle	60
3.3.6	Effet de Mémoire	66
3.3.7	Exemple Numérique	69
3.3.8	Modèle intervalle et paire d'inégalités	75
3.4	Simulation de Modèles Intervalle	75
3.4.1	Sémantique de Simulation	77
3.4.2	Méthode numérique pour calculer l'enveloppe	77
3.4.3	Prédiction de l'enveloppe englobante : Modèle <i>FFIR</i>	80
3.4.4	Prédiction de l'enveloppe englobante : Modèle <i>FIO</i>	82
3.4.5	Résultats et Discussions	85
3.5	Conclusion	86
4	Modèle Multi-Mode	89
4.1	Introduction	89
4.2	Objectif	90
4.3	Identification Multi-Modes : commutation déterminée	92
4.3.1	Jeu de données	92
4.3.2	Structure de modèle	94
4.3.3	Sémantique d'identification	94
4.3.4	Critère d'optimisation	94
4.3.5	Calcul des paramètres du modèle	95
4.4	Simulation Multi-Modes : commutation déterminée	95
4.4.1	Sémantique de simulation	95
4.4.2	Méthode numérique pour calculer l'enveloppe	96
4.5	Identification Multi-Modes : commutation indéterminée	98
4.5.1	Variables caractéristiques	100
4.5.2	Partitionnement du régime opératoire	100
4.5.3	Détermination des domaines de validité des modèles locaux	108
4.5.4	Agrégation des modèles	116
4.6	Simulation Multi-Modes : commutation indéterminée	118
4.6.1	Sémantique de simulation	118

4.6.2	Calcul de l'enveloppe englobante	118
4.7	Exemple : Système multi-mode	122
4.8	Conclusion	128
5	Conclusion et Perspectives	131
A	Modèles Mathématiques	137
B	Représentation géométrique de la sémantique d'identification	143
	Bibliographie	148

Chapitre 1

Introduction

Un problème sans solution est un problème mal posé.

Albert Einstein

1.1 Problématique

Les activités industrielles et humaines font régulièrement les grands titres des actualités en générant bon nombre d'incidents, d'accidents ou d'événements critiques ou catastrophiques, à cause de défaillances humaines, matérielles ou d'agressions liées à l'environnement du site étudié (risque naturel, effet domino, ...). Pour tenter de réduire au mieux le niveau de risque et le rendre acceptable au niveau réglementaire et par l'opinion publique, des méthodes ont été développées pour évaluer les risques potentiels, prévoir l'occurrence des problèmes et minimiser les impacts lorsqu'ils arrivent. L'ensemble de ces méthodes répondant à l'objectif de la maîtrise des risques définit la discipline de la *sûreté de fonctionnement*.

Lorsqu'un système physique est en fonctionnement, un des objectifs d'une analyse de sûreté est d'être sûr que les variables caractérisant l'état du système suivent une certaine trajectoire ou restent dans une zone donnée compte tenu de toutes les perturbations et les incertitudes sur les paramètres et les mesures du système. Cette notion est vaste puisqu'elle peut s'appliquer à la trajectoire physique d'un bras mobile de robot devant éviter tous les obstacles pouvant se présenter devant lui (mur, opérateur) ou au domaine de stabilité d'un système asservi que les ingénieurs se doivent de respecter.

Une solution originale pour ce problème est proposée dans ce document à l'aide d'une approche ensembliste qui consiste à calculer l'ensemble des trajectoires possibles générées en présence des perturbations dues à l'environnement du système et à la méconnaissance des modèles du système se traduisant par des incertitudes sur la valeur de certains de ses paramètres. Dans le cas de notre problème, les incertitudes peuvent porter sur les valeurs des paramètres, l'état initial ou les mesures du système. Pour garantir le résultat

de l'analyse de sûreté, toutes les trajectoires doivent être envisagées, pour toutes les perturbations et les incertitudes considérées.

Si le système est simple, cet ensemble de trajectoires, appelé *enveloppe* par la suite, peut être déterminé de façon analytique. Cependant, pour les systèmes plus complexes, des approches numériques doivent être envisagées et pour cela, les incertitudes doivent être caractérisées numériquement.

Dans une approche ensembliste, les valeurs candidates pour une variable incertaine peuvent être caractérisées de deux façons différentes :

- La première consiste à encadrer toutes les valeurs possibles de la variable par un domaine de forme déterminée, possédant des propriétés particulières, comme par exemple les boîtes, les ellipsoïdes ou les polytopes.
- La seconde consiste à encadrer, de façon extérieure ou intérieure, le domaine de la variable par une union de domaines simples, généralement des boîtes.

Après avoir caractérisé les incertitudes sur les paramètres, l'état initial et les variables observées par l'une de ces deux méthodes - forme déterminée ou union de domaines simples-, on évalue les domaines des variables d'état par rapport à la dynamique du système. À chaque étape de l'évaluation, les images de ces domaines sont ensuite englobées par la forme initiale choisie pour les domaines. L'ensemble des domaines calculés nous donne l'enveloppe des trajectoires possibles des variables d'état du système au cours de temps.

Pour mieux l'illustrer, on utilise comme exemple un système pédagogique de type bacs présenté sur la figure 1.1.

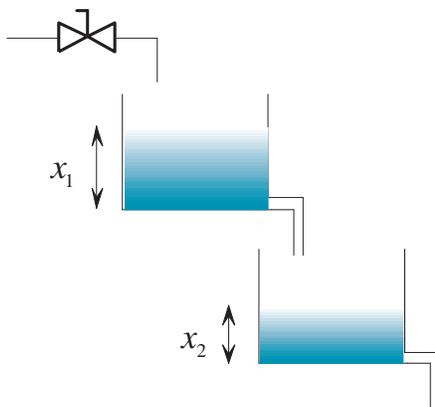


FIG. 1.1 – Exemple de deux bacs en série.

Le système est composé de deux bacs d'eau en série. Le bac du haut est alimenté en eau et le débit qui correspond à l'entrée du système, est commandé par une vanne. Le bac du bas est alimenté directement par la sortie du bac du haut. Les variables d'état x_1 et x_2 du système sont les hauteurs d'eau dans les deux bacs et l'objectif est d'être sûr que l'eau ne déborde jamais des bacs. Pour analyser ce système, on envisage plusieurs

types d'incertitudes.

Le premier type d'incertitude concerne les mesures. Pour déterminer l'état initial du système, on a besoin de mesurer les valeurs initiales de x_1 et x_2 . Cette mesure peut ne pas être précise et donc, les valeurs mesurées de x_1 et x_2 sont des valeurs approximatives. C'est aussi le cas pour le débit d'eau d'entrée qui est mesuré par un débitmètre.

Le deuxième type d'incertitude concerne la dynamique du système. Ce système est généralement représenté par un modèle du deuxième ordre. Les paramètres de ce modèle sont souvent choisis comme des constantes réelles invariantes dans le temps. Pourtant, ces valeurs sont des approximations des valeurs réelles des paramètres. Par exemple, dans un tel modèle, on considère généralement que la section de chaque bac est constante, ce qui n'est pas le cas dans un système réel. Un autre exemple concerne le débit d'eau qui sort de chaque bac. On suppose souvent par simplification, lorsqu'on linéarise une non-linéarité autour d'un point de fonctionnement, que ce débit est une fonction linéaire de la hauteur d'eau dans le bac et on ignore certains phénomènes physiques qui affectent ce débit (formation de bulles au niveau de la restriction, ...).

Dans le cas où on représente les paramètres ou les observations mal connus par des variables incertaines caractérisées non pas par des valeurs constantes, mais par des supports intervalle de valeurs admissibles, on obtient un modèle ensembliste ou intervalle calculant l'enveloppe des trajectoires du système, c'est-à-dire l'ensemble des valeurs possibles des variables caractérisant l'état du système¹.

1.2 Modèle intervalle

Pour évaluer un système et analyser son comportement, on a besoin d'un modèle du système². Le modèle du système définit les liens entre les variables physiques du système. Si les liens sont définis par des équations mathématiques, le modèle est un modèle mathématique. Ce modèle peut être un modèle dans lequel les paramètres sont parfaitement connus ou un modèle dans lequel les valeurs des paramètres sont mal connues.

On reprend l'exemple de la figure 1.1. On suppose que la dynamique du système est parfaitement connue et donc, on connaît les valeurs exactes des paramètres. Le comportement du système est représenté par le modèle d'état discret suivant :

$$x_{k+1} = A \times x_k + B \times u_k, \quad x_k \in \mathbb{R}^{2 \times 1}, \quad u \in \mathbb{R}^{1 \times 1}$$
$$A = \begin{bmatrix} 0,8187 & 0 \\ 0,1722 & 0,9048 \end{bmatrix} \quad B = \begin{bmatrix} 1,8127 \\ 0,1811 \end{bmatrix} \quad (1.2.1)$$

On veut calculer l'enveloppe de chaque variable d'état du système lorsque l'état

¹Une autre approche possible pour prendre en compte des incertitudes est d'utiliser un modèle probabiliste. Des inconvénients de ce type de modèle sont donnés dans la section 2.3.2

² On donnera les définitions d'un système, d'un modèle et de différents types de modèles plus précisément dans le chapitre suivant.

initial est mal connu et appartient à un domaine donné. Ce domaine est caractérisé par le vecteur d'intervalle $[x_0] = [[0 \ 1][0 \ 1]]^T$. Pour une commande TOR³ sur la vanne d'entrée telle que :

$$\begin{aligned} u_k &= 1, \quad k < 9 \\ u_k &= 0, \quad k \geq 9 \end{aligned}$$

on calcule l'enveloppe des variables d'état du système. Sur la figure 1.2, les enveloppes des variables x_1 et x_2 ont été calculées et sur la figure 1.3, une zone de fonctionnement du système a été représentée. Sur cette seconde figure, on voit également pour comparaison, les trajectoires d'un certain nombre de points pris aléatoirement qui appartiennent initialement au domaine de l'état initial du système. Les trajectoires de ces points ont toujours été contenues dans l'enveloppe des variables d'état du système. Sur cette figure, la zone interdite est une zone où les variables d'état du système ne doivent pas entrer. Dans l'exemple des deux bacs où x_1 et x_2 représentent les hauteurs physiques de l'eau dans chaque bac, leurs valeurs ne doivent pas dépasser la hauteur physique de chaque bac. Dans l'exemple précédent, l'incertitude sur les paramètres du modèle n'est pas prise en compte.

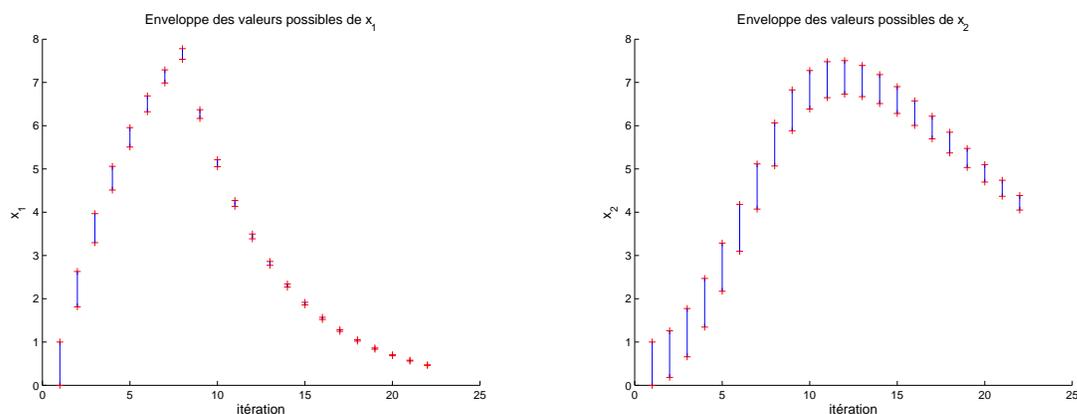


FIG. 1.2 – Enveloppes des variables d'état du système.

Pour une analyse de sûreté, les incertitudes sur les paramètres du modèle doivent aussi être prises en compte. Dans l'approche ensembliste par intervalle, on caractérise chacun des paramètres du modèle par un intervalle. Par exemple, si on ajoute $\pm 0,5\%$ d'incertitude sur tous les paramètres du modèle précédent, le comportement du modèle sera représenté par le modèle intervalle ci-dessous :

$$[x]_{k+1} = [A] \times [x]_k + [B] \times u_k$$

³Tout-Ou-Rien

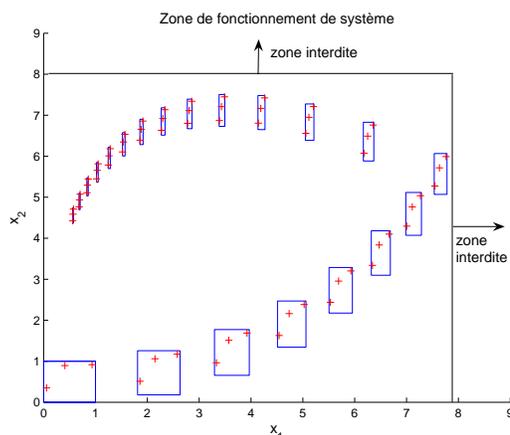


FIG. 1.3 – Zone de fonctionnement du système.

$$[A] = \begin{bmatrix} [0, 8146 & 0, 8228] & [0 & 0] \\ [0, 1713 & 0, 1730] & [0, 9003 & 0, 9093] \end{bmatrix} \quad [B] = \begin{bmatrix} [1, 8036 & 1, 8217] \\ [0, 1802 & 0, 1820] \end{bmatrix}$$

Si on évalue l'image de la boîte de l'état initial du système par le modèle d'état incertain en utilisant l'outil intervalle⁴, on obtient des enveloppes des variables d'état qui sont plus larges que les précédentes. Cela est dû à l'imprécision plus grande du modèle. Sur la figure 1.4, les enveloppes de x_1 et x_2 ont été représentées et sur la figure 1.5, une zone de fonctionnement de ce système a été représentée. Sur cette seconde figure, un point aléatoire a été choisi dans le domaine de l'état initial du système. Après le premier pas de calcul, l'image du point par rapport à la dynamique incertaine du système a été caractérisée par une boîte et ensuite, cette boîte a été évaluée. A chaque étape de calcul, l'image de la boîte est bien comme attendu dans l'enveloppe des trajectoires des variables d'état du système.

Les enveloppes calculées par un modèle intervalle surestiment très souvent le domaine réel des variables d'état du système. Représenter une forme quelconque par une boîte et utiliser l'outil intervalle pour calculer l'image de cette boîte sont deux sources de surestimations sur les enveloppes calculées. De manière générale, l'image d'un ensemble de points par rapport à une fonction réelle n'est pas une boîte. Si on représente cette image par une boîte (un vecteur d'intervalles), on introduit par approximation extérieure des valeurs supplémentaires dans le domaine calculé. De plus, si on utilise l'outil intervalle pour calculer l'image d'un ensemble de points, cet outil ne garantit pas que la boîte calculée soit la boîte la plus petite qui englobe l'image des points. Ceci est la deuxième source de surestimation sur les enveloppes calculées. Dans le cas de l'utilisation des modèles récursifs, comme un modèle d'état, ce pessimisme peut s'accumuler au cours des itérations et peut conduire à l'explosion des domaines calculés. Cet effet s'appelle *l'effet d'enveloppement* qui sera expliqué plus en détail dans la section 3.2.4.

Dans les exemples précédents, après chaque itération, le domaine des variables d'état

⁴On développera l'outil intervalle dans la section 3.2

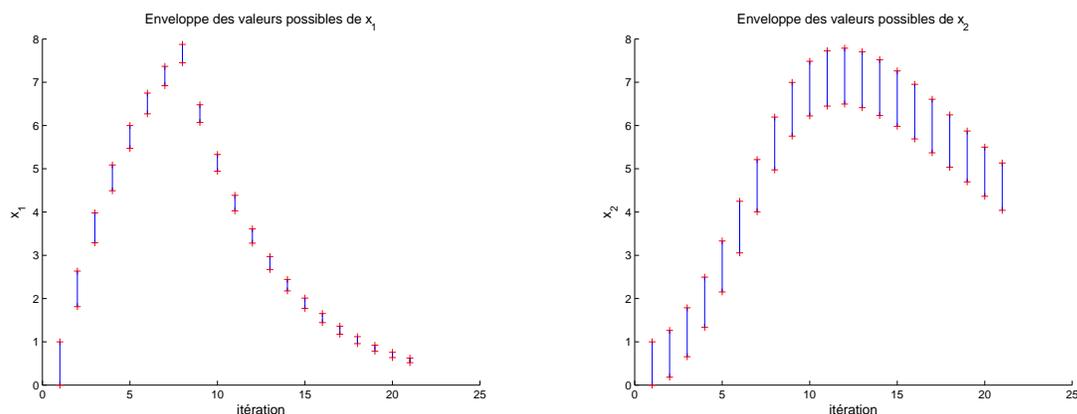


FIG. 1.4 – Enveloppes des variables d’état du système lorsque l’état initial et le comportement du système sont mal connus.

est ramené à sa forme initiale (une boîte). Autrement dit, après chaque itération, on a ajouté des domaines supplémentaires aux domaines des variables d’état. L’effet d’enveloppement n’était cependant pas suffisamment important ici pour que les enveloppes calculées explosent.

1.3 Contribution de ce travail de thèse

Dans les sections précédentes, on a expliqué brièvement le contexte de ce travail de thèse. On a aussi vu comment un modèle intervalle peut être utilisé pour trouver l’enveloppe englobante des trajectoires possibles des variables d’état du système et vérifier si elles restent à l’intérieur d’une zone donnée. Pour cela, on avait supposé que le modèle du système est donné et que l’incertitude sur l’état initial des variables d’état, les paramètres et les variables observées du système est connue.

Pourtant, le problème d’identification de systèmes incertains reste toujours ouvert. Ce problème qui est l’objet de cette thèse consiste à répondre à quatre questions.

La première question concerne la structure du modèle. Cette structure peut être choisie parmi les structures qui existent dans la littérature comme les modèles *ARX*, *ARMAX*, etc.

La deuxième question concerne les paramètres du modèle et l’approche numérique utilisée pour calculer les valeurs des paramètres. Généralement, l’identification de paramètres d’un modèle déterministe revient à une estimation optimale des valeurs des paramètres du modèle au sens d’un critère choisi. Cependant, l’identification d’un modèle intervalle consiste à calculer un ensemble des valeurs possibles pour chacun des paramètres du modèle tel qu’il explique le comportement du système à tous les instants. Cet ensemble est ensuite caractérisé par un intervalle.

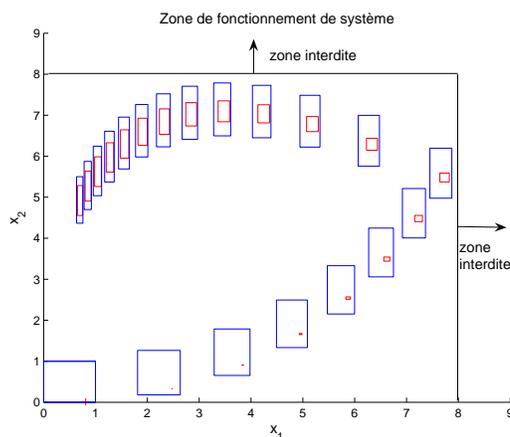


FIG. 1.5 – Zone de fonctionnement du système.

La troisième question concerne le domaine de variation (le support) de chacun des paramètres du modèle. Les bornes des supports des paramètres doivent être optimisées compte tenu de l'objectif d'appréhender toutes les trajectoires possibles du système, de façon à ce que l'enveloppe des trajectoires des variables d'état du système soit l'enveloppe la plus petite possible au sens d'un critère à définir. Cela nous amènera à résoudre un problème d'optimisation sous contraintes.

On peut toujours utiliser un modèle linéaire intervalle avec une structure simple pour expliquer les comportements complexes et non-linéaires d'un système. La conséquence est d'avoir un modèle moins précis⁵ par rapport à un modèle non-linéaire avec une structure plus complexe. Pour bénéficier des avantages d'une structure simple pour le modèle ainsi que pour améliorer la précision du modèle (diminuer la taille de l'enveloppe), la partie non-linéaire peut être divisée en plusieurs morceaux, chacun expliqué par un modèle local. L'ensemble des modèles locaux détermine le modèle multi-mode du système.

La dernière question à laquelle répondre est de trouver et de proposer une méthode pour diviser la partie non-linéaire en plusieurs morceaux telle que chaque morceau puisse être expliqué par un modèle linéaire. Sur la figure 1.6, une courbe non-linéaire par rapport à l'entrée u_k a été représentée tout d'abord par un modèle mono-mode intervalle et ensuite par un modèle multi-mode intervalle. On peut voir que dans le deuxième cas, la taille de l'enveloppe englobante est plus petite que pour le premier cas.

On essaie dans ce travail de thèse d'apporter une solution au problème d'identification de systèmes incertains dans un objectif d'analyse de sûreté de système. Ce document est constitué de quatre chapitres :

L'état de l'art. Dans ce chapitre, on présentera la définition et les différents types de modèles. Ensuite, on se concentrera sur les modèles mathématiques et on étudiera les travaux liés au problème d'identification. On étudiera également les travaux

⁵La précision d'un modèle ensembliste est une fonction de la taille de l'enveloppe calculée par le modèle et sera définie dans le chapitre trois.

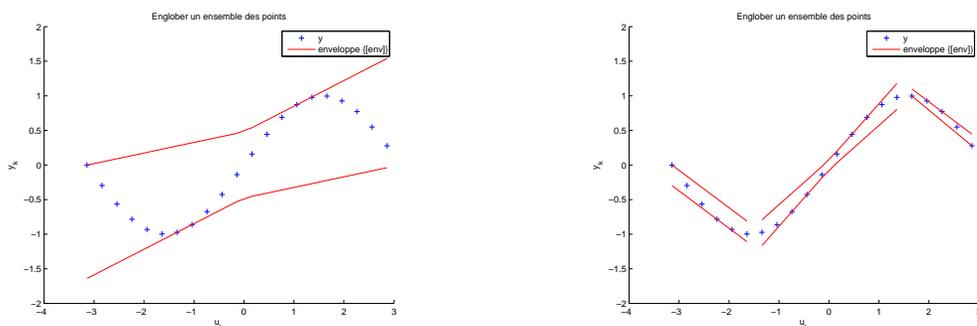


FIG. 1.6 – Modélisation mono-mode et multi-mode.

effectués dans le contexte des modèles hybrides et multi-mode.

Identification et simulation ensembliste. Dans ce chapitre, on proposera une méthode d'identification de systèmes incertains par l'approche ensembliste dans le but de déterminer les supports de variation des paramètres du modèle.

Identification et simulation multi-mode. Un modèle linéaire intervalle est, en général, capable d'expliquer le comportement linéaire ainsi que non-linéaire du système. Cependant, dans ce chapitre, pour améliorer la précision du modèle intervalle, un modèle multi-mode sera utilisé et une approche méthodologique sera proposée pour identifier ce modèle.

Conclusion et Perspectives. Dans ce dernier chapitre, une conclusion sera apportée sur l'ensemble de ce travail de thèse. On va également présenter quelques perspectives de ce travail.

Chapitre 2

État de l'art

*Si j'ai vu plus loin, c'est que j'étais assis sur
des épaules de géants.*

Isaac Newton

2.1 Introduction

Ce chapitre propose un état de l'art des travaux sur l'identification et la simulation de systèmes via différentes approches. Dans la deuxième section, on présentera les concepts d'un système et d'un modèle et en troisième section, les modèles mathématiques seront étudiés. Le modèle mathématique d'un système représente le comportement du système. Cependant, à cause de l'erreur de modèle, la sortie évaluée par le modèle est une estimation de la sortie du système à expliquer. Dans cette section, les différentes approches qui existent pour prendre en compte l'incertitude dans un modèle mathématique seront aussi analysées. Pour évaluer un modèle, on a besoin d'un simulateur ; en section quatre, ces simulateurs seront présentés. La section cinq étudie l'identification des systèmes complexes et non-linéaires et en dernière section, on étudiera les modèles multi-mode incertains.

2.2 Systèmes et Modèles

Brièvement, un *système* est un ensemble d'entités/objets, réels ou abstraits, fonctionnant de manière unitaire et en interaction ou liaison permanente avec au moins un autre composant/élément. En ingénierie, le terme *système* est utilisé pour un ensemble complexe et structuré de composants mécaniques, électroniques et informatiques en interaction permanente et assurant une fonction d'usage. Dans un système physique, un certain nombre de variables peuvent être observées. Les signaux observables qui nous intéressent s'appellent des *sorties*. Le système est affecté par des stimuli externes. Les signaux externes qui peuvent être manipulés s'appellent des *entrées*. Les autres s'appellent

des *perturbations*. Quand on modélise un système, on a besoin de définir les interactions entre les variables physiques du système. Les relations mathématiques entre les variables définissent alors un *modèle* du système. Par cette définition et pour un système donné, plusieurs types des modèles avec différents degrés de formalisme mathématique peuvent être définis. L'application visée détermine le degré raisonnable de complexité du modèle. L. Ljung dans [Ljung, 1999] a classé les modèles en quatre catégories.

- *Les modèles mentaux* sont des modèles dans lesquels aucun formalisme mathématique n'est utilisé. Ces modèles sont employés largement dans la vie quotidienne. Un exemple de ce type de modèles est l'évolution de la température à l'intérieur d'une maison. Pendant les grosses chaleurs de l'été, si on allume la climatisation, la température baisse et si on l'éteint, la température monte.
- Certains systèmes peuvent être décrits par des tables et des graphes et conduisent à des *modèles graphiques*. Par exemple, le comportement non-linéaire d'un composant électronique peut être facilement modélisé par un modèle graphique. Ces modèles s'utilisent plutôt dans la conception de produits en ingénierie.
- Une modélisation plus fine nécessite que les relations entre les variables physiques soient décrites par des équations mathématiques comme des équations aux différences ou différentielles. Ces modèles sont des *modèles mathématiques* ou *analytiques*. Les modèles mathématiques peuvent être à temps-continu ou à temps-discret, à paramètres localisés ou distribués, linéaires ou non-linéaires, etc ... signifiant le type des équations utilisées dans le modèle. Les modèles mathématiques sont utilisés dans presque tous les domaines de la science et surtout en ingénierie et en physique. La nature et la complexité des modèles utilisés dépendent des objectifs visés. Par exemple, les modèles mathématiques peuvent servir à effectuer des simulations et prédire le comportement du système étudié à des fins de commande, de diagnostic, de sûreté de fonctionnement etc. Pour étudier l'aspect philosophique des modèles mathématiques, on propose l'ouvrage "The logic of Scientific Discovery" de Popper [Popper, 1934].
- Le modèle d'un système utilisé dans une simulation sur ordinateur est un logiciel. Pour un système complexe, ce logiciel peut être constitué de plusieurs sous-programmes interconnectés et de tables de conversions qu'il n'est pas possible de représenter analytiquement. Le terme *modèle informatique* est utilisé pour une telle description informatisée. Les modèles informatiques sont de plus en plus utilisés. Ils sont proposés surtout dans le domaine de la prise de décision pour les systèmes complexes.

Il faut indiquer que cette classification n'est pas une classification stricte. On peut naturellement avoir un modèle qui se situe à la fois dans plusieurs catégories. Par exemple, les *modèles flous* sont des modèles à la fois mathématiques et mentaux. Dans la prochaine section, on étudiera plus précisément les modèles mathématiques que l'on va utiliser dans la suite de ce travail.

2.3 Modèles Mathématiques

Si dans le modèle d'un système, les relations entre les variables physiques sont décrites par des équations mathématiques, celui-ci appartient à la catégorie des modèles mathématiques. Il existe généralement une différence entre le comportement réel du système et le comportement représenté par le modèle, ou plus exactement un écart entre la sortie du système et son estimation obtenue à partir du modèle utilisé. Cet écart s'appelle par la suite, *l'erreur de modélisation*.

Les modèles mathématiques peuvent être classifiés en plusieurs catégories par rapport à la nature des équations mathématiques qui lient les variables du modèle. Les quatre grandes caractéristiques des modèles mathématiques sont la dynamique du modèle, la linéarité par rapport aux paramètres du modèle, la distribution des paramètres et la notion du temps dans les équations du modèle. Une autre catégorie pour les modèles mathématiques se fait par rapport à la façon dont l'erreur de modélisation est prise en compte. J. Armengol Liobet dans sa thèse [Armengol, 1999] a classé les modèles mathématiques en cinq catégories :

- Statique ou Dynamique
- Linéaire ou Non-Linéaire
- à Paramètres Localisés ou à Paramètres Distribués
- à Temps Continu ou Discret
- Déterministe ou Probabiliste

La cinquième catégorie classe les modèles mathématiques par rapport à la façon de prendre en compte l'erreur de modélisation. Cependant, il convient d'ajouter *les modèles ensemblistes* à cette catégorie, puisque dans ceux-ci, l'erreur de modélisation est traitée différemment en comparaison des modèles déterministes et probabilistes.

Par exemple, un modèle dynamique peut représenter le comportement d'un système par des équations aux différences ou différentielles. Les équations différentielles ordinaires peuvent être utilisées pour un modèle à paramètres localisés et les équations aux dérivées partielles pour un modèle à paramètres distribués. Également, on peut représenter le comportement d'un système par un modèle statique non-linéaire en temps continu. Les paramètres de ce modèle peuvent être choisis localisés et l'incertitude peut être caractérisée par des intervalles.

L'imprécision d'un modèle peut être due à différentes raisons : mauvaise connaissance ou connaissance partielle du système physique, hypothèses simplificatrices, phénomènes négligés, ... Ceci génère des erreurs de modélisation qui peuvent s'interpréter comme des erreurs sur les valeurs des paramètres du modèle ou sur le choix de sa structure. Ces erreurs peuvent être prises en compte et représentées en les modélisant sous la forme d'incertitudes :

- **Incertitude paramétrique du modèle.** Dans le cas où les paramètres du modèle sont mal connus, le modèle ne calcule pas la valeur juste de la sortie du système. C'est aussi le cas quand on utilise un modèle temps-invariant par rapport aux paramètres de modèle lorsque la dynamique du système varie dans le temps.
- **Incertitude non-paramétrique du modèle.** C'est le cas où la structure du modèle n'est pas cohérente avec le comportement du système. Un exemple pour ce cas correspond au fait de modéliser un système du deuxième ordre par un modèle du premier ordre.

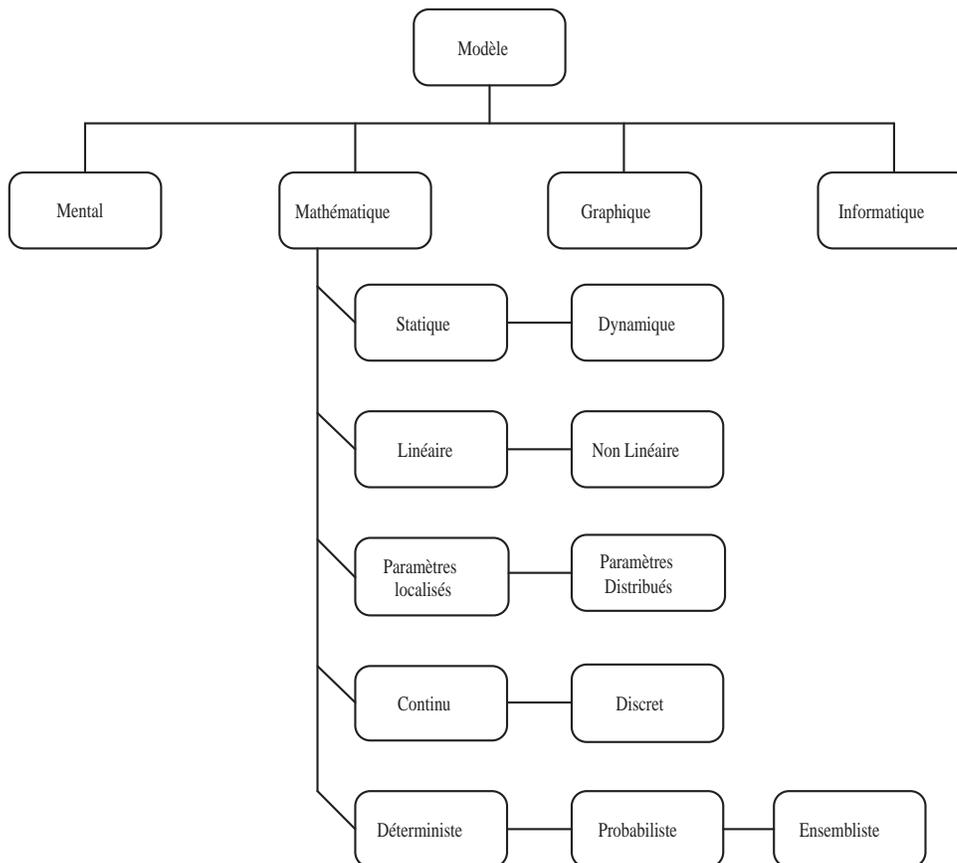


FIG. 2.1 – Classification des modèles mathématiques.

Pour une analyse de sûreté, les incertitudes paramétrique et non-paramétrique doivent être prises en compte. Dans la suite, on étudiera les modèles mathématiques par rapport à la façon dont l'erreur de modélisation est prise en compte dans le modèle.

2.3.1 Modèle Déterministe

Un *modèle déterministe* est un modèle dans lequel les paramètres sont des constantes réelles en temps-invariant. A chaque instant, la sortie d'un modèle déterministe dans

l'espace de sortie est un point ; on la qualifie donc de précise¹. Cependant, très souvent à cause de l'erreur de modélisation, la sortie estimée n'est pas juste ; autrement dit, il existe une différence entre la sortie du système et celle du modèle. Dans un modèle déterministe, les paramètres du modèle sont optimisés de telle façon qu'une fonction dépendant de l'erreur de modélisation, comme l'énergie de l'erreur, soit minimale.

Les travaux sur le problème d'identification des modèles déterministes sont vastes. Parmi les ouvrages sur ce sujet, on peut citer [Davis, 1985] et [Soderstrom & Stoica, 1987] qui ont traité le problème d'un point de vue théorique, tandis que [Box & Jenkins, 1970] et [Spriet & Vansteenkiste, 1982] l'ont fait avec une approche plutôt orientée "applications". L'ouvrage de [Kashyap & Rao, 1976] s'est focalisé sur la sélection et la validation des modèles alors que [Ljung & Soderstrom, 1983] et [Young, 1984] ont proposé des techniques d'identification recursive. On peut aussi citer plusieurs travaux dans les disciplines diverses comme [Malinvaud, 1980] en économie, [Godfrey, 1983] en biologie, [Dudley, 1983] en électromagnétique et [Markel & Gray, 1976] en traitement du signal. Une bibliographie plus détaillée en identification de systèmes se trouve dans [Ljung, 1999]. Pourtant, la question " Comment l'incertitude paramétrique et non-paramétrique peut être prise en compte dans les modèles mathématiques ?" reste ouverte dans ce cadre.

2.3.2 Modèle Probabiliste

Dans un modèle probabiliste, l'erreur de modélisation est caractérisée par une loi de distribution sur la sortie du modèle. Par conséquent et contrairement à un modèle déterministe dans lequel pendant la simulation, l'erreur de modélisation est inconnue, on peut ici déterminer une enveloppe pour la sortie du système par rapport à la probabilité d'existence de la sortie du système dans cette enveloppe.

Actuellement, les modèles probabilistes sont les modèles les plus utilisés dans presque tous les domaines de la science. Une des raisons importantes de ce fait est l'ancienneté des modèles probabilistes par rapport aux autres types de modèles. Les modèles probabilistes sont considérés généralement comme une extension des modèles déterministes.

Sur le sujet d'identification des modèles probabilistes, les ouvrages de [Anderson, 1994], [Hannan, 1970] et [Brillinger, 1981] peuvent être cités. Une grande partie des ouvrages cités précédemment sur l'identification des modèles déterministes traitent aussi le problème d'identification des modèles probabilistes.

Cependant, les modèles probabilistes souffrent de trois inconvénients :

- Dans certains cas, il n'est pas possible de déterminer une loi de distribution sur l'erreur de modélisation.
- Bien qu'on arrive à définir une telle loi, il est souvent difficile de respecter les propriétés de cette distribution lorsque les variables du modèle évoluent.
- Même si on arrive à définir une loi de distribution sur l'erreur de modélisation ainsi

¹La sortie d'un modèle est précise quand à chaque instant, la valeur de la sortie du modèle est une valeur bien connue à l'opposé des variables probabilistes ou ensemblistes.

que respecter les propriétés de cette distribution pendant l'évolution des variables du modèle, il n'y a aucune garantie que la sortie du système soit dans l'enveloppe calculée.

2.3.3 Modèle Ensembliste

L'approche ensembliste permet d'intégrer l'incertitude paramétrique dans un modèle mathématique. Dans cette approche, une variable incertaine² est définie par un ensemble des valeurs possibles et cet ensemble peut prendre deux formes différentes :

- La première consiste à encadrer l'ensemble des valeurs possibles des variables par un domaine ayant une forme déterminée, comme par exemple des boîtes, des ellipsoïdes ou des parallélotopes.
- La seconde consiste à encadrer, de façon extérieure ou intérieure, le domaine réel de ces variables par une union de domaines simples.

Le domaine qui caractérise les valeurs possibles d'une variable incertaine s'appelle le *support* de la variable. Dans un modèle mathématique, une variable incertaine peut être un paramètre du modèle, l'état initial du système ou des entrées et des sorties.

Approche ensembliste et analyse par intervalle

Dans l'analyse par intervalle, les valeurs possibles d'une variable incertaine sont caractérisées par une borne supérieure et une borne inférieure³.

Historiquement, Archimède était une des premières personnes qui a utilisé des bornes bilatères dans son travail notamment pour calculer la valeur de π . La méthode utilisée par Archimède a été expliquée dans "The works of Archimedes" [Heath, 1897]. Depuis, on voit plusieurs fois les traces de l'utilisation des variables bornées en science.

Au 20ème siècle, le concept d'une fonction à variables bornées a été introduit par W. H. Young dans [Young, 1908] et ensuite, des opérations arithmétiques sur les variables multi-valeurs sont développées par [Young, 1931]. L'aspect des intervalles fermés a été proposé par P. S. Dwyer dans [Dwyer, 1951] et finalement, l'analyse par intervalle dans la forme qu'on connaît aujourd'hui a été systématisée et développée dans [Warmus, 1956], [Warmus, 1961], [Moore, 1962] et [Moore, 1966].

L'analyse par intervalle a été originalement utilisée par R. E. Moore [Moore, 1962], [Moore, 1966] dans le domaine de l'informatique pour gérer les erreurs de calcul et d'arrondis. Dans ce domaine et suite aux travaux de R. E. Moore, [Alefeld & Herzberger, 1983], [Neumaier, 1990], [Herzberger, 1994], [Figueiredo & Stolfi, 1997], [Kreinovich et al., 1998], [Kraemer & Gudenberg, 2001], et [Neumaier, 2001] ont continué de développer cet outil.

²Une variable incertaine est une variable dont la valeur est mal connue.

³On va présenter l'outil à intervalle et l'analyse par intervalle dans le chapitre suivant.

En dehors du domaine de l'informatique dans lequel l'analyse par intervalle a été originalement utilisé, celui-ci a trouvé sa place dans d'autres domaines scientifiques. Un de ces domaines avec une importance majeure en science est l'optimisation globale. Les méthodes d'optimisation classiques souffrent de deux inconvénients majeurs :

- en utilisant les algorithmes d'optimisation classiques, il est toujours possible qu'au lieu de l'optimum global, on trouve un optimum local.
- dans le cas où il existe plusieurs optimums, les méthodes classiques ne sont capables de trouver qu'un seul optimum à la fois.

Néanmoins, les algorithmes basés sur l'analyse par intervalle sont capable de trouver l'ensemble des optimums globaux. Une condition nécessaire est que l'extension intervalle de la fonction à optimiser soit *monotone par inclusion*⁴. Dans ce domaine, les ouvrages de [Ratschek & Rokne, 1988], [Kearfott, 1996], [Ratz, 1998], [Hansen & Walster, 2004] et [Fiedler et al., 2006] sont des travaux remarquables. Le problème d'optimisation globale en grande échelle est un problème interdisciplinaire sur lequel l'analyse par intervalle peut avoir un impact majeur. Dans ce cadre, le projet COCONUT [Bliet et al., 2001] a été développé.

En dehors des domaines indiqués ci-dessus, l'analyse par intervalle est potentiellement capable d'avoir une influence positive dans les autres domaines. A. Neumaier a présenté les challenges les plus importants pour lesquels l'analyse par intervalle peut apporter un plus [Neumaier, 2002].

Approche ensembliste en ingénierie et en automatique

L'ouvrage de [Kolev, 1993] peut être cité comme un des premiers ouvrages qui a utilisé l'analyse par l'intervalle pour étudier des problèmes en ingénierie. En automatique, les travaux sont classifiés en plusieurs catégories correspondant à des problèmes classiques étudiés en automatique. Une sélection des travaux effectués dans chaque catégorie est présentée dans ce qui suit :

- En identification et estimation des paramètres, on peut citer les ouvrages de [Milanese et al., 1996], [Walter & Pronzato, 1997] et [Jaulin et al., 2001a].
- [Hadj-Sadok & Gouze, 1998], [Valdes-Gonzalez, 2002], [Combastel, 2003], [Valdes-Gonzalez et al., 2003], [Flaus & Adrot, 2003], [Hanebeck et al., 2003], , [Combastel, 2005], [Raissi et al., 2006] sont des travaux sur l'estimation d'état pour les systèmes incertains.
- Le problème de diagnostic et de détection de défauts est étudié dans [Ploix, 1998], [Armengol, 1999], [Adrot, 2000], [Armengol et al., 2003], [Armengol et al., 2004], [Adrot & Flaus, 2004].

⁴ Dans le chapitre suivant, on définira la propriété de monotonie par inclusion de l'extension intervalle d'une fonction réelle.

- Une propriété importante d'un modèle est la stabilité de celui-ci. Dans [Henrion, 2001], [Petras, Chen, Vinagre & Podlubny, 2003] et [Petras, Chen & Vinagre, 2003], ce problème a été étudié pour des modèles incertains avec des paramètres bornés.
- Une autre branche en automatique est la commande robuste. Les travaux de [Jaulin et al., 2001a], [Rauh et al., 2005], [Rauh et al., 2005], [Rauh & Hofer, 2005] et [Hanebeck et al., 2003] et [Henrion, 2001] sont sur ce sujet.

Identification de modèles intervalle

Pour obtenir un modèle mathématique à partir d'un système existant, on peut utiliser deux méthodes dépendant de nos connaissances sur le système.

- Approche *modèle de connaissance* ou *boite blanche*.
- Approche *modèle de comportement* ou *boite noire*.

Dans l'approche *modèle de connaissance*, la première étape est d'étudier la structure du système, de trouver les variables caractéristiques du système et de déterminer les lois physiques qui lient les variables. Les relations entre les variables caractéristiques sont généralement des équations différentielles ou des équations aux différences obtenues en procédant à des bilans. Cela détermine la structure du modèle. Dans l'étape suivante, les paramètres du modèle sont calculés à partir de la connaissance à priori des paramètres ou à partir des observations sur l'entrée et la sortie du système et en appliquant des techniques d'identification. La dernière étape de cette approche est de valider le modèle pour l'application visée. Parmi les ouvrages sur cette approche, on peut citer [Wellstead, 1979] et [Nicholson, 1981]. Dans [Close et al., 2001], ce problème a été traité pour des applications en ingénierie.

Dans la seconde approche, on ne cherche pas à déterminer très précisément les phénomènes physiques qui se passent à l'intérieur du système. De ce fait, il faut choisir une structure mathématique liant les entrées et les sorties du système. Cette structure peut être dynamique ou statique, linéaire ou non linéaire, continue ou discrète et elle est sélectionnée parmi un ensemble de structures candidates. Après avoir choisi la structure convenable, les paramètres du modèle sont calculés à partir des entrées et des sorties observées en optimisant un critère d'optimisation et en dernière étape, le modèle est validé pour l'application visée.

Sur la figure 2.2, les étapes d'identification de systèmes par l'approche modèle de connaissance et l'approche modèle de comportement ont été illustrées. La boucle de l'identification de système proposée dans [Ljung, 1999] a été illustrée sur la figure 2.3.

Dans ce document, on utilisera l'approche modèle de comportement pour identifier le système. On utilisera également les intervalles pour caractériser l'incertitude et par conséquent, le modèle sera un modèle ensembliste par intervalle ou plus brièvement, un *modèle intervalle*. On suppose aussi que le jeu de données a déjà été obtenu par

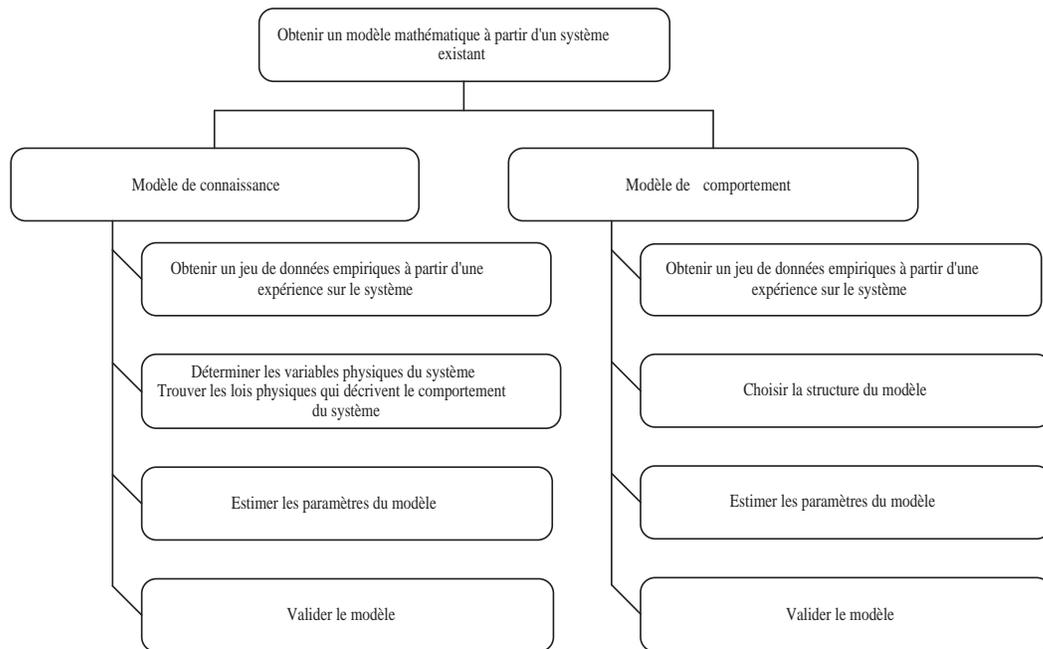


FIG. 2.2 – Approche modèle de connaissance et approche modèle de comportement.

expérience sur le système. Donc, par rapport au schéma de la figure 2.3, il faut choisir la structure du modèle (identification de structure), identifier les paramètres du modèle (identification de paramètres) et valider le modèle pour l'application visée.

Choix de la structure. Peu de travaux ont déjà été effectués sur le choix de la structure des modèles intervalles. Souvent, la structure du modèle a été pré-définie sur les mêmes critères qu'un modèle déterministe. La structure d'un modèle intervalle peut être choisie parmi l'ensemble des modèles classiques *ARX*, *ARMAX*, *OE* etc.

Identification des paramètres. Dans un modèle intervalle, avant de calculer les paramètres du modèle, il faut définir la sémantique d'identification qui détermine les propriétés des supports des paramètres du modèle ; c'est-à-dire ce que représentent ces intervalles. L'identification d'un modèle intervalle consiste à :

Définir la sémantique d'identification. La sémantique d'identification est définie par un formalisme mathématique qui détermine ce que représentent les supports des paramètres du modèle. Dans le cas d'un modèle intervalle, il est composé des quantificateurs \forall , \exists , \neg avec les paramètres et les variables du système. Cette sémantique est définie par rapport à l'objectif d'identification et l'application visée pour le modèle.

Calculer les supports des paramètres du modèle. Pour cela, on a besoin d'une approche numérique. Il existe plusieurs approches numériques pour caractériser les bornes des paramètres d'un modèle intervalle. [Armengol, 1999] a proposé d'utiliser des intervalles modaux. Dans le travail de O. Adrot [Adrot

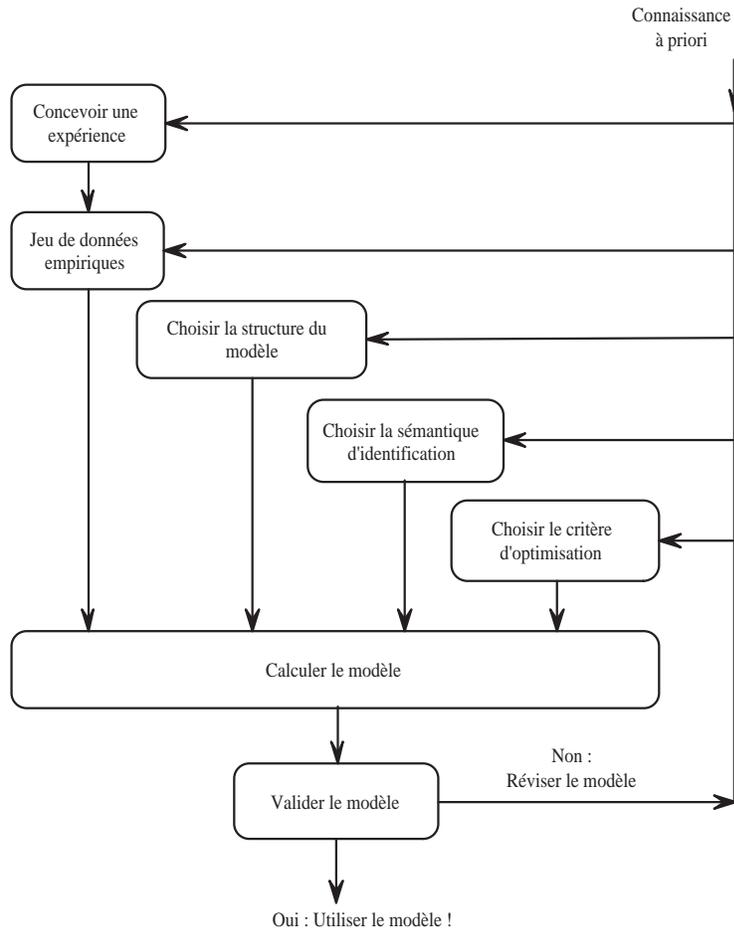


FIG. 2.3 – Boucle de l'identification de système.

et al., 2004] et le travail de S. Ploix [Ploix et al., 1999], ce problème a été transformé comme un problème d'optimisation sous un ensemble des contraintes à respecter. Parallèlement, certains travaux ont utilisé la méthode de l'inversion ensembliste [Milanese et al., 1996]. Le choix de l'approche numérique dépend de la sémantique d'identification.

Un exemple simple de différentes sémantiques d'identification d'un modèle intervalle est donné dans ce qui suit. Soit $\{x_k, [y]_k\}$ ⁵ un ensemble d'observations sur l'entrée scalaire x_k et la sortie intervalle scalaire $[y]_k$ obtenues à partir d'une expérience sur le système. On veut expliquer ces observations par un modèle linéaire à variables bornées (modèle intervalle) ayant la forme suivante :

$$y_k = a \times x_k + b \text{ pour } a \in [a], b \in [b]$$

⁵Dans ce document, x est un scalaire avec une valeur certaine (respect. un vecteur avec des valeurs certaines) et $[x]$ est un intervalle (respect. un vecteur intervalle).

On peut calculer les supports $[a]$ et $[b]$ tels que :

$$\forall k, \forall y_k \in [y_k], \exists a \in [a], \exists b \in [b] \mid y_k = a \times x_k + b$$

Par cette sémantique, on obtient un modèle intervalle pour lequel à tous les instants, la sortie observée est expliquée par au moins une valeur contenue dans $[a]$ et dans $[b]$. La sortie observée et la sortie calculée par le modèle sont montrées sur la figure 2.4. L'interprétation géométrique de ce domaine est détaillée dans l'annexe B. On peut aussi les calculer tels que :

$$\forall k, \forall a \in [a], \forall b \in [b], \exists y_k \in [y_k], \mid y_k = a \times x_k + b$$

Cette sémantique détermine toutes les valeurs faisables de a et de b qui expliquent au moins un point dans l'intervalle de la sortie observée $[y]_k$. Autrement dit, la trajectoire du modèle, pour toutes les valeurs contenues dans $[a]$ et dans $[b]$ passe toujours à l'intérieur du support de la sortie observée. Cette sémantique est ainsi appelée le *FPS*⁶. La sortie observée et la sortie calculée par le modèle sont montrées sur la figure 2.5. Par ces deux sémantiques, on obtient deux modèles intervalle différents. Dans [Calm et al., 2006] une méthode systématique a été proposée pour caractériser les paramètres d'un modèle intervalle en utilisant différents choix de quantificateurs.

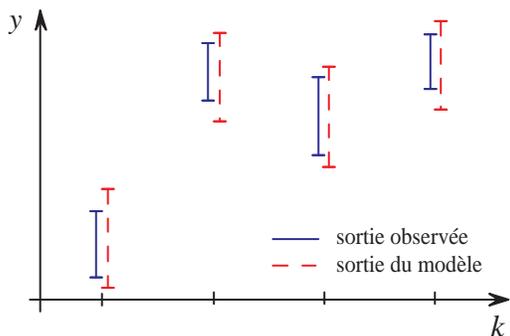


FIG. 2.4 – Première sémantique.

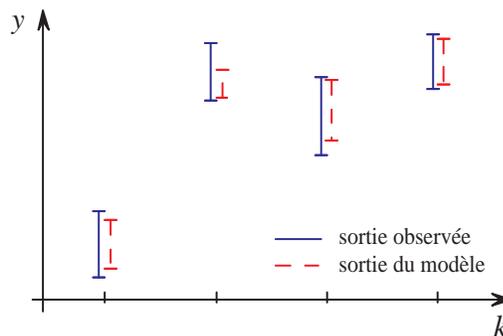


FIG. 2.5 – Deuxième sémantique.

2.3.4 Conclusion

Dans cette section, on a étudié brièvement les modèles déterministes, probabilistes et ensemblistes et on a vu comment l'erreur de modélisation est gérée dans ceux-ci.

⁶Feasible Parameter Set

On a expliqué que dans un modèle déterministe, la sortie du modèle est une valeur certaine et elle n'apporte aucune information sur l'erreur de modélisation. En revanche, dans un modèle probabiliste, l'erreur de modélisation est caractérisée par une loi de distribution. Cependant, il est souvent difficile de trouver une loi pour cette distribution et de respecter cette loi pendant l'évaluation du modèle.

L'approche ensembliste permet de prendre en compte l'incertitude paramétrique d'un modèle mathématique. L'approche ensembliste nous permet aussi d'analyser le comportement du système pour un ensemble de valeurs de l'état initial ou de l'entrée du système. Dans un modèle intervalle, qui est un cas particulier des modèles ensemblistes, les variables incertaines sont caractérisées par des intervalles et si les supports de ces variables sont bien choisis, on garantit que la sortie du système est incluse dans l'enveloppe de la sortie. Par conséquent, l'approche ensembliste apparaît être une approche adaptée pour une analyse de sûreté. Bien entendu, avant d'utiliser ce modèle, il est nécessaire de le construire, et nous verrons donc comment en identifier les paramètres (ou plus exactement, leurs supports) dans le chapitre suivant.

2.4 Simulation des Modèles

Par le terme *simulation* d'un système, on veut dire calculer les séquences futures des états du système [Berleant & Kuipers, 1992], [Bonarini & Bontempi, 1994]. Pour cela, en plus du modèle qui représente le système, il nous faut un simulateur. Le simulateur évalue le modèle et produit une estimation du comportement du système. Il existe trois types de simulateurs par rapport à la nature des modèles qui vont être détaillés dans ce qui suit.

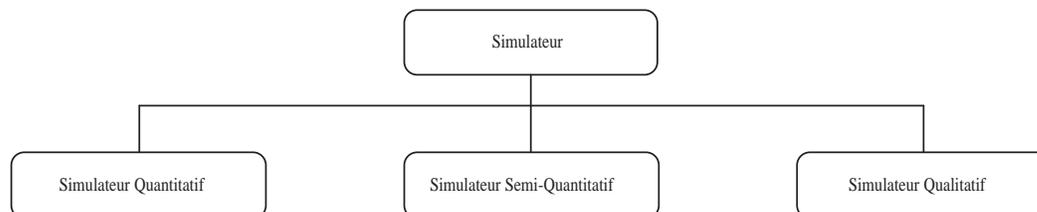


FIG. 2.6 – Différents types de simulateurs

2.4.1 Simulateur Quantitatif

C'est un simulateur qui est utilisé surtout pour les modèles mathématiques. Si le modèle est un modèle déterministe ou probabiliste dans lequel les paramètres ou/et les entrées sont certains, la sortie du modèle est calculée directement en fonction de ses paramètres et de ses entrées. Ce calcul est purement numérique et cette classe des simulateurs s'appelle les *simulateurs quantitatifs*. L'ouvrage de [Cellier & Kofman, 2006] explique comment les modèles des systèmes dynamiques décrits généralement par des

équations différentielles ordinaires ou partielles sont simulés sur une plateforme informatique.

2.4.2 Simulateur Qualitatif

Dans la littérature, il existe un autre type de simulateurs s'appelant les *simulateurs qualitatifs*. Ils sont utilisés pour simuler les modèles qualitatifs. Ce type de modèles est basé sur le raisonnement logique et ne sera pas traité dans ce document. Pour plus d'informations, il est possible de consulter [Kay, 1996].

2.4.3 Simulateur Semi-Quantitatif

Les simulateurs *semi-quantitatifs* se situent à la frontière entre les simulateurs quantitatifs qui sont utilisés pour simuler les modèles mathématiques et les simulateurs qualitatifs qui sont utilisés pour simuler les modèles qualitatifs.

Le simulateur d'un modèle intervalle est semi-quantitatif, puisque pour simuler ce modèle, on a besoin d'une part d'un modèle mathématique, mais aussi d'une sémantique de simulation qui est un formalisme mathématique composé des quantificateurs sur des paramètres et des variables du système.

2.4.4 Simulateur d'un modèle intervalle

Dans les parties précédentes, on a présenté brièvement une classification des simulateurs par rapport au type de modèle utilisé. Dans cette partie, on se concentre sur les simulateurs des modèles ensemblistes et plus précisément des modèles intervalle. Un modèle intervalle est un modèle incertain et on utilise un simulateur semi-quantitatif pour évaluer ce type de modèles.

Dans un modèle intervalle, la sortie incertaine du modèle est généralement caractérisée par une enveloppe.

Définition 2.4.1. *Au sens de la géométrie, une enveloppe est définie comme une courbe ou une surface qui est tangente à tous les membres d'un système ou des courbes ou des surfaces.*

Les courbes dans ce cas, sont des trajectoires possibles du système qui ont été reproduites par le modèle. L'ensemble des trajectoires du système varie en fonction de la façon dont l'incertitude de modèle est prise en compte. La façon dont l'incertitude du modèle est prise en compte, est définie par un formalisme mathématique qui s'appelle la *sémantique de simulation*. Par conséquent, l'enveloppe de l'ensemble de trajectoires du système dépende de la sémantique de simulation. La sémantique de simulation est déterminée par l'objectif de simulation. Pour calculer l'enveloppe qui est définie par la

sémantique de simulation, on choisit une méthode numérique. L'ensemble de la sémantique de simulation ainsi que la méthode numérique pour calculer l'enveloppe sont deux éléments qui définissent le simulateur semi-quantitatif. Sur la figure 2.7, la structure d'un simulateur pour un modèle à intervalle a été illustrée.

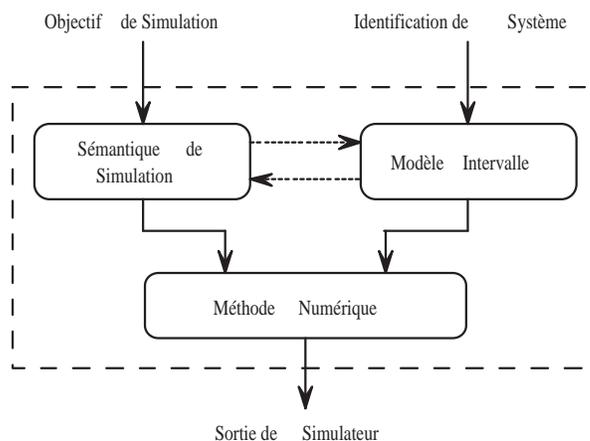


FIG. 2.7 – Structure d'un simulateur semi-quantitatif pour un modèle à intervalle.

Sémantique de simulation

La sémantique de simulation dépend de l'objectif de simulation et est une composition des quantificateurs \forall , \exists , \neg sur les paramètres et les variables du modèle. Elle détermine ce que représente le modèle en terme de comportement compte tenu des incertitudes considérées. Pour mieux montrer les différentes sémantiques de simulation, on utilise l'exemple suivant. Soit f , un modèle à intervalle de la forme suivante :

$$f : \hat{y}_k = a \times x_k + b, \quad a \in [a], \quad b \in [b]$$

où x_k est la variable physique et a et b sont les paramètres bornés du modèle caractérisés respectivement par les intervalles $[a]$ et $[b]$. Différentes sémantiques de simulation peuvent être définies pour ce modèle et chaque sémantique détermine un ensemble d'enveloppes. On donne quelques exemples. Soit $[env]_k$, le domaine qui définit par la sémantique de simulation à l'instant k . Ce domaine est également caractérisé par un intervalle et donc, on aura :

- $[env]_k = \{\hat{y}_k = a \times x_k + b \mid \forall a \in [a], \forall b \in [b]\}$. L'enveloppe $\{[env]\} = \{[env]_1, [env]_2, \dots\}$ correspond aux trajectoires du système pour toutes les valeurs possibles de a et b .
- $[env]_k = \{\hat{y}_k = a \times x_k + b \mid \forall a, b = 0\}$. L'enveloppe $\{[env]\} = \{[env]_1, [env]_2, \dots\}$ correspond aux trajectoires du système pour toutes les valeurs possibles de la

variable a en éliminant l'effet de la variable b sur la sortie estimée. Le choix $b = 0$ peut être interprété comme négliger le terme additif du modèle dans le calcul de la sortie du système.

Pour une étude plus approfondie sur la sémantique de simulation des modèles ensemblistes, on vous renvoie aux travaux de [Armengol et al., 2001], [Armengol, 1999] et [Calm et al., 2006].

Approche numérique pour calculer l'enveloppe

La deuxième étape pour évaluer un modèle intervalle est de trouver une approche numérique pour calculer $\{\{env\}\}$, l'enveloppe des trajectoires du système qui est définie par la sémantique de simulation. Avant de présenter les méthodes numériques existantes, on définit les notions d'approximations extérieure et intérieure de $\{\{env\}\}$.

Définition 2.4.2. Une approximation extérieure $\overline{\{\{env\}\}}$ est une enveloppe qui contient l'enveloppe $\{\{env\}\}$. C'est à dire : $\{\{env\}\} \subseteq \overline{\{\{env\}\}}$

Définition 2.4.3. Une approximation intérieure $\underline{\{\{env\}\}}$ est une enveloppe qui est contenue dans l'enveloppe $\{\{env\}\}$. C'est à dire : $\underline{\{\{env\}\}} \subseteq \{\{env\}\}$

Par cette définition, on constate que les approximations intérieure et extérieure de l'enveloppe $\{\{env\}\}$ ne sont pas uniques. Par la suite, on présentera les méthodes existantes pour calculer l'enveloppe $\{\{env\}\}$.

Calcul exact de l'enveloppe. Parfois, il est facile de déterminer l'enveloppe $\{\{env\}\}$ par les expressions mathématiques des bornes supérieure et inférieure de l'enveloppe. C'est le cas de l'exemple au dessus et plus généralement, dans ce travail puisqu'on utilise des modèles ensemblistes linéaires par rapport aux paramètres.

Ensemble de modèles précis. S'il n'est pas possible de déterminer les bornes de l'enveloppe par des expressions mathématiques, les approches numériques sont envisagées. L'une d'entre elles est d'étudier un grand nombre de modèles déterministes dont les paramètres appartiennent aux supports des paramètres du modèle intervalle. La stratégie pour le choix des modèles dépend de la sémantique de simulation. Ils peuvent être choisis systématiquement par le maillage de l'espace des paramètres ou aléatoirement. La méthode de Monte-Carlo est bien connue dans ce contexte. L'enveloppe calculée par cette approche est très souvent une approximation intérieure de l'enveloppe qui est définie par la sémantique de simulation, puisqu'il peut exister une trajectoire telle que ce soit une trajectoire possible du système et, qu'elle ne soit pas dans l'ensemble de trajectoires des modèles déterministes. Plus le nombre des modèles déterministes est élevé, meilleure est l'enveloppe $\{\{env\}\}$.

Problème d'optimisation Si les bornes de l'enveloppe $\{[env]\}$ ne peuvent pas être déterminées facilement par une expression mathématique, on utilise la deuxième méthode pour approcher l'enveloppe par une approximation. Dans certains cas, cette solution ne nous convient pas dans la mesure où on ne peut plus garantir nos résultats. Ceci est critique dans le cas de l'analyse de sûreté. Dans le cas de l'exemple des bacs sur la figure 1.1 en introduction de ce travail, on ne pourra certifier qu'il n'y aura pas un débordement du liquide dans l'un des bacs puisque toutes les valeurs possibles de la sortie du système ne sont pas calculées. C'est le cas surtout dans les contextes comme le contexte de la sûreté de fonctionnement. De ce fait, l'enveloppe $\{[env]\}$ est approchée par une approximation intérieure et une approximation extérieure. Il faut se méfier et savoir qu'il existe un nombre infini d'approximations intérieures et extérieures pour l'enveloppe ; c'est-à-dire n'importe quelle enveloppe qui contient (respect. qui est contenue dans) l'enveloppe $\{[env]\}$ est une approximation extérieure (respect. intérieure) de celle-ci. Dans plupart des travaux, le critère d'optimisation est la différence entre ces deux approximations et le problème de calculer l'enveloppe $\{[env]\}$ revient à un problème d'optimisation où il faut réduire l'espace frontière entre les deux approximations. Dans ce contexte, [Armengol, 1999] a utilisé des intervalles modaux et [Kletting et al., 2005] a utilisé l'inversion ensembliste pour calculer ces deux approximations.

2.4.5 Conclusion

Dans cette section, on a présenté trois catégories de simulateurs pour différents types de modèles en fonction de leurs natures. Les simulateurs qualitatifs pour les modèles qualitatifs, les simulateurs quantitatifs pour les modèles déterministes et probabilistes et les simulateurs semi-quantitatifs qui peuvent être utilisés pour les modèles incertains.

Les modèles ensemblistes se situent parmi les modèles mathématiques incertains. De ce fait, on a proposé d'utiliser un simulateur semi-quantitatif pour simuler un système qui est représenté par un modèle ensembliste. On a aussi vu qu'un simulateur semi-quantitatif consiste en deux parties. Une partie est la sémantique de simulation qui est définie par l'objectif de simulation et la deuxième partie est la méthode numérique pour calculer l'enveloppe qui est définie par la sémantique de simulation. Trois approches numériques ont été aussi rappelées pour calculer cette enveloppe.

2.5 Modélisation Non-Linéaire

Pour reproduire l'évolution d'un système dans le temps et analyser son comportement, on a besoin d'un modèle mathématique du système. Ce modèle définit les liens entre les variables d'état, les entrées et les sorties du système. Ces liens sont représentés par des équations mathématiques et s'ils sont linéaires, le système est représenté par un modèle linéaire. Cependant pour un système non-linéaire, ces liens ne sont plus linéaires et les variables du système sont reliées par des équations non-linéaires. Dans ce cas et pour représenter le comportement non-linéaire du système, les modèles non-linéaires sont

utilisés. Pour cela, on peut également utiliser des approches alternatives comme le réseau de neurones, le modèle multi-expert, le modèle flou et le modèle multi-mode.

2.5.1 Modèles Non-Linéaires

Dans les méthodes d'identification classiques, on passe par trois étapes. A la première étape, on essaie de trouver une structure convenable pour le modèle -identification de structure- et à la deuxième étape, on optimise les paramètres du modèle -identification de paramètres-. Le modèle est ensuite validé pour l'application visée.

Il existe un nombre important de modèles non-linéaires dont le modèle *Volterra-Wiener*, qui est l'un des plus anciens. Cependant, à cause du nombre important de paramètres, ce modèle ne présente pas beaucoup d'intérêt en pratique. Les modèles *Hammerstein* et *Wiener* sont deux autres structures possibles pour des systèmes non-linéaires. Ils sont constitués de deux blocs distincts en parallèle, un bloc non-linéaire statique (respect. linéaire dynamique) auquel on ajoute un autre bloc linéaire dynamique (respect. non linéaire statique). Cependant, il existe peu de techniques d'identification pour ces modèles. Une autre possibilité est le modèle *NARMAX* qui est une extension du modèle linéaire *ARMAX* [Johansen & Foss, 1993].

Les modèles non-linéaires ne sont pas limités aux modèles présentés ci-dessus. Pourtant, l'identification de ceux-ci souffre de problèmes communs. Pour un système avec un comportement complexe, trouver une structure mathématique qui détermine les liens entre les variables et qui représente le comportement du système est parfois difficile à réaliser. Plus le système est complexe, plus ce choix est difficile. En plus, pour une structure complexe, l'identification paramétrique conduit très souvent à résoudre un problème d'optimisation difficile à résoudre. Ces deux problèmes deviennent encore plus difficiles à résoudre pour les modèles incertains.

De ce fait, pour modéliser un système avec un comportement complexe et non-linéaire, au lieu d'utiliser un modèle non-linéaire, on préfère utiliser des approches alternatives développées dans les sections suivantes.

2.5.2 Réseaux de Neurones

Les réseaux de neurones étaient utilisés initialement pour modéliser le fonctionnement des neurones biologiques. L'utilisation de ces modèles pour représenter les systèmes complexes et non-linéaires a connu un grand succès avec de nombreuses applications dans des domaines divers.

La structure des réseaux de neurones est une structure simple et générale qui peut être utilisée pour des comportements non-linéaires indépendamment du type de non-linéarité. Un réseau de neurones consiste en plusieurs couches ; chaque couche contenant un certain nombre de neurones. L'entrée d'un neurone est la somme pondérée des sorties des neurones de la couche précédente et sa sortie est calculée par la fonction d'activation du neurone. L'identification de la structure d'un réseau consiste à déterminer le nombre

de couches, le nombre de neurones dans chaque couche et la fonction de combinaison de neurones. Pour identifier les paramètres du réseau, la méthode la plus courante est la méthode de rétropropagation qui est une extension de la méthode des moindres carrés. Pour une étude plus approfondie sur l'identification de systèmes par les réseaux de neurones, on vous renvoie à l'ouvrage [Norgaard et al., 2006].

2.5.3 Modèles Multi-Experts

Une autre approche pour modéliser un système non-linéaire est d'utiliser un modèle multi-expert. Cette approche est basée sur l'idée que les données sont générées par un ensemble d'experts. Un expert est une fonction avec un vecteur de paramètres et un vecteur d'entrées. Le vecteur d'entrées est traité simultanément par tous les modules experts et chaque expert fournit une sortie. Le champ d'action d'un expert est déterminé par une fonction d'activation et cette fonction d'activation pondère la contribution de chaque expert dans la sortie de modèle. Cette approche est une interprétation probabiliste, puisque la fonction d'activation s'interprète comme la probabilité que la sortie du modèle relève de l'expert associé [Jacobs & Jordan, 1991].

2.5.4 Modèles Flous

La logique floue a été originalement utilisée pour caractériser l'incertitude linguistique et cela se faisait à l'aide de fonctions d'appartenance. Ensuite et dans le contexte de modélisation, la logique floue a été utilisée pour intégrer les savoir faire des experts dans les modèles mathématiques par des règles "si *prémisse* alors *conséquence*". De même, elle peut être utilisée pour modéliser le comportement complexe et non-linéaire d'un système. Un exemple de ce type de modèles est le modèle Takagi-Sugeno [Takagi & Sugeno, 1985].

2.5.5 Modèles Hybrides : Approche Multi-Mode

Le terme hybride signifie le mélange de deux types de méthodes ou de deux objets fondamentalement différents. Les réseaux de neurones hybrides correspondent au cas où des réseaux de neurones artificiels et la logique floue sont combinés. La commande numérique d'un système continu peut aussi être considérée comme un système hybride puisqu'il combine les systèmes à temps continu avec des systèmes à temps discret [Lemmon et al., 1999]. Les modèles hybrides sont utilisés dans trois disciplines majeures en science appliquées : la théorie de la commande des systèmes continus⁷, la théorie des systèmes à événements discrets⁸ et l'informatique⁹. Une étude sur ce thème a été faite dans [Flaus, 1998]. Les modèles multi-modes sont un sous-ensemble des modèles hybrides. Ils sont une alternative pour représenter un système avec un comportement complexe et non-linéaire.

⁷Continuous Control Systems Theory

⁸Discrete-Event System Theory

⁹Computer Science

L'approche multi-mode apparaît intéressante à exploiter puisque par cette approche, on représente le comportement complexe d'un système par un modèle multi-mode qui consiste en un ensemble de modèles locaux plus simples où chaque modèle local explique le comportement du système dans une zone donnée. Cela nous permet d'évaluer ce modèle plus facilement en généralisant les techniques d'identification et d'analyse qui existent pour les modèles linéaires. L'approche multi-mode sera utilisée dans la suite dans ce document au chapitre quatre pour prendre en compte l'incertitude structurelle du modèle et améliorer la précision de la modélisation.

Ensemble des travaux effectués

Dans cette partie, on va présenter une sélection des travaux sur les modèles multi-mode en automatique. Il faut mentionner que le nombre très important des travaux dans ce contexte nous a obligé à faire une sélection.

Le travail de [Takagi & Sugeno, 1985] est l'un des premiers travaux sur l'identification d'un système multi-mode. Ce travail se situe dans le contexte de la commande. L'espace d'entrée est partitionné en ensembles flous et chaque ensemble est modélisé par un modèle linéaire ou non-linéaire. Le modèle multi-mode est l'interpolation des modèles locaux. Cette interpolation est faite par un ensemble de règles "si *prémisse* alors *conséquence*". Ce modèle s'appelle le modèle *Tagaki-Sugeno (TS)*. Dans ce contexte, on peut aussi mentionner les modèles *Tagaki-Sugeno-Kang (TSK)* [Sugeno & Kang, 1988], les modèles *flous quasi linéaires (QLFM)*¹⁰ [Yager & Filev, 1993] et les modèles *flous quasi non-linéaires (QNFM)*¹¹ [Filev, 1991].

Le travail de [Johansen & Foss, 1992] se situe dans le contexte de la commande adaptative. Dans ce travail, un algorithme pour estimer et optimiser les paramètres des modèles locaux à partir des observations a été présenté. La structure des modèles locaux est pré-définie. Cette structure peut être une série de Taylor, une représentation d'état ou une structure non-linéaire. Les zones de fonctionnement ainsi que les fonctions de validité des modèles locaux sont aussi pré-définies. La sortie du modèle multi-mode est la somme pondérée des sorties des modèles locaux.

Dans la continuité du travail précédent, le travail de [Johansen & Foss, 1995] se situe dans le contexte de l'identification. Il a présenté un algorithme heuristique où le découpage du régime opératoire du système, l'identification de la structure et l'identification des paramètres se font en parallèle. L'espace de fonctionnement est un sous-ensemble de l'espace d'entrée. La sortie du modèle est la somme pondérée des sorties des modèles locaux. Dans [Johansen & Foss, 1997], le même problème a été traité. Néanmoins, les auteurs ont utilisé des informations a priori pour déterminer les variables caractéristiques et les zones de fonctionnement des modèles locaux.

Le travail de [Boukhris et al., 2001] se situe dans le contexte de la détection de défauts. Le système étudié est un système non-linéaire et variant dans le temps. Un modèle multi-mode TS¹² a été utilisé pour modéliser le système. Les modèles locaux sont des relations

¹⁰Quasi Linear Fuzzy Model

¹¹Quasi Non-Linear Fuzzy Model

¹²Tagaki-Sugeno

entrée-sortie. Puisque le système est un système mono-entrée/mono-sortie, le problème du choix des variables caractéristiques ne s'est pas posé.

Dans leur travail, A. N. Venkat et al. ont proposé une approche méthodologique pour décomposer le comportement non-linéaire du système étudié [Venkat et al., 2003]. Cette décomposition a été faite dans l'espace de fonctionnement en régime permanent¹³. L'espace d'entrée est divisé en zones locales. Le nombre optimal et nécessaire des partitions est déterminé par une méthode basée sur la validation du modèle par des mesures. La structure des modèles locaux peut être linéaire comme les structures *ARX* et *ARMAX* ou non-linéaire comme *NARX*. Pour calculer la sortie du modèle multi-mode, les modèles locaux sont interpolés par une agrégation floue. La méthodologie proposée est appliquée à la commande d'un procédé chimique non-linéaire.

Dans sa thèse, K. Gasso a traité le problème de la modélisation multi-mode [Gasso, 2000]. Il a proposé une méthodologie pour choisir le degré de liberté du modèle, le régresseur, la structure du modèle et l'optimisation des paramètres du modèle. Il a aussi présenté une méthode pour décomposer l'espace d'état du système en zones de fonctionnement local. Un compromis entre le nombre des modèles et la précision de la modélisation doit être fait. Après avoir décomposé l'espace d'état, il a étudié la possibilité d'agréger des modèles locaux. Cette approche a été appliquée sur le problème d'estimation des variations de la concentration d'ozone dans l'air.

Dans le contexte de la commande des systèmes non-linéaires, une comparaison a été faite entre trois différents type de *contrôleur PID à séquence de gains*¹⁴ [Ling & Edgar, 1997]. Un contrôleur *PID* à séquence de gains contient un ensemble de contrôleurs linéaires ayant une structure identique. Au moment de la commutation entre les contrôleurs, les valeurs de paramètres du contrôleur *PID* à séquence de gains conventionnel¹⁵ changent. Ce changement est brusque et il n'est pas désirable. Dans le contrôleur *PID* à séquence de gains flou¹⁶, ce changement se fait à l'aide de la fonction d'appartenance. Pour mieux approcher la dynamique du système, le contrôleur *PID* à séquence de gains flou basé sur le modèle¹⁷ a été proposé. Ce travail ne traite pas de l'identification multi-mode et il s'est concentré plutôt sur l'application d'un contrôleur multi-mode.

Un certain nombre de phénomènes physiques ont des comportements hybrides. Par exemple, l'évolution de la température pendant la fonte de la glace consiste en trois phases : la phase solide où la température évolue (augmente) jusqu'à ce qu'elle arrive au point de fusion, la phase de fusion (mélange de glace et d'eau) où la température reste constante et égale au point de fusion et la phase liquide où la température évolue à nouveau. Dans le travail de [Boillereaux et al., 2002], ce phénomène a été représenté par un modèle multi-mode dans le contexte de la modélisation et de l'analyse de procédés de décongélation. Dans ce travail, la pression est considérée variante et par conséquent, le point de fusion varie en fonction de la pression. Une étude d'observabilité et de commandabilité a été aussi réalisée.

¹³Steady State Space

¹⁴Gain Scheduling PID Controllers

¹⁵Conventional PID Gain Scheduling Controller

¹⁶Standard PID Fuzzy Gain Scheduling

¹⁷Model-Based PID Fuzzy Gain Scheduling

Dans le travail de [Balluchi et al., 2001], les auteurs proposent une méthodologie pour concevoir un observateur hybride. Cet observateur consiste en deux parties : une première partie qui détermine le mode actif du système à chaque instant et une deuxième partie qui utilise l'information provenant de la partie précédente, l'entrée et la sortie du système pour estimer l'état du système. Cette méthodologie est ensuite appliquée à la synthèse d'un modèle de transmission de puissance dans un véhicule. Cette problématique a été plus détaillée dans [Balluchi et al., 2002]. Ils proposent aussi une solution pour déterminer le mode actif dans un modèle multi-mode. Dans ce contexte, on peut aussi citer [Ragot et al., 2003], [Babaali & Egerstedt, 2005].

Dans le contexte de diagnostic des systèmes multi-mode, [Domlan, 2006] a traité deux problèmes. Le premier problème est de déterminer le mode de fonctionnement actif du système à un instant donné à partir des observations sur le système. Cela se fait à l'aide des outils de diagnostic à base de modèles. Après avoir déterminé le mode actif, il propose une approche pour identifier le mécanisme de gestion des transitions d'un mode de fonctionnement à un autre, ce qui consiste dans notre cas à *déterminer le domaine de validité des modèles locaux*, à l'aide de méthodes de reconnaissance de forme et de classification. Pour cela, il s'intéresse à une classe particulière de structure dans laquelle la loi de commutation du système est obtenue en partitionnant un espace associé à des grandeurs connues du système. Contrairement à la plupart des travaux dans la littérature dans lesquels la loi de commutation dépend à l'état du système, dans le travail de E.A. Domlan, les grandeurs sont restreintes à l'entrée et à la sortie du système.

Identification d'un Modèle Multi-Mode

L'identification d'un modèle multi-mode se décompose en six points :

Les variables caractéristiques sont des variables du modèle qui vont permettre de modéliser le comportement non-linéaire du système. Autrement dit, ce sont des variables qui interviennent dans les équations non-linéaires du modèle et qui vont représenter le comportement du système. Connaître ces variables nous aide à choisir une structure convenable pour les modèles locaux, déterminer l'espace de validité des modèles locaux et décomposer cet espace en plusieurs zones locales de fonctionnement.

La structure des modèles locaux est choisie parmi des structures candidates. Pour que l'identification des paramètres soit plus simple, on préfère utiliser une structure linéaire comme le modèle entrée-sortie, le modèle réponse impulsionnelle et le modèle réponse indicielle. Il est plus simple de choisir une structure identique pour tous les modèles locaux. Néanmoins, il est aussi possible d'identifier la structure de chaque modèle local individuellement.

L'espace de validité est l'espace dans lequel les domaines de validité des modèles locaux sont déterminés. Cet espace est généralement l'espace des variables caractéristiques du modèle.

La stratégie pour partitionner le régime opératoire du système ou partitionner l'espace de validité. Le domaine de validité d'un modèle local est un sous-espace de l'espace de validité. L'espace de validité doit être décomposé en plusieurs zones tels que dans chaque zone, le système puisse être représenté par un modèle local. Il existe deux méthodes pour partitionner cet espace :

- Le partitionnement se fait en utilisant la connaissance a priori sur le système.
- Dans le cas où la connaissance sur le système est limitée, le partitionnement se fait en parallèle avec l'identification des paramètres des modèles locaux.

L'espace de validité peut être décomposé en sous-espaces disjoints ou en sous-espaces avec des zones en commun. Le domaine de validité de chaque modèle local peut avoir différentes formes comme le triangle, le trapézoïde, la gaussienne ou le sigmoïde. Il peut aussi être déterminé par une forme comme une enveloppe convexe autour d'un certain nombre de points, ou un partitionnement flou de l'espace de validité.

La fonction de validité est une fonction de l'espace de validité dans R qui détermine, à un instant donné, la validité de chaque modèle local. Cette fonction est généralement utilisée pour pondérer les sorties des modèles locaux et calculer la sortie du modèle multi-mode.

La fonction d'interpolation est une fonction par laquelle la sortie d'un modèle multi-mode est calculée à partir des sorties des modèles locaux. La sortie du modèle multi-mode est généralement une somme des sorties normalisées des modèles locaux pondérée par les fonctions de validité. Bien que cette fonction soit la fonction la plus utilisée, elle n'est pas la seule possibilité et on peut définir d'autres fonctions linéaires ou non-linéaires pour calculer la sortie du modèle multi-mode.

Dans l'ouvrage [Murray-Smith & Johansen, 1997a], le problème d'identification multi-mode a été traité.

2.5.6 Conclusion

Dans cette section, différentes approches pour représenter le comportement complexe et non-linéaire d'un système ont été présentées. On a vu que l'identification de structure et de paramètres d'un système complexe et non-linéaire n'est pas facile à réaliser par les méthodes d'identification classiques. Pour cette raison, d'autres approches ont été envisagées comme le réseau de neurone, le modèle multi-expert, le modèle flou et le modèle multi-mode. Dans un modèle multi-mode, le régime opératoire du système est partitionné et est représenté par un ensemble de modèles simples.

2.6 Modèles Multi-Modes Intervalle

Un modèle intervalle explique les observations d'un système en générant un tube de trajectoires devant les contenir. Cependant, si on utilise un modèle linéaire intervalle pour représenter un système non-linéaire, l'enveloppe englobante peut être une enveloppe large¹⁸. L'imprécision de l'enveloppe englobante dans un modèle intervalle est le pendant de l'erreur de modélisation pour les modèles déterministes et probabiliste.

On a déjà vu dans la section 2.5 que pour un système non-linéaire, on peut utiliser plusieurs approches ; parmi celles-ci on a proposé d'utiliser l'approche multi-mode. Néanmoins, très peu de travaux ont déjà été faits sur l'identification et la simulation des modèles multi-mode intervalle.

Pour un modèle d'état déterministe multi-mode, [Shahriari et al., 2004] ont calculé l'ensemble des trajectoires possibles du système dont l'état initial était incertain et caractérisé par un vecteur d'intervalle. Dans ce travail, les zones de fonctionnement étaient des polytopes convexes dans l'espace d'état.

Dans [Dang, 2000], le problème de vérification a été traité pour un modèle multi-mode. Dans ce travail, le modèle du système était un modèle d'état déterministe. A cause de l'incertitude sur l'état initial et les frontières de commutations, ce modèle est considéré comme un modèle incertain.

2.7 Conclusion

Dans ce chapitre, un état de l'art des méthodes d'identification et de simulation des systèmes a été présenté. Dans les trois premières sections, on a vu que l'approche ensembliste est une façon de traiter l'incertitude paramétrique du modèle. De plus, l'existence de la sortie du système dans l'enveloppe du modèle intervalle est garantie et ceci montre l'intérêt d'exploiter cet outil pour modéliser les systèmes dans le contexte de la sûreté de fonctionnement. Les méthodes de simulation d'un modèle ont été étudiées dans la section quatre et différentes méthodes pour simuler un modèle intervalle ont été présentées.

Dans les sections 2.5 et 2.6, les méthodes d'identification et de simulation des systèmes complexes avec des dynamiques non-linéaires ont été étudiées. On a vu qu'à cause de la complexité des méthodes d'identification classiques pour ce type de systèmes, les chercheurs envisagent d'autres approches dont l'approche multi-mode. Par cette approche, on représente une dynamique complexe par un ensemble de modèles simples. Cela facilite l'identification de structure et des paramètres du modèle.

Cependant et malgré un nombre important des travaux qui ont déjà été réalisés sur l'identification et la simulation des modèles ensemblistes (pour prendre en compte l'incertitude paramétrique) ainsi que les modèles multi-mode (pour prendre en compte l'incertitude non-paramétrique), très peu de travaux ont été faits sur l'identification et la simulation multi-mode ensembliste.

¹⁸Plus une enveloppe est large, plus elle est imprécise. La définition de la précision d'une enveloppe est donnée dans le chapitre trois.

Dans ce travail, on essaie d'apporter une solution au problème d'identification d'un modèle multi-mode intervalle dans un objectif d'analyse de sûreté de fonctionnement. Dans le chapitre trois, on verra la démarche pour identifier un modèle intervalle en utilisant les observations prélevées sur le système. Ce modèle sera ensuite utilisé pour prédire l'enveloppe englobante de la sortie du système. Dans le chapitre quatre, pour améliorer la qualité de prédiction, l'approche multi-mode sera envisagée. Dans ce chapitre, le problème de l'identification et de la simulation d'un modèle multi-mode intervalle sera étudié.

Chapitre 3

Modèle Intervalle

Les hommes construisent trop de murs et pas assez de ponts.

Isaac Newton

3.1 Introduction

Dans le chapitre précédent, nous avons justifié les raisons qui nous ont conduit à utiliser l'approche ensembliste pour atteindre nos objectifs de recherche. Dans ce document, nous présenterons cette approche puisqu'elle nous permet de prendre en compte l'incertitude paramétrique dans un modèle mathématique. De plus, elle est capable de simuler le comportement du système en calculant l'enveloppe englobante des variables d'état et de la sortie du système pour un ensemble de valeurs d'entrées et de l'état initial. La forme choisie pour caractériser le support d'une variable incertaine est un intervalle et donc dans la deuxième section, on présentera la définition précise d'un intervalle ainsi que l'outil intervalle.

Dans la troisième section, on traitera le problème d'identification de modèles intervalle et on présentera les méthodes de calcul de paramètres des modèles intervalle. On présentera ensuite l'effet de mémoire et le problème du manque d'information dans un modèle dynamique. On proposera une solution pour ce problème en identification ainsi qu'en simulation.

Dans la section quatre, le problème de simulation d'un modèle intervalle sera étudié. Comme on l'a déjà vu dans le chapitre précédent, pour simuler un modèle, on a besoin d'un simulateur. Puisque le modèle est un modèle intervalle, le simulateur sera un simulateur semi-quantitatif. Par rapport à l'objectif de simulation et à l'aide du modèle déjà identifié, ce simulateur sera développé.

Les méthodes d'identification et de simulation seront ensuite appliquées sur un exemple académique.

3.2 Outil Intervalle

Dans cette section, on étudiera brièvement, l’outil intervalle qui est l’outil fondamental sur lequel repose le calcul ensembliste [Moore, 1966], [Neumaier, 1990], [Jaulin et al., 2001a].

3.2.1 Pourquoi un intervalle ?

La représentation précise d’un nombre réel est en général impossible. Par exemple, écrire $\sqrt{2} = 1,414$ est inexact, puisque $1,414$ élevé au carré vaut exactement $1,999396$ et non 2 . Dans un calculateur, les nombres flottants ne sont souvent que des approximations des nombres réels, et les opérations arithmétiques ne sont en général que des approximations inexactes des opérations arithmétiques associées. Par exemple, si on calcule avec un processeur à la norme IEEE 754 $\text{Sin}(\text{Cos}^{-1}(-1,0))$, on obtient $1.22461e-16$ et non 0 . Plus grave, l’évaluation de $\text{Sin}(\text{Cos}^{-1}(-1,0)) == 0$ renvoie *faux* ! L’outil intervalle nous permet de réaliser des opérations dont il est garanti que le résultat soit vrai. Par exemple, on pourra écrire $2 \in [2, 2]$, puis calculer la formule vraie $\sqrt{2} \in [1,414, 1,415]$.

Qu’est ce qu’un intervalle ?

Initialement on appelle intervalle réel un ensemble convexe de nombres délimité par deux nombres réels constituant une borne inférieure et une borne supérieure. Un intervalle contient tous les nombres réels compris entre ses deux bornes.

Représentation d’un intervalle

Dans l’arithmétique des intervalles, un nombre réel x est représenté par une paire de nombres flottants \underline{x} et \bar{x} . Dire que x est représenté par cette paire signifie que x appartient à l’intervalle $[\underline{x}, \bar{x}]$, autrement dit :

$$\underline{x} \leq x \leq \bar{x}$$

Différents types d’intervalles

La définition précédente d’un intervalle regroupe les intervalles de types suivants :

$$\begin{aligned}\{x \in R \mid a < x < b\} &=]a, b[\\ \{x \in R \mid a \leq x \leq b\} &= [a, b] \\ \{x \in R \mid a < x \leq b\} &=]a, b] \\ \{x \in R \mid a \leq x < b\} &= [a, b[\end{aligned}$$

Les intervalles du premier type sont appelés des *intervalles ouverts* ; les seconds des *intervalles fermés*, et les deux derniers des *intervalles semi ouverts*. À ces intervalles sont ajoutés les ensembles de réels inférieurs à une valeur, ou supérieurs à une valeur. On ajoute donc les intervalles de ce type :

$$\begin{aligned} \{x \in R \mid x < a\} &=] - \infty a[\\ \{x \in R \mid x \leq a\} &=] - \infty a] \\ \{x \in R \mid x > a\} &=]a \infty[\\ \{x \in R \mid x \geq a\} &= [a \infty[\end{aligned}$$

Dans ce document, puisqu'on utilise seulement des intervalles fermés, on les appelle pour plus de simplicité des *intervalles*.

Définition 3.2.1. Le centre x_c de l'intervalle $[x] = [\underline{x} \bar{x}]$ est défini comme la valeur moyenne de sa borne supérieure et inférieure :

$$\text{centre}([x]) = \frac{\bar{x} + \underline{x}}{2}$$

Définition 3.2.2. Le rayon λ_x de l'intervalle $[x] = [\underline{x} \bar{x}]$ est défini comme la moitié de la différence entre sa borne supérieure et inférieure :

$$\text{rayon}([x]) = \frac{\bar{x} - \underline{x}}{2}$$

Opérations mathématiques sur les intervalles

L'outil intervalle permet de réaliser des opérations arithmétique dont le résultat est toujours juste. Il est garanti que la solution réelle d'une opération arithmétique est incluse dans le domaine calculé par cet outil. On définit l'image d'une paire d'intervalles par rapport aux opérations arithmétiques comme suit :

$$[z] = [x] \text{ op } [y] = \{x \text{ op } y \mid x \in [x] \wedge y \in [y]\}, \text{ op} \in \{+, -, \times, \div\} \quad (3.2.1)$$

Le formalisme mathématique de l'équation 3.2.1 qui détermine l'image d'une paire d'intervalles par rapport aux opérations arithmétiques s'appelle la *sémantique* des opérations. Ce formalisme garantit que l'image de $[x]$ et $[y]$ par rapport à une opération arithmétique est l'image de x et y par rapport à l'opération pour toutes les valeurs de $x \in [x]$ et $y \in [y]$.

Opérations arithmétiques Considérant la sémantique des opérations mathématiques, les opérations arithmétiques sont définies comme suit :

$$\begin{aligned} - [x] &= -[\underline{x} \bar{x}] = [-\bar{x} \ - \underline{x}] \\ [x] + [y] &= [\underline{x} + \underline{y} \ \bar{x} + \bar{y}] \end{aligned}$$

$$\begin{aligned}
 [x] - [y] &= [x] + (-[y]) = [\underline{x} - \bar{y} \quad \bar{x} - \underline{y}] \\
 [x] \times [y] &= [\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}) \quad \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})] \\
 [x]^2 &= \begin{cases} [\min(\underline{x}^2, \bar{x}^2) \quad \max(\underline{x}^2, \bar{x}^2)] & \text{si } \underline{x}\bar{x} > 0 \\ [0 \quad \max(\underline{x}^2, \bar{x}^2)] & \text{si } \underline{x}\bar{x} \leq 0 \end{cases}
 \end{aligned} \tag{3.2.2}$$

Un cas particulier est la multiplication d'un scalaire par l'intervalle $[-1 \ 1]$:

$$\begin{aligned}
 a \times [-1 \ 1] &= [\min(-a, a) \quad \max(-a, a)] \\
 &= [-|a| \ |a|]
 \end{aligned} \tag{3.2.3}$$

Opérations sur des ensembles Une intersection d'intervalles de R est un intervalle si l'intersection est non vide. Une union d'intervalles de R n'est pas toujours un intervalle. Ce sera un intervalle si les deux intervalles de départ ne sont pas disjoints. L'intervalle qui découle d'une union d'intervalles est composé des éléments (les nombres) allant de la borne inférieure du premier intervalle à la borne supérieure du deuxième intervalle. L'intersection et l'union d'une paire d'intervalles sont définies comme suit :

$$\begin{aligned}
 [z] &= [x] \cap [y] = \{z \in R \mid z \in [x] \wedge z \in [y]\} \\
 &= \begin{cases} [\max(\underline{x}, \underline{y}) \quad \min(\bar{x}, \bar{y})] & \text{si } \max(\underline{x}, \underline{y}) \leq \min(\bar{x}, \bar{y}) \\ \phi & \text{sinon} \end{cases} \\
 [z] &= [x] \cup [y] = \{z \in R \mid z \in [x] \vee z \in [y]\} \\
 &= \begin{cases} [\min(\underline{x}, \underline{y}) \quad \max(\bar{x}, \bar{y})] & \text{si } [x] \cap [y] \neq \phi \\ [x] \cup [y] & \text{sinon} \end{cases}
 \end{aligned} \tag{3.2.4}$$

Par exemple :

$$\begin{aligned}
 [2 \ 5] \cap [1 \ 4] &= [2 \ 4] \\
 [2 \ 5] \cup [1 \ 4] &= [1 \ 5] \\
 [1 \ 2] \cap [3 \ 5] &= \phi \\
 [1 \ 2] \cup [3 \ 5] &= [1 \ 2] \cup [3 \ 5]
 \end{aligned}$$

Tests sur des intervalles Un intervalle est un ensemble de valeurs réelles délimitées par deux bornes. Donc, un test inégalité est vrai si et seulement si il est vrai pour toutes les valeurs de l'intervalle :

$$\begin{aligned}
 [x] \leq c & \quad \text{si} \quad \bar{x} \leq c \\
 [x] < c & \quad \text{si} \quad \bar{x} < c \\
 [x] \geq c & \quad \text{si} \quad \underline{x} \geq c \\
 [x] > c & \quad \text{si} \quad \underline{x} > c \\
 c \in [x] & \quad \text{si} \quad \underline{x} \leq c \leq \bar{x} \\
 [x] \subseteq [y] & \quad \text{si} \quad \underline{x} \geq \underline{y} \wedge \bar{x} \leq \bar{y} \\
 [x] \subset [y] & \quad \text{si} \quad \underline{x} > \underline{y} \wedge \bar{x} < \bar{y}
 \end{aligned}
 \tag{3.2.5}$$

Définition 3.2.3. *Le vecteur intervalle est l'extension d'un vecteur des variables réelles dont les scalaires sont remplacés par des intervalles.*

Un vecteur intervalle définit alors dans l'espace un orthotope aligné avec les axes du repère (appelé aussi boîte ou pavé) de dimension adéquate.

3.2.2 Extension intervalle d'une fonction réelle

Soit f une fonction réelle de n scalaires $\{x_1, x_2, \dots, x_n\}$. Nous définissons la fonction intervalle $[f]$ de n intervalles $\{[x_1], [x_2], \dots, [x_n]\}$ comme une extension de f telle que :

$$[f]_{(x_1, x_2, \dots, x_n)} = f_{(x_1, x_2, \dots, x_n)}, \quad \forall x_i \in [x_i]$$

Ceci signifie qu'une fonction intervalle contient par définition l'ensemble image de la fonction réelle dont elle est l'extension. En revanche, elle peut aussi contenir des valeurs supplémentaires, on parle alors de pessimisme ou de surestimation ou d'approximation extérieure. De manière générale, on a :

$$\{f_{(x_1, x_2, \dots, x_n)} \mid x_i \in [x_i]\} \subseteq [f]_{([x_1], [x_2], \dots, [x_n])}$$

L'extension d'une fonction réelle n'étant pas unique, son choix est important pour limiter ce pessimisme. Par exemple, pour les fonctions polynomiales, on a :

$$a_0 + [x](a_1 + [x](a_2 + [x](a_3 + \dots))) \subseteq a_0 + a_1[x] + a_2[x]^2 + a_3[x]^3 + \dots$$

Définition 3.2.4. Soit f une fonction réelle de R^n dans R définie à l'aide des opérateurs arithmétiques $+$, $-$, \div et \times ainsi que des fonctions élémentaires. On notera $[f_{naturelle}]$ l'extension naturelle de la fonction en remplaçant chaque occurrence de x_i dans f par l'intervalle $[x_i]$ et chaque opérateur ou fonction arithmétique réel par l'opérateur ou la fonction intervalle associé.

Pourquoi l'extension intervalle d'une fonction réelle ?

Généralement, on utilise des méthodes numériques pour déterminer l'image d'ensembles de points lorsque cette image ne peut pas être calculée analytiquement. Pour cela, on emploie traditionnellement les méthodes qui utilisent des procédés de tirage aléatoire comme la méthode de Monte-Carlo. Cependant, le calcul de l'image par ces méthodes est très souvent laborieux. De plus, le domaine calculé par ces méthodes est une approximation intérieure de l'image complète de l'ensemble des points.

Dans l'approche par intervalle, l'ensemble des points est caractérisé par un vecteur d'intervalles (une boîte) et ensuite, l'image de cette boîte est calculée par l'extension intervalle de la fonction réelle. Ce calcul est beaucoup moins coûteux en temps de calcul par rapport aux méthodes précédentes et de plus, le résultat est une approximation extérieure de l'image des points qui peut être calculée en un coup. Cette dernière est une propriété qui est primordiale pour des analyses de sûreté de fonctionnement puisqu'un modèle mathématique est une fonction mathématique de l'espace d'entrée dans l'espace de sortie. Donc, pour garantir que la sortie du modèle englobe la sortie du système, on peut utiliser l'extension intervalle de la fonction du modèle.

Exemple 3.2.1

L'ensemble image de $[x] = [-1 \ 1]$ par rapport à la fonction réelle $f(x) = x^2 + 2x + 1$ est :

$$\{f_{(x)} \mid x \in [-1 \ 1]\} = [0 \ 4]$$

Cette image peut être estimée par rapport à une extension intervalle de cette fonction qui n'est pas forcément l'extension naturelle de celle-ci. On écrit l'extension de la fonction de différentes façons :

$$\begin{aligned} [f_1]_{([x])} &= [x]^2 + 2[x] + 1 = [0 \ 1] + [-2 \ 2] + 1 = [-1 \ 4] \supseteq [0 \ 4] \\ [f_2]_{([x])} &= [x] \times [x] + 2[x] + 1 = [-1 \ 1] + [-2 \ 2] + 1 = [-2 \ 4] \supseteq [0 \ 4] \\ [f_3]_{([x])} &= ([x] + 1)^2 = [0 \ 2]^2 = [0 \ 4] \supseteq [0 \ 4] \end{aligned}$$

où la première expression définit l'extension naturelle de la fonction f . Dans tous les cas, l'ensemble image de $[x]$ par la fonction est incluse dans les intervalles images des extensions intervalles de la fonction réelle.

Exemple 3.2.2

L'objectif de cet exemple est de calculer l'ensemble image d'un vecteur intervalle par rapport à une transformation non-linéaire. Sur la figure 3.1, l'ensemble des valeurs possibles des variables $\{x_1, x_2\}$ a été caractérisé par le

vecteur d'intervalle $[x] = [[x_1] [x_2]] = [[-1 \ 1] [-1 \ 1]]$. On veut déterminer l'image de cet ensemble par rapport à une transformation qui est définie comme suit :

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1^2 \\ x_2^2 + x_1 x_2 \end{bmatrix} \quad (3.2.6)$$

L'ensemble image des points par rapport à cette transformation est difficile à déterminer. Pour avoir une idée de la forme de cette image dans l'espace $\{y_1, y_2\}$, la méthode de Monte-Carlo a été utilisée. On a choisi plusieurs valeurs de points contenus dans $[x]$ et on a calculé les images de ces points par rapport à la transformation. L'approximation extérieure de l'image est ensuite calculée par l'outil intervalle comme suit :

$$\begin{aligned} [y_1] &= [x_1] \times [x_1] = [-1 \ 1] \\ [y_2] &= [x_2] \times [x_2] + [x_1] \times [x_2] = [-1 \ 1] + [-1 \ 1] = [-2 \ 2] \end{aligned}$$

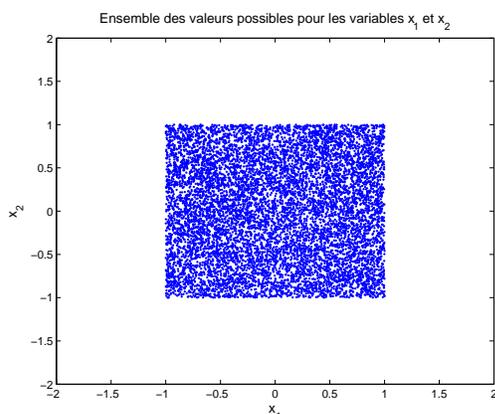


FIG. 3.1 – Valeurs possibles pour le vecteur x .

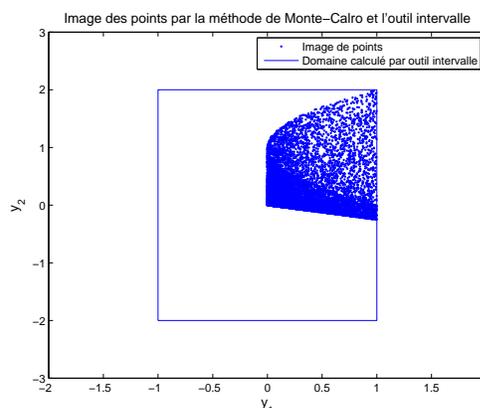


FIG. 3.2 – Images par la méthode Monte-Carlo et l'outil intervalle.

Ces deux images ont été dessinées sur la figure 3.2. On constate que les images des points sont bien incluses dans la boîte calculée par l'outil intervalle.

Définition 3.2.5. Une extension intervalle d'une fonction est monotone par inclusion si :

$$[x_1] \subseteq [x_2] \implies [f]_{([x_1])} \subseteq [f]_{([x_2])}$$

C'est une propriété essentielle qu'on utilisera par la suite. Les fonctions réelles de R^n dans R définies à l'aide des opérateurs $+$, $-$, \div et \times sont toujours des fonctions monotones par inclusion. La structure du modèle qui sera proposée dans la section 3.3.2 et qui sera utilisée dans ce document repose sur des fonctions de ce type. Sans cette propriété, l'inclusion de la solution dans le domaine calculé par l'outil intervalle n'est plus garantie.

3.2.3 Forme alternative pour représenter un intervalle

Un intervalle est originalement déterminé par ses bornes supérieure et inférieure. Dans le but de simplifier les opérations mathématiques sur les intervalles, on va utiliser une forme alternative pour représenter un intervalle. Cette nouvelle forme s'appelle la forme normalisée d'un intervalle.

Forme normalisée d'un intervalle

Un intervalle peut être déterminé par son centre et son rayon ; c'est-à-dire :

$$\begin{aligned} [x] &= [\underline{x} \ \bar{x}] = x_c + \lambda_x \times [-1 \ 1] \\ x_c &= \text{centre}([x]) = \frac{\bar{x} + \underline{x}}{2}, \quad \lambda_x = \text{rayon}([x]) = \frac{\bar{x} - \underline{x}}{2} \geq 0 \end{aligned} \quad (3.2.7)$$

Cette nouvelle forme peut être aussi utilisée pour un vecteur d'intervalles. Soit

$$[x]_{n \times 1} = [[x_1] \ [x_2] \ [x_3] \ \dots \ [x_n]]^T$$

un vecteur d'intervalle dont les éléments sont des intervalles $[x_i]$. La forme normalisée de cet intervalle est comme suit :

$$\begin{aligned} [x] = x_c + \lambda_x \times [v] &= \begin{bmatrix} x_{c,1} \\ x_{c,2} \\ \vdots \\ x_{c,n} \end{bmatrix}_{n \times 1} + \begin{bmatrix} \lambda_{x_1} & 0 & \dots & 0 \\ 0 & \lambda_{x_2} & \dots & 0 \\ & & \ddots & \\ & & & \lambda_{x_n} \end{bmatrix}_{n \times n} \times [v]_{n \times 1} \\ x_{c,i} &= \text{centre}([x_i]), \quad \lambda_{x_i} = \text{rayon}([x_i]) \geq 0, \quad [v] = [[-1 \ 1] \ \dots \ [-1 \ 1]]^T \end{aligned} \quad (3.2.8)$$

Opérations arithmétiques en utilisant la forme normalisée

En utilisant la forme normalisée, l'image d'une paire d'intervalles par rapport aux opérations arithmétiques est déterminée par les centres et les rayons des intervalles. Soit $[x]$ et $[y]$ une paire d'intervalles. En considérant l'équation 3.2.3 et en remplaçant les intervalles $[x]$ et $[y]$ par leurs formes normalisées, on aura :

$$\begin{aligned} [x] &= x_c + \lambda_x \times [-1 \ 1] \\ &= [x_c - \lambda_x \ x_c + \lambda_x] \\ -[x] &= -x_c - \lambda_x \times [-1 \ 1] \\ &= [-x_c - \lambda_x \ -x_c + \lambda_x] \\ [x] + [y] &= (x_c + \lambda_x \times [-1 \ 1]) + (y_c + \lambda_y \times [-1 \ 1]) \\ &= [x_c + y_c - (\lambda_x + \lambda_y) \ x_c + y_c + (\lambda_x + \lambda_y)] \\ [x] - [y] &= (x_c + \lambda_x \times [-1 \ 1]) - (y_c + \lambda_y \times [-1 \ 1]) \end{aligned}$$

$$\begin{aligned}
 &= [x_c - \lambda_x \ x_c + \lambda_x] + [-y_c - \lambda_y \ -y_c + \lambda_y] \\
 &= [x_c - y_c - (\lambda_x + \lambda_y) \ x_c - y_c + (\lambda_x + \lambda_y)] \quad (3.2.9)
 \end{aligned}$$

Le résultat de l'addition et de la soustraction d'une paire d'intervalles est un intervalle. Pour multiplier un scalaire par un intervalle, on aura :

$$\begin{aligned}
 a \times [x] &= a \times (x_c + \lambda_x \times [-1 \ 1]) \\
 &= ax_c + a\lambda_x \times [-1 \ 1] \\
 &= [ax_c - |a|\lambda_x \ ax_c + |a|\lambda_x]
 \end{aligned}$$

La multiplication d'une paire d'intervalles sera comme suit :

$$\begin{aligned}
 [x] \times [y] &= (x_c + \lambda_x \times [-1 \ 1]) \times (y_c + \lambda_y \times [-1 \ 1]) \\
 &= x_c y_c + x_c \lambda_y \times [-1 \ 1] + y_c \lambda_x \times [-1 \ 1] + \lambda_x \lambda_y \times [-1 \ 1] \\
 &= x_c y_c + [-(|x_c| \lambda_y + |y_c| \lambda_x + \lambda_x \lambda_y) \ + (|x_c| \lambda_y + |y_c| \lambda_x + \lambda_x \lambda_y)] \\
 &= [x_c y_c - (|x_c| \lambda_y + |y_c| \lambda_x + \lambda_x \lambda_y) \ x_c y_c + (|x_c| \lambda_y + |y_c| \lambda_x + \lambda_x \lambda_y)]
 \end{aligned}$$

Par contre, le résultat du produit scalaire de deux vecteurs d'intervalles $[x]_{n \times 1}$ et $[y]_{n \times 1}$ est l'intervalle $[z]_{1 \times 1}$:

$$\begin{aligned}
 [z] = [x]^T \times [y] &= (x_c + \lambda_x \times [v_{\lambda_x}])^T \times (y_c + \lambda_y \times [v_{\lambda_y}]) \\
 &= (x_c^T + [v_{\lambda_x}]^T \times \lambda_x^T) \times (y_c + \lambda_y \times [v_{\lambda_y}]) \\
 &= x_c^T y_c + x_c^T \lambda_y [v_{\lambda_y}] + [v_{\lambda_x}]^T \lambda_x^T y_c + [v_{\lambda_x}]^T \lambda_x^T \lambda_y [v_{\lambda_y}] \quad (3.2.10)
 \end{aligned}$$

où $[z] = [\underline{z}, \bar{z}]$. En considérant l'expression de l'équation 3.2.10, les bornes supérieure et inférieure de cet intervalle sont comme suit :

$$\begin{aligned}
 \bar{z} &= x_c^T y_c + (Sum(|x_c^T| \lambda_y) + Sum(\lambda_x^T |y_c|) + Sum(Diag(\lambda_x^T \lambda_y))) \\
 \underline{z} &= x_c^T y_c - (Sum(|x_c^T| \lambda_y) + Sum(\lambda_x^T |y_c|) + Sum(Diag(\lambda_x^T \lambda_y))) \quad (3.2.11)
 \end{aligned}$$

La fonction $Sum(x)$ calcule la somme des éléments du vecteur x et le vecteur $Diag(\lambda)$ est composé des éléments situés sur la diagonale de la matrice carrée λ . Par exemple, dans l'équation 3.2.8, on a :

$$Sum(x_c) = \sum_{i=1}^n x_{c,i}$$

$$Diag(\lambda_x) = [\lambda_{x_1} \quad \lambda_{x_2} \quad \dots \quad \lambda_{x_n}]^T$$

La forme normalisée permet de simplifier le calcul par intervalle. C'est le cas de la multiplication d'une paire d'intervalles et surtout le produit scalaire d'une paire de vecteurs d'intervalles dont les expressions des bornes supérieure et inférieure sont données dans l'équation 3.2.11 et qui seront utilisées très souvent par la suite. Ce formalisme permettra, puisque le problème d'identification des paramètres d'un modèle intervalle sera posé, d'établir sans difficulté les expressions analytiques des contraintes à satisfaire et du critère de précision à minimiser.

3.2.4 Effet d'enveloppement dans le calcul par intervalle

L'image d'un vecteur d'intervalles par rapport à une fonction n'est pas en général une boîte. Donc, englober cette image par un vecteur d'intervalle conduit à une source de surestimation. Dans le cas de l'utilisation d'algorithmes itératifs, ce pessimisme risque de s'accumuler au cours des itérations et peut conduire à l'explosion des domaines calculés. L'effet d'accumulation de ce pessimisme au cours des itérations s'appelle *l'effet d'enveloppement*.

On reprend l'exemple de la transformation non-linéaire de l'équation 3.2.6. L'image de $[x] = [[-1 \ 1], [-1 \ 1]]^T$ par rapport à cette transformation est une forme non-linéaire. On a une approximation de cette forme sur la figure 3.3 qui a été obtenue par la méthode de Monté-Carlo. Si on englobe cette forme par un vecteur d'intervalles, c'est-à-dire une boîte, on introduit les points supplémentaires dans le domaine calculé. Sur la figure 3.3, cette image a été représentée par la boîte suivante :

$$\begin{aligned} [y_1] &= [x_1]^2 = [0 \ 1] \\ [y_2] &= [x_2]^2 + [x_1] \times [x_2] = [0 \ 1] + [-1 \ 1] = [-1 \ 2] \end{aligned}$$

Cette boîte est le domaine calculé par l'arithmétique par intervalle. Cependant, réduire l'image à chaque étape de calcul à une simple boîte n'est pas la seule source de surestimation. Une autre source de pessimisme est la surestimation dues aux opérations intervalles elles-mêmes. Par exemple, l'image de $[x]$ peut aussi être calculé comme suit :

$$\begin{aligned} [y_1] &= [x_1] \times [x_1] = [-1 \ 1] \times [-1 \ 1] = [-1 \ 1] \\ [y_2] &= [x_2] \times [x_2] + [x_1] \times [x_2] = [-1 \ 1] + [-1 \ 1] = [-2 \ 2] \end{aligned}$$

Cette boîte a été aussi représentée sur la figure 3.3.

L'accumulation de la surestimation et l'explosion du domaine de calcul sont des phénomènes rencontrés assez souvent dans les modèles itératifs intervalle. Par exemple, si on considère $\pm 10\%$ d'incertitude sur les paramètres du modèle de l'équation 1.2.1, on obtient :

$$[x_{k+1}] = [A] \times [x_k] + [B] \times u_k$$

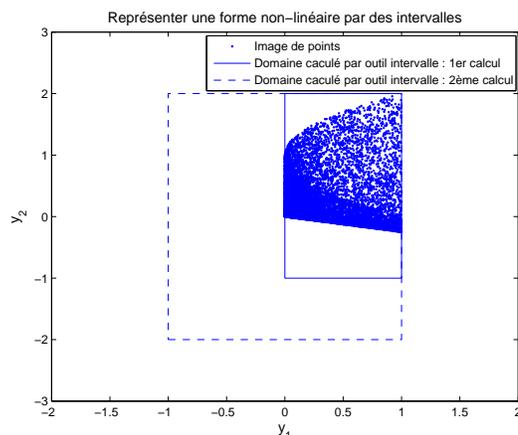


FIG. 3.3 – Deux sources de l’effet d’enveloppement dû à la surestimation de l’image réelle par un vecteur d’intervalle et utilisation de différentes extensions intervalle de la fonction réelle.

$$[A] = \begin{bmatrix} [0, 7368 & 0, 9006] & [0 & 0] \\ [0, 1549 & 0, 1894] & [0, 8143 & 0, 9953] \end{bmatrix} \quad [B] = \begin{bmatrix} [1, 6314 & 1, 9939] \\ [0, 1630 & 0, 1992] \end{bmatrix}$$

et si on évalue l’état initial $[x_0] = [[0 \ 1], [0 \ 1]]^T$ par rapport à cette dynamique, à cause de l’effet d’enveloppement, le domaine calculé va exploser. Ce phénomène est montré sur la figure 3.4.

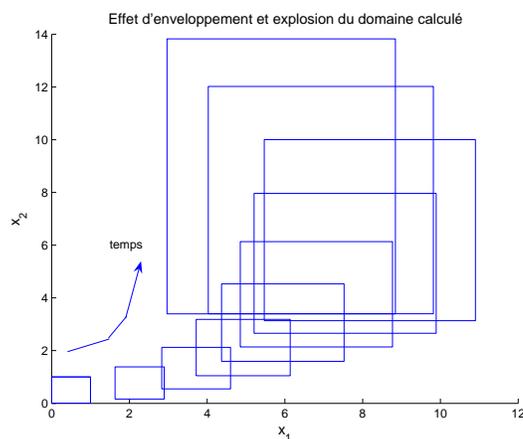


FIG. 3.4 – Explosion du domaine de calcul à cause de l’effet d’enveloppement.

L’effet d’enveloppement n’apparaît pas seulement pendant l’évaluation des modèles incertains. Il est présent aussi dans le cas des modèles déterministes où l’état initial est incertain. Par exemple, pour le modèle déterministe de l’équation 3.2.12, si l’état initial est caractérisé par $[x_0] = [[0 \ 0, 5], [0 \ 0, 5]]^T$, la taille d’enveloppe englobante des variables

d'état augmente après chaque itération. L'évaluation de cette boite a été illustrée sur la figure 3.5.

$$\begin{aligned}
 x_{k+1} &= Ax_k + Bu_k \\
 A &= \begin{bmatrix} 0.98 & -0.05 \\ 0.9 & 0.55 \end{bmatrix} & B &= \begin{bmatrix} -8 \\ 1 \end{bmatrix} \\
 u_k &= 0, 1 \quad 0 < k < 8 \\
 u_k &= 0 \quad k \geq 8
 \end{aligned} \tag{3.2.12}$$

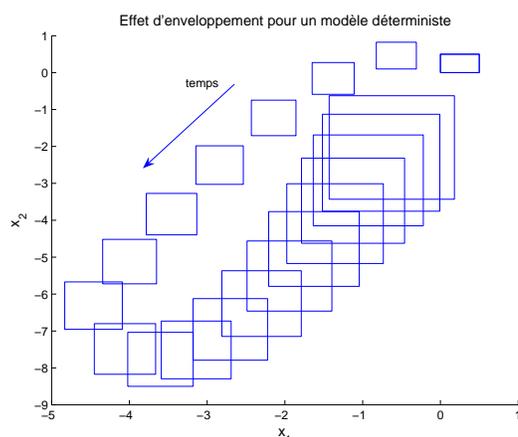


FIG. 3.5 – Évaluation de l'enveloppe des variables d'état.

3.3 Identification d'un Modèle Intervalle

Dans cette section, on proposera une solution au problème d'identification d'un système par un modèle intervalle. La structure du modèle est une structure linéaire de type *réponse impulsionnelle à horizon fini* ou *relation entrée-sortie*. Les paramètres du modèle sont des variables incertaines bornées variantes dans le temps caractérisées par des intervalles. Ces paramètres doivent être calculés tels que le modèle explique tous les comportements du système. Autrement dit, l'enveloppe calculée par le modèle doit encadrer toutes les observations sur la sortie du système :

$$\begin{aligned}
 \forall k, y_k &\leq \bar{y}_k \\
 y_k &\geq \underline{y}_k
 \end{aligned}$$

où y_k est la sortie du système est où \hat{y}_k est la sortie estimée par le modèle. De plus, cette enveloppe doit être la plus petite possible. L'exemple d'un modèle linéaire intervalle est montré pour un ensemble des points dans l'espace de sortie sur la figure 3.6. Ce problème sera traité au fur et à mesure par la suite.

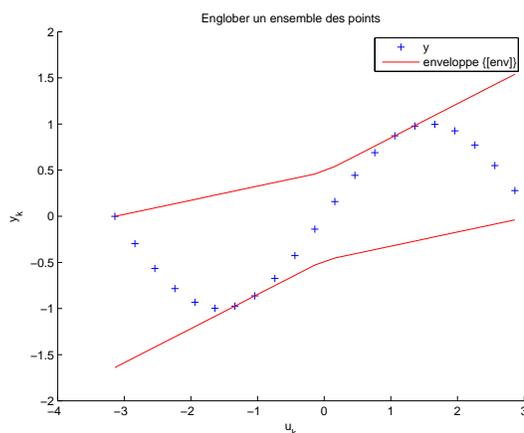


FIG. 3.6 – Modèle linéaire intervalle.

Lors de la discussion sur l'identification de modèles intervalle dans le chapitre précédent, on a vu que pour déterminer les paramètres d'un modèle intervalle à partir d'un jeu de données, de la structure du modèle et d'un critère d'optimisation, il faut aussi définir la sémantique d'identification à partir de l'objectif de modélisation. Par conséquent, nous modifions la boucle classique de l'identification de la figure 2.3. La nouvelle boucle sur la figure 3.7 détermine la démarche d'identification d'un modèle intervalle. Par la suite, on suivra cette nouvelle boucle étape par étape.

3.3.1 Jeu de données expérimental

Pour identifier les paramètres du modèle, comme pour toutes les autres méthodes d'identification, la première étape est de préparer un jeu de données expérimental. Ce jeu de données doit être suffisamment riche pour représenter tous les comportements du système. Pour l'obtenir, il faut faire une expérience sur le système. Pendant cette expérience, on excite le système par différentes valeurs d'entrée et on mesure la sortie. Des mesures sur l'entrée et la sortie du système produisent l'ensemble des paires $DS = \{(u_k, y_k)\}$ qui sera utilisé par la suite pour identifier les paramètres du modèle. Différentes méthodes existent pour concevoir une telle expérience et choisir un signal convenable pour exciter le système dans la littérature. Ce choix est un problème classique en automatique [Ljung, 1999].

Bien souvent, on n'a pas la possibilité d'effectuer une expérience en excitant librement le système. Donc, on est obligé d'utiliser les données acquises pendant le fonctionnement normal du système. Cela nous empêche d'identifier les paramètres dans certaines zones de

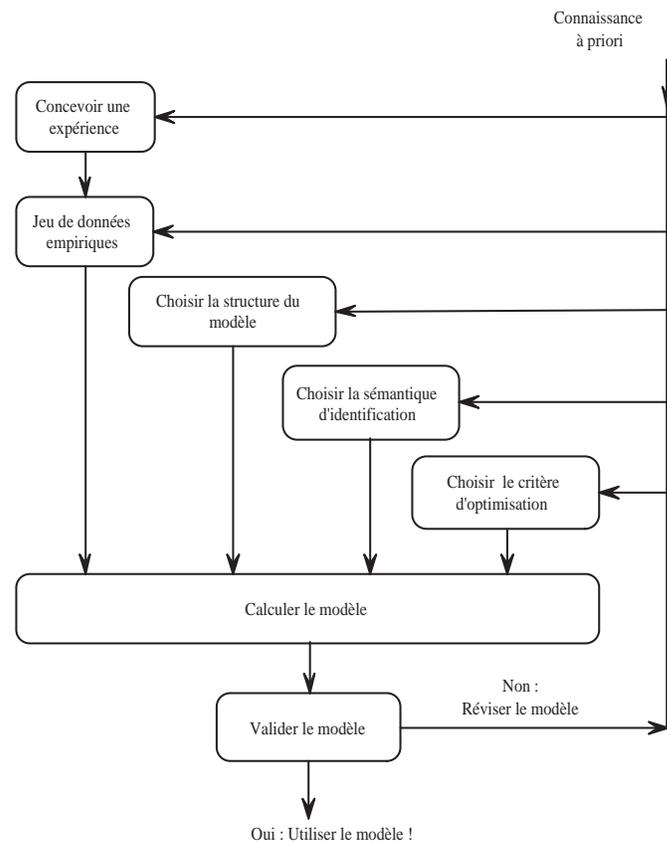


FIG. 3.7 – Boucle d'identification de modèle intervalle.

fonctionnement. Dans ce travail, on suppose que DS est donné et qu'il est suffisamment riche pour représenter tous les comportements du système à modéliser.

3.3.2 Structure de modèle

La deuxième étape dans la procédure d'identification proposée sur la figure 3.7 est de choisir la structure de modèle. Dans cette section, on présentera différentes structures pour les modèles mathématiques. On commencera par étudier le modèle réponse impulsionnelle qui permet une caractérisation complète du système, et ensuite, une famille de modèles boîte noire sera présentée. Considérant les structures présentées et l'objectif de ce travail, on choisira ensuite une structure pour le modèle qui sera utilisé dans tout ce qui suit.

Réponse impulsionnelle

En considérant un système avec une entrée scalaire $u(t)$ et une sortie scalaire $y(t)$, il est bien connu que si le système est linéaire, temps-invariant et causal, il peut être représenté par sa *réponse impulsionnelle continue* $g(\tau)$ comme suit [Ljung, 1999]¹ :

$$y(t) = \int_{\tau=0}^{\infty} g(\tau)u(t - \tau)d\tau \quad (3.3.1)$$

En connaissant $\{g(\tau)\}_{\tau=0}^{\infty}$ et $u(s)$ pour $s \leq t$, on peut théoriquement calculer la sortie correspondante $y(s)$ pour toutes les valeurs d'entrée. Donc, la réponse impulsionnelle permet une caractérisation complète du système.

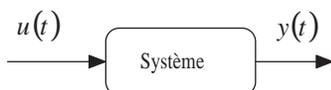


FIG. 3.8 – Système.

On travaille très souvent avec les entrées et les sorties échantillonnées puisque c'est le mode courant d'acquisition de données. Par conséquent, on suppose qu'on observe la sortie aux instants d'échantillonnage $t_k = kT$, $k = 1, 2, \dots$. T s'appelle la *période d'échantillonnage*. Bien souvent, dans les applications basées sur les plateformes informatiques, l'entrée $u(t)$ est gardée constante pendant la période T ; c'est-à-dire :

$$u(t) = u_k, \quad kT \leq t \leq (k + 1)T \quad (3.3.2)$$

Cette hypothèse simplifie l'analyse des systèmes. En substituant l'équation 3.3.2 dans 3.3.1, on obtient :

$$\begin{aligned} y(kT) &= \int_{\tau=0}^{\infty} g(\tau)u(kT - \tau)d\tau = \sum_{i=1}^{\infty} \int_{\tau=(i-1)T}^{iT} g(\tau)u(kT - \tau)d\tau \\ &= \sum_{i=1}^{\infty} \left[\int_{\tau=(i-1)T}^{iT} g(\tau) d\tau \right] u_{k-i} = \sum_{i=1}^{\infty} g_T(i)u_{k-i} \end{aligned} \quad (3.3.3)$$

où

$$g_T(i) = \int_{\tau=(i-1)T}^{iT} g(\tau)d\tau \quad (3.3.4)$$

¹Pour présenter la réponse impulsionnelle et les modèles de boîte noire les plus connus, on emploiera les notations qui ont été utilisées dans l'ouvrage [Ljung, 1999].

L'équation 3.3.3 détermine la sortie aux instants d'échantillonnages et la séquence $\{g_T(i)\}_{i=1}^{\infty}$ s'appelle la *réponse impulsionnelle discrète* du système.

Dans le cas général, on doit utiliser les équations 3.3.2, 3.3.3 et 3.3.4 pour calculer la sortie $y(kT)$. Cependant, pour simplifier les notations, on effectuera un changement de variable et on suppose que $k \leftarrow kT$ ². On obtient donc,

$$y_k = \sum_{i=1}^{\infty} g_i u_{k-i}, \quad k = 1, 2, \dots \quad (3.3.5)$$

où

$$g_i = g_T(i)$$

Bien souvent et à cause des perturbations, l'équation 3.3.5 n'est pas juste. Les perturbations sont des signaux externes qui ne peuvent pas être manipulés. On prend une approche pragmatique et on suppose que l'effet des perturbations peut être intégré dans un terme d'erreur additif ϵ_k sur la sortie de système :

$$y_k = \sum_{i=1}^{\infty} g_i u_{k-i} + \epsilon_k \quad (3.3.6)$$

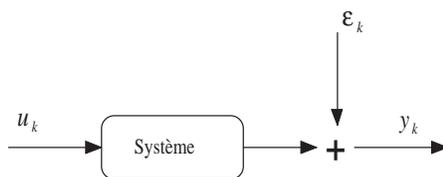


FIG. 3.9 – Système avec l'erreur.

Dans l'approche stochastique, les perturbations sont représentées comme la réalisation de systèmes stochastiques. On peut parfois mesurer les perturbations séparément. Cependant, elles sont souvent perceptibles uniquement par leurs effets sur la sortie. L'étude de tels systèmes et le calcul de ϵ_k à l'instant k en utilisant une densité de probabilité conditionnelle à plusieurs dimensions³ est dans le plupart des cas très laborieux. Par conséquent, on préfère utiliser une approche plus simple. Supposons que l'erreur additive soit donnée comme suit :

$$\epsilon(k) = \sum_{i=0}^{\infty} h_i e_{k-i} \quad (3.3.7)$$

²Dans ce document, pour un modèle dynamique, k représente le temps et l'instant d'échantillonnage. Cependant, pour un modèle statique, il est seulement utilisé pour énumérer l'indice de données

³Conditional joint probability density function

où $\{e(k)\}$ est une séquence de variables aléatoires et indépendantes avec une loi de distribution⁴ donnée. L'équation 3.3.7 fonctionne pour un certain nombre de lois de distributions de ϵ_k . Pourtant, elle est assez souple pour être utilisée pour la plupart des applications pratiques.

Dans la littérature, les sommes 3.3.6 et 3.3.7 sont souvent décrites par les opérateurs *avance*⁵ et *retard*⁶. On définit l'opérateur avance q :

$$qu_k = u_{k+1}$$

et l'opérateur retard q^{-1}

$$q^{-1}u_k = u_{k-1}$$

et donc, on peut re-écrire l'équation 3.3.5 comme suit :

$$\begin{aligned} y_k &= \sum_{i=1}^{\infty} g_i u_{k-i} = \sum_{i=1}^{\infty} g_i (q^{-i} u_k) \\ &= \left[\sum_{i=1}^{\infty} g_i q^{-i} \right] u_k = G(q) u_k \end{aligned} \quad (3.3.8)$$

$G(q)$ s'appelle la *fonction de transfert* du système linéaire représentée par l'équation 3.3.5. On peut aussi déterminer la fonction de transfert de l'erreur ϵ comme suit :

$$H(q) = \sum_{i=0}^{\infty} h_i q^{-i}$$

Donc, la description basique pour un système linéaire en considérant l'effet des perturbations sur la sortie comme un terme additif sera :

$$y_k = G(q)u_k + H(q)e_k \quad (3.3.9)$$

Spécifier un modèle probabiliste consiste donc à déterminer la fonction G , la fonction H et la loi de distribution f_e de l'erreur :

$$\begin{aligned} y_k &= G(q)u_k + H(q)e_k \\ f_e(\cdot), \text{ la FDP de } e \end{aligned} \quad (3.3.10)$$

⁴Probability density function

⁵forward shift operator

⁶backward shift operator

Dans l'approche ensembliste, l'effet des perturbations sur la sortie est caractérisé par une variable incertaine et bornée e_k . Dans cette approche, au lieu de déterminer $f_e(\cdot)$, on doit identifier le support de cette variable.

Pour finir, utiliser des séquences infinies $\{g_i\}$ et $\{h_i\}$ n'est pas utilisable en pratique et on préfère déterminer les fonctions de transfert G et H sous la forme d'un nombre fini de valeurs réelles. Dans la suite, on présentera une famille de modèles basés sur la fonction de transfert. De telles structures s'appellent des *modèles boîte noire*.

Modèles basés sur la fonction de transfert

La façon la plus immédiate pour déterminer G et H est de les représenter comme des fonctions rationnelles en q de façon à ce que les paramètres du modèle soient les coefficients du numérateur et du dénominateur. Pour étudier les différentes façons de réaliser une telle paramétrisation, il est possible de consulter l'annexe A. On expliquera par la suite le modèle ARX ⁷ qui sera utilisé comme structure du modèle dans ce document.

Modèle ARX

Probablement, la relation la plus simple entre l'entrée et la sortie observées peut être décrite par une équation aux différences comme suit :

$$y_k + a_1 y_{k-1} + \dots + a_{n_a} y_{k-n_a} = b_1 u_{k-1} + \dots + b_{n_b} u_{k-n_b} + e_k \quad (3.3.11)$$

Puisque l'erreur e_k se présente sous la forme d'un unique terme additif, ce modèle s'appelle très souvent un *modèle à erreur d'équation*. Les paramètres du modèle sont :

$$\theta = [a_1 \ a_2 \ \dots \ a_{n_a} \ b_1 \ \dots \ b_{n_b}]$$

Si on introduit les notations suivantes :

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

et

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

on voit que :

⁷AutoRegression eXtra input

$$G(q) = \frac{B(q)}{A(q)}, \quad H(q) = \frac{1}{A(q)}$$

Dans le cas où $n_a = 0$, ce modèle devient un modèle réponse impulsionnelle à horizon fini⁸.

Si à l'instant k , l'erreur e_k est indéterminée, on utilisera la forme déterministe du modèle. Dans ce cas, on peut montrer que si y_s et u_s pour $s < k$ sont connus, la prédiction à un pas de la sortie de système par le modèle de l'équation 3.3.9 est calculée par l'expression suivante [Ljung, 1999] :

$$\hat{y}(k, \theta) = H^{-1}(q)G(q)u_k + [1 - H^{-1}(q)]y_k \quad (3.3.12)$$

On utilise θ dans l'expression $\hat{y}(k, \theta)$ puisque la valeur prédite de la sortie dépend au vecteur de paramètres de modèle θ . Pour un modèle ARX , l'équation 3.3.12 devient :

$$\hat{y}(k, \theta) = B(q)u_k + [1 - A(q)]y_k \quad (3.3.13)$$

On introduit la notation

$$\varphi_k = [-y_{k-1} \dots -y_{k-n_a} \quad u_{k-1} \dots u_{k-n_b}]^T$$

En utilisant le vecteur φ_k , l'équation 3.3.13 peut être re-écrite :

$$\hat{y}(k, \theta) = \theta^T \varphi_k \quad (3.3.14)$$

Donc, la prédiction peut se faire par un produit scalaire entre le vecteur des données φ_k qui est connu et le vecteur des paramètres θ . En statistique, un tel modèle s'appelle une *régression linéaire* et le vecteur φ_k s'appelle le *vecteur de régression*. La simplicité de ce modèle pour prédire la sortie du système en fait le premier choix pour de nombreuses applications. En considérant l'équation 3.3.14, la sortie observée est égale à :

$$y_k = \hat{y}(k, \theta) + e_k$$

Aspect statique de la structure ARX

Le modèle ARX qui a été présenté est reconnu dans la littérature comme un modèle dynamique. Dans ce modèle, la sortie, qui est scalaire, est calculée en fonction des paramètres du modèle (qui doivent être identifiés) ainsi que d'un certain nombre d'entrées

⁸Finite Impulse Response (FIR).

et/ou de sorties ordonnées du système décalées dans le temps (qui sont données). Autrement dit, le modèle est une fonction de l'espace d'entrée et/ou de l'espace de sortie du système dans lesquels l'ordre des points est important. Chacun de ces espaces a une seule dimension (figure 3.10).

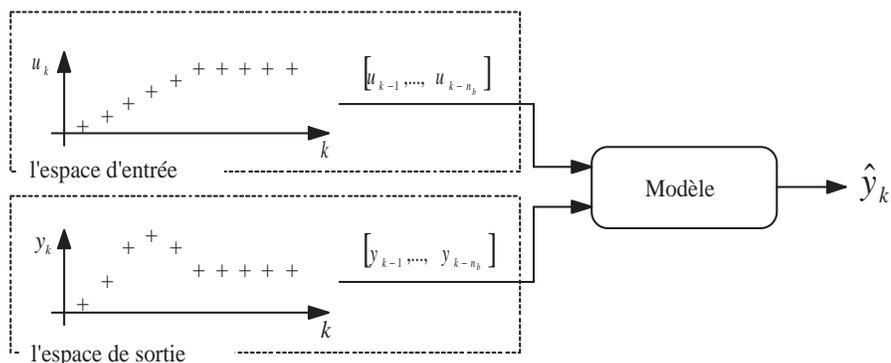


FIG. 3.10 – Représentation dynamique de la structure *ARX*.

Pourtant, un modèle *ARX* peut aussi être interprété comme un modèle statique multi-entrée mono-sortie. Dans ce cas, l'espace d'entrée du modèle statique est l'espace du vecteur de régression du modèle *ARX*. Dans cet espace, la séquence des entrées et des sorties décalées dans le temps est projetée sur les dimensions de l'espace et par conséquent, la sortie du modèle peut être calculée pour chaque point dans cet espace indépendamment des autres points. L'indice k pour les points dans cet espace ne représente plus la notion du temps et il est seulement utilisé pour les énumérer (figure 3.11).

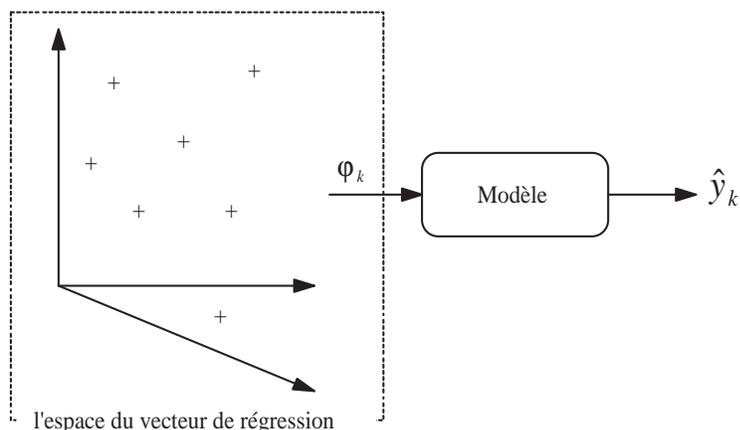


FIG. 3.11 – Représentation statique de la structure *ARX*.

De cette remarque, on justifiera dans la section 4.5.3 du chapitre suivant le fait de déterminer le domaine de validité de chaque modèle local dans l'espace du vecteur de régression. Pourtant, on gardera dans ce qui suit la terminologie de modèle dynamique pour la structure *ARX* puisque c'est le terme qui est utilisé dans la littérature.

Structure proposée

Le modèle qu'on va utiliser dans ce document est un modèle de type *ARX* avec une entrée scalaire u_k et une sortie scalaire y_k et qui s'écrit sous la forme suivante (dans un contexte d'identification) :

$$A(q)y_k = B(q)u_k + e_k$$

On propose deux structures pour les polynômes A et B :

- $n_a = 0, n_b = HG$
- $n_a = HG, n_b = HG$

Le premier cas spécifie un modèle réponse impulsionnelle à horizon fini et le deuxième cas est un modèle relation entrée-sortie. *HG* s'appellera dans la suite *l'horizon glissant*⁹ du modèle et pour déterminer A et B , les valeurs des paramètres $\{a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}\}$ doivent être déterminées. Dans un système physique et à cause de l'incertitude paramétrique, les paramètres du modèle peuvent être considérés comme variant dans le temps. Pour prendre en compte l'incertitude paramétrique, les paramètres du modèle sont considérées comme des variables incertaines bornées et donc, à l'instant k , le vecteur des paramètres du modèle est représenté par :

$$\theta_k = [a_{k,1} \dots a_{k,n_a} b_{k,1} \dots b_{k,n_b}]^T$$

Les valeurs que $a_{k,i}$ et $b_{k,i}$ prennent aux différents instants sont supposées indépendantes. Dans l'approche probabiliste, l'effet des perturbations est intégrée dans le terme additif e_k . Dans cette approche, $\{e_k\}$ est une série de valeurs aléatoires et indépendantes caractérisée par une loi de distribution. Dans l'approche proposée dans ce document, l'erreur à l'instant k est représentée par une variable incertaine et bornée e_k . Cette variable s'appellera *l'erreur additive*.

Pour ramener le modèle sur une forme vectorielle, on définit le vecteur de régression φ_k qui contient *HG* entrées et/ou *HG* sorties du système décalées dans le temps comme suit :

$$\begin{aligned} \text{si } n_a = 0, \quad n_b = HG \text{ alors } & \varphi_k = [u_{k-1} \dots u_{k-HG}]^T \\ & \theta_k = [b_{k,1} \dots b_{k,HG}]^T \\ \text{si } n_a = HG, \quad n_b = HG \text{ alors } & \varphi_k = [-y_{k-1} \dots -y_{k-HG} \quad u_{k-1} \dots u_{k-HG}]^T \\ & \theta_k = [a_{k,1} \dots a_{k,HG} \quad b_{k,1} \dots b_{k,HG}]^T \end{aligned}$$

En considérant l'équation A.0.4, la prédiction à un pas de la sortie du système se fait par :

⁹Moving Horizon

$$\widehat{y}_k = \theta_k^T \times \varphi_k \quad (3.3.15)$$

et la sortie observée à l'instant k est égale à :

$$y_k = \widehat{y}_k + e_k \quad (3.3.16)$$

Le vecteur θ_k représente les valeurs des paramètres du modèle à l'instant k . On caractérisera les paramètres incertains par des supports intervalles :

$$\begin{aligned} [\theta] &= [[a_1] \dots [a_{n_a}] [b_1] \dots [b_{n_b}]]^T \\ [e] &= [\underline{e}, \bar{e}] \end{aligned}$$

Le vecteur des variables incertaines θ_k et la variable e_k sont variants dans le temps tandis que leurs supports $[\theta]$ et $[e]$ sont invariants dans le temps. On verra dans la section 3.3.3 que ces supports doivent être déterminés tels que :

$$\forall k, y_k \in [env]_{(k, [\theta], [e])} = [\widehat{y}]_k + [e] \quad (3.3.17)$$

où¹⁰

$$[\widehat{y}]_k = [\theta]^T \varphi_k$$

$[env]_{(k, [\theta], [e])}$ englobe l'ensemble des valeurs possibles de la sortie du système à l'instant k . Puisque ce domaine est une fonction du modèle ainsi que de l'erreur additive, il comporte les arguments $([\theta], [e])$.

Si y_s et u_s sont connus pour $s < k$, l'équation 3.3.17 prédit l'enveloppe englobante de la sortie à l'instant k . L'enveloppe englobante de la sortie de système par le prédicteur 3.3.17 est déterminée par :

$$\{[env]_{([\theta], [e])}\} = \{[env]_{(HG+1, [\theta], [e])}, [env]_{(2, [\theta], [e])}, [env]_{(3, [\theta], [e])}, \dots\} \quad (3.3.18)$$

◇

Dans ce document, pour gérer l'effet de mémoire¹¹ de la structure proposée, on aura aussi besoin d'un modèle statique intervalle. Ce modèle est un modèle linéaire avec une entrée scalaire u_k et une sortie scalaire y_k ¹² de la forme suivante :

¹⁰L'indice k dans $[\widehat{y}]_k$ précise que le support de la variable incertaine \widehat{y} varie dans le temps et dépend de k .

¹¹On étudiera l'effet de mémoire dans la section 3.3.6

¹²Comme on en a déjà parlé, dans un modèle statique, l'indice k est seulement utilisé pour énumérer les données.

$$y_k = b_k u_k + e_k$$

où b_k , le paramètre de modèle à l'instant k , est aussi un scalaire. Le support de cette variable est l'intervalle $[b]$. Pour rester homogène, on introduit :

$$\begin{aligned}\varphi_k &= u_k \\ \theta_k &= b_k\end{aligned}$$

et donc, la prédiction de la sortie pour l'entrée u_k est calculée par¹³ :

$$\hat{y}_k = \theta_k^T \varphi_k$$

et la sortie observée est égale à :

$$y_k = \hat{y}_k + e_k \tag{3.3.19}$$

L'extension de l'intervalle de l'équation 3.3.19 doit garantir que :

$$\forall k, y_k \in [env]_{(k, [\theta], [e])} = [\hat{y}]_k + [e]$$

qui est homogène avec la structure *ARX* proposée.

Dans ce document, identifier les paramètres du modèle intervalle où brièvement, identifier le modèle intervalle

$$[env]_{(k, [\theta], [e])} = [\theta]^T \times \varphi_k + [e] \tag{3.3.20}$$

consiste à déterminer le support $[\theta]$ des paramètres ainsi que le support $[e]$ de l'erreur additive en considérant deux aspects :

- Le modèle doit représenter le comportement du système dans nos objectifs de sûreté de fonctionnement. Cela signifie qu'il doit être capable d'expliquer les différents comportements observés du système.
- De plus, il doit être le plus précis possible au sens d'un critère à préciser.

Ces deux points vont être détaillés dans ce qui suit. Cette définition est comparable avec la définition d'un modèle probabiliste 3.3.10.

Remarque. L'équation 3.3.20 détermine aussi le prédicteur à un pas de l'enveloppe englobante pour le modèle *ARX* et le modèle statique proposés. Ce prédicteur peut être utilisé dans une analyse pour laquelle on a accès aux sorties du système. \diamond

¹³Puisque θ_k est un scalaire, on a $\theta_k = \theta_k^T$

On a déjà vu dans les sections précédentes qu'un intervalle ou un vecteur intervalle peut s'écrire sous la forme normalisée 3.2.8. C'est aussi le cas pour le vecteur d'intervalles $[\theta]$ et l'intervalle $[e]$ qui peuvent être déterminés par $\theta_c \in R^{n \times 1}$, $\lambda_\theta \in R^{n \times n}$, $e_c \in R$ et $\lambda_e \in R$ ¹⁴. On verra dans la section 3.3.5 que le problème d'identification du modèle se traduira par un problème d'optimisation des paramètres θ_c , λ_θ , e_c et λ_e sous contraintes.

Remarque. On a déjà expliqué que le domaine $[env]_{(k, [\theta], [e])}$ est une fonction du modèle et de l'erreur additive; c'est pour cette raison qu'il porte les indices $([\theta], [e])$. Pourtant, pour simplifier les notations, on éliminera dans la suite ces indices en se rappelant que $[env]_k$ dépend toujours de $[\theta]$ et $[e]$. \diamond

3.3.3 Sémantique d'identification

Compte tenu de l'objectif d'identification qui a été présenté dans la section *Problématique* du chapitre un, le support des paramètres $[\theta]$ du modèle ainsi que le support $[e]$ de l'erreur additive doivent être identifiés tels que :

$$\forall k, y_k \in [env]_k = [\theta]^T \times \varphi_k + [e] \quad (3.3.21)$$

autrement dit :

$$\forall k, \exists \theta_k \in [\theta], \exists e_k \in [e] \mid y_k = \theta_k^T \times \varphi_k + e_k \quad (3.3.22)$$

De plus, l'enveloppe englobante de la sortie calculée par le modèle en utilisant $[\theta]$ et $[e]$ comme paramètres doit être l'enveloppe la plus précise au sens du critère d'optimisation qui sera défini dans la section 3.3.4.

Puisque les paramètres du modèle sont considérés indépendants, le domaine déterminé par le vecteur intervalle $[\theta]$ est une boîte dans l'espace des paramètres. On a déjà vu dans l'équation 3.2.8 qu'un vecteur intervalle peut être décrit sous la forme normalisée. On a donc :

$$[\theta] = \theta_c + \lambda_\theta \times [v]$$

où le vecteur θ_c détermine le centre de la boîte et la matrice diagonale λ_θ définit la largeur de cette boîte sur chaque axe. Cette boîte est montrée sur la figure 3.12 pour un modèle réponse impulsionnelle avec l'horizon glissant égale à $HG = 2^{15}$. L'équation

¹⁴Pour le modèle *ARX*, $n = n_a + n_b$ et pour le modèle statique, $n = 1$.

¹⁵Cet horizon glissant a été choisi pour faciliter la représentation graphique. De manière plus général, $[\theta]$ est une boîte dans l'espace des paramètres en plusieurs dimensions.

entre l'entrée et la sortie de ce modèle dans le cas où l'erreur additive est négligée est comme suit :

$$y_k = b_{1,k}u_{k-1} + b_{2,k}u_{k-2}$$

Les valeurs y_k , u_{k-1} et u_{k-2} sont des observations sur le système et sont donc connues. Si on résout l'équation du modèle par rapport à ses paramètres $b_{1,k}$ et $b_{2,k}$ qui doivent être identifiés, on obtient l'expression suivante :

$$l_k : b_{2,k} = -\frac{u_{k-1}}{u_{k-2}}b_{1,k} + \frac{1}{u_{k-2}}y_k$$

qui détermine une droite l_k dans l'espace des paramètres. Dans le cas où

$$u_{k-2} = 0$$

on obtient :

$$l_k : b_{1,k} = \frac{1}{u_{k-1}}y_k$$

Toutes les valeurs sur la droite l_k sont des solutions des paramètres $b_{1,k}$ et $b_{2,k}$ pour le triplet $\{y_k, u_{k-1}, u_{k-2}\}$. Pour différentes valeurs de k , les triplets $\{y_k, u_{k-1}, u_{k-2}\}$ conduisent à un ensemble de droites dans l'espace $\{b_1, b_2\}$ et chaque droite détermine les solutions possibles de $b_{1,k}$ et $b_{2,k}$ pour le triplet respectif. La sémantique d'identification dit que les bornes des intervalles $[b_1]$ et $[b_2]$ doivent être identifiés telles que :

$$\forall k, \exists b_{1,k} \in [b_1], \exists b_{2,k} \in [b_2] \mid y_k = b_{1,k}u_{k-1} + b_{2,k}u_{k-2}$$

Autrement dit, la boîte $[\theta] = [[b_1] [b_2]]^T$ doit contenir au moins un point de toutes les droites l_k pour différentes valeurs de k .

Dans le cas où l'erreur additive est présente, on a :

$$y_k = b_{1,k}u_{k-1} + b_{2,k}u_{k-2} + e_k$$

La solution de cette équation par rapport aux paramètres $b_{1,k}$ et $b_{2,k}$ est :

$$l_k : b_{2,k} = -\frac{u_{k-1}}{u_{k-2}}b_{1,k} + \frac{1}{u_{k-2}}y_k - \frac{1}{u_{k-2}}e_k \quad (3.3.23)$$

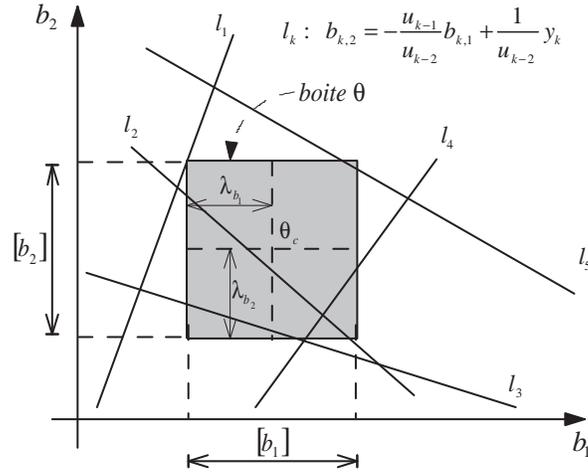


FIG. 3.12 – Espace des paramètres et boîte $[\theta]$.

et si $u_{k-2} = 0$, on a :

$$l_k : b_{1,k} = \frac{1}{u_{k-1}} y_k - \frac{1}{u_{k-1}} e_k$$

En écrivant le support de l'erreur additive sous la forme normalisée :

$$[e] = e_c + \lambda_e \times [-1 \ 1]$$

et en substituant cette forme dans l'équation 3.3.23, pour chaque k , on obtient une bande qui est encadrée par les deux bornes suivantes :

$$\begin{aligned} \bar{l}_k : b_{2,k} &= -\frac{u_{k-1}}{u_{k-2}} b_{1,k} + \frac{1}{u_{k-2}} y_k - \frac{1}{u_{k-2}} e_c + \frac{\lambda_e}{|u_{k-2}|} \\ \underline{l}_k : b_{2,k} &= -\frac{u_{k-1}}{u_{k-2}} b_{1,k} + \frac{1}{u_{k-2}} y_k - \frac{1}{u_{k-2}} e_c - \frac{\lambda_e}{|u_{k-2}|} \end{aligned}$$

qui seront sous la forme

$$\begin{aligned} \bar{l}_k : b_{1,k} &= \frac{1}{u_{k-1}} y_k - \frac{1}{u_{k-1}} e_c + \frac{\lambda_e}{|u_{k-1}|} \\ \underline{l}_k : b_{1,k} &= \frac{1}{u_{k-1}} y_k - \frac{1}{u_{k-1}} e_c - \frac{\lambda_e}{|u_{k-1}|} \end{aligned}$$

pour $u_{k-2} = 0$. En considérant l'équation 3.3.22, la sémantique d'identification pour ce modèle est décrit :

$$\forall k, \exists b_{1,k} \in [b_1], \exists b_{2,k} \in [b_2] \exists e_k \in [e] \mid y_k = b_{1,k}u_{k-1} + b_{2,k}u_{k-2} + e_k$$

Autrement dit, les supports $[\theta]$ et $[e]$ doivent être déterminés tels que la boîte $[\theta]$ a au moins un point en commun avec toutes les bandes pour différentes valeurs de k . L'interprétation géométrique de ce domaine est détaillé dans l'annexe B.

3.3.4 Critère d'optimisation

Dans ce document, le critère d'optimisation des supports $[\theta]$ et $[e]$ des paramètres du modèle intervalle 3.3.20 à minimiser est défini comme la taille de l'enveloppe englobante :

Définition 3.3.1. *La taille de l'enveloppe englobante est définie par le rayon moyen de l'intervalle $[env]_k$ pendant un intervalle de temps de $k = T_1$ à $k = T_2$. Puisque $[env]_k$ est un intervalle, on aura :*

$$\begin{aligned} J &= \frac{1}{T_2 - T_1 + 1} \sum_{k=T_1}^{T_2} \text{rayon}([env]_k) \\ &= \frac{1}{T_2 - T_1 + 1} \sum_{k=T_1}^{T_2} \text{rayon}([\hat{y}]_k + [e]) \\ &= \frac{1}{T_2 - T_1 + 1} \sum_{k=T_1}^{T_2} (\bar{\hat{y}}_k + \bar{e} - \underline{\hat{y}}_k - \underline{e}) \end{aligned} \quad (3.3.24)$$

Une autre définition qui sera utilisée pendant ce travail est la précision d'une enveloppe englobante.

Définition 3.3.2. *La précision d'une enveloppe englobante est la valeur de la fonction exponentiel de $(-J)$ où J est le critère d'optimisation :*

$$P = \exp(-J) \quad (3.3.25)$$

Le choix de la fonction exponentiel de $-J$ est pour borner la variation de J (qui est entre 0 et ∞) dans l'intervalle $[0 \ 1]$. Pour une largeur nulle de l'enveloppe englobante, la précision est égale à l'unité ; c'est-à-dire :

$$J = 0 \Rightarrow P = 1$$

Dans ce cas, les bornes supérieures et inférieures de l'enveloppe englobante sont identiques et puisque la sortie du système doit être incluse dans cette enveloppe, on peut conclure que la sortie du système et la sortie du modèle sont identiques. Un tel modèle

est un modèle déterministe avec une erreur additive nulle qui prédit la sortie précisément. L'autre extrême $P = 0$ aura lieu quand $J \rightarrow \infty$.

La sensibilité plus élevée de $\exp(-J)$ (c'est-à-dire $\frac{\Delta \exp(-J)}{\Delta J}$) pour les petites valeurs de J nous permet d'avoir un indicateur de précision sensible à la variation du rayon de l'enveloppe dans la zone où J se trouve plus souvent. Cela nous permet de partitionner le régime opératoire du système plus précisément en zones locales. Ce partitionnement sera détaillée dans la section 4.5.2.

Le rayon moyen et la précision de l'enveloppe sont deux quantités qui seront utilisées pour mesurer la qualité de l'enveloppe englobante.

3.3.5 Calcul des paramètres d'un modèle intervalle

Compte tenu de la sémantique d'identification qui a été déterminée dans les équations 3.3.21 et 3.3.22, l'enveloppe englobante doit encadrer la sortie pour tous les instants k :

$$\forall k, y_k \in [env]_k = [\theta]^T \times \varphi_k + [e] \quad (3.3.26)$$

De plus, on cherche l'enveloppe la plus petite possible pour la structure de modèle choisie. Cette condition a été formalisée par l'équation 3.3.24.

Par la suite, on va expliquer comment les supports des paramètres du modèle et de l'erreur additive peuvent être identifiés. On appellera brièvement cette procédure, par la suite, *l'identification des paramètres*. Dans un premier temps, on traite le problème quand les observations sont connues avec exactitude. Ensuite, on étudiera le cas plus général où les observations sont connues avec incertitude ; autrement dit lorsqu'elles sont incertaines.

Observations certaines

Dans ce cas, les valeurs de l'entrée et de la sortie observées sont bien connues :

$$\forall k, \quad \begin{aligned} u_k &\in R \\ y_k &\in R \end{aligned}$$

Donc, le vecteur de régression est certain :

$$\varphi_k \in R^{n \times 1}$$

En substituant la forme intervalle normalisée du vecteur des paramètres $[\theta]$ et de l'erreur additive $[e]$ dans l'équation 3.3.26, on obtient :

$$\begin{aligned}
 [env]_k &= [\theta]^T \times \varphi_k + [e] \\
 &= (\theta_c + \lambda_\theta \times [v])^T \times \varphi_k + (e_c + \lambda_e[-1 \ 1]) \\
 &= (\theta_c^T \times \varphi_k + e_c) + ([v]^T \times \lambda_\theta^T \times \varphi_k + \lambda_e \times [-1 \ 1])
 \end{aligned}$$

De l'équation 3.2.11, la borne supérieure \overline{env}_k et la borne inférieure \underline{env}_k de l'intervalle $[env]_k$ sont déduites :

$$\begin{aligned}
 \overline{env}_k &= (\theta_c^T \times \varphi_k + e_c) + (Diag(\lambda_\theta)^T |\varphi_k| + \lambda_e) \\
 \underline{env}_k &= (\theta_c^T \times \varphi_k + e_c) - (Diag(\lambda_\theta)^T |\varphi_k| + \lambda_e)
 \end{aligned} \tag{3.3.27}$$

La condition 3.3.26 impose que :

$$\begin{aligned}
 \forall k, y_k \in [env]_k \Rightarrow \quad &+y_k - \overline{env}_k \leq 0 \\
 &-y_k + \underline{env}_k \leq 0
 \end{aligned} \tag{3.3.28}$$

En substituant les bornes supérieure et inférieure de l'intervalle $[env]_k$ dans l'équation 3.3.28 -la condition d'inclusion de la sortie dans l'enveloppe englobante- et l'équation 3.3.24 -le critère d'optimisation- on aura :

$$\begin{aligned}
 \forall k, \quad &-\lambda_e \leq 0 \\
 &-\lambda_\theta \leq 0 \\
 &+y_k - (\theta_c^T \times \varphi_k + e_c) - (Diag(\lambda_\theta)^T |\varphi_k| + \lambda_e) \leq 0 \\
 &-y_k + (\theta_c^T \times \varphi_k + e_c) - (Diag(\lambda_\theta)^T |\varphi_k| + \lambda_e) \leq 0 \\
 J &= \frac{1}{M} \sum_{k=1}^M (\overline{env}_k - \underline{env}_k) = \frac{2}{M} \sum_{k=1}^M (Diag(\lambda_\theta)^T |\varphi_k| + \lambda_e)
 \end{aligned} \tag{3.3.29}$$

Les deux premières inégalités viennent du fait que le rayon d'un intervalle ne peut pas être une valeur négative. Le système 3.3.29 détermine un problème d'optimisation linéaire du critère J par rapport aux paramètres sous un ensemble de contraintes linéaires par rapport à ces paramètres. Donc, l'identification des supports des paramètres du modèle de l'équation 3.3.20 se traduit par un problème d'optimisation sous contraintes. Puisque le critère à minimiser et les contraintes sont linéaires par rapport aux paramètres du modèle, l'optimum calculé par les méthodes numériques d'optimisation tel le Simplex conduira à un optimum global.

◇

Jusqu'à maintenant, on considérait que les mesures étaient précises, ce qui n'est pas le cas en général en pratique. Dans cette partie, on traite le cas où des éléments du jeu de données sont incertains. La cause de cette incertitude peut être l'imprécision des instruments de mesures, les perturbations de l'environnement sur les instruments, etc. Indépendamment des raisons pour lesquelles les mesures sont incertaines, cette incertitude sera aussi caractérisée par des variables bornées avec un support intervalle.

Pour mieux suivre la démarche de l'identification des paramètres du modèle intervalle en présence d'incertitudes sur les observations, on considère deux cas :

- Incertitude sur la sortie observée
- Incertitude sur l'entrée et la sortie observée

Il faut noter que les travaux qui ont été déjà réalisés pour identifier un modèle intervalle considérant la sémantique d'identification 3.3.21 et 3.3.22 ont toujours traité le cas où les observations sont certaines.

Sortie observée incertaine

Dans ce cas, pendant l'expérience sur le système, la sortie du système a été observée avec à une certaine imprécision. On modélise cette imprécision en définissant un support $[y]_k$ suffisamment grand autour de la valeur observée de manière à ce qu'il contienne la valeur réelle de la sortie du système y_k . En clair, la valeur exacte et inconnue y_k de la sortie se trouve quelque part dans l'intervalle connu $[y]_k$.

$$\forall k, y_k \in [y]_k$$

Par conséquent, le jeu de données devient $DS = \{u_k, [y]_k\}$.

Modèle statique et réponse impulsionnelle. Dans le modèle statique et le modèle réponse impulsionnelle, le vecteur de régression φ_k contient les entrées du système. Puisque dans ce cas, les entrées observées sont certaines, le vecteur de régression est aussi certain :

$$\varphi_k \in R^{n \times 1}$$

Par conséquent, l'équation 3.3.27 est toujours valide pour les bornes supérieure et inférieure de l'intervalle $[env]_k$. La valeur réelle de la sortie du système à l'instant k peut être quelconque dans l'intervalle $[y]_k = [\underline{y}_k, \overline{y}_k]$. Pour garantir que la sortie du système sera expliquée par le modèle, c'est-à-dire incluse dans l'enveloppe englobante, il faut que :

$$\forall k, y_k \in [y]_k \subseteq [env]_k$$

Autrement dit :

$$\forall k, \quad +\overline{y}_k - \overline{env}_k \leq 0$$

$$-\underline{y}_k + \underline{env}_k \leq 0 \quad (3.3.30)$$

En substituant les bornes supérieure et inférieure de l'intervalle $[env]_k$ qui ont été exprimées par l'équation 3.3.27 dans l'équation 3.3.30 -la condition d'inclusion de la sortie dans l'enveloppe englobante- et l'équation 3.3.24 -le critère d'optimisation-, le problème d'identification s'écrit :

$$\begin{aligned} \forall k, \quad & -\lambda_e \leq 0 \\ & -\lambda_\theta \leq 0 \\ & +\bar{y}_k - (\theta_c^T \times \varphi_k + e_c) - (Diag(\lambda_\theta)^T |\varphi_k| + \lambda_e) \leq 0 \\ & -\underline{y}_k + (\theta_c^T \times \varphi_k + e_c) - (Diag(\lambda_\theta)^T |\varphi_k| + \lambda_e) \leq 0 \\ J = & \frac{2}{M} \sum_{k=1}^M (Diag(\lambda_\theta)^T |\varphi_k| + \lambda_e) \end{aligned} \quad (3.3.31)$$

Dans ce problème, les paramètres θ_c , λ_θ , e_c et λ_e doivent être calculés tels qu'ils minimisent J . Le système 3.3.31 est toujours un problème d'optimisation linéaire par rapport aux paramètres sous un ensemble de contraintes linéaires.

Modèle relation entrée-sortie. Dans le modèle relation entré-sortie, le vecteur de régression φ_k contient les entrées et les sorties de système. Puisque dans ce cas, les sorties observées sont incertaines, le vecteur de régression est un vecteur intervalle. Ce vecteur peut être écrit sous la forme normalisée :

$$[\varphi]_k = \varphi_{k,c} + \lambda_{\varphi_k} \times [v]$$

De l'équation 3.2.10 qui détermine le résultat de la multiplication de deux vecteurs intervalles et l'équation 3.2.11 qui exprime les bornes supérieure et inférieure de l'intervalle solution, on obtient une nouvelle expression pour les bornes de $[env]_k$:

$$\begin{aligned} \overline{env}_k &= (\theta_c^T \varphi_{k,c} + e_c) + (Sum(|\theta_c^T| \lambda_{\varphi_k}) + Sum(\lambda_\theta^T |\varphi_{k,c}|) \\ &\quad + Sum(Diag(\lambda_\theta^T \lambda_{\varphi_k})) + \lambda_e) \\ \underline{env}_k &= (\theta_c^T \varphi_{k,c} + e_c) - (Sum(|\theta_c^T| \lambda_{\varphi_k}) + Sum(\lambda_\theta^T |\varphi_{k,c}|) \\ &\quad + Sum(Diag(\lambda_\theta^T \lambda_{\varphi_k})) + \lambda_e) \end{aligned} \quad (3.3.32)$$

Comme l'on a déjà expliqué, la valeur réelle de la sortie du système à l'instant k peut être quelconque dans l'intervalle $[y]_k = [\underline{y}_k, \bar{y}_k]$. Pour garantir que la sortie du système soit incluse dans l'enveloppe englobante, il faut que :

$$\begin{aligned} \forall k, \quad & +\bar{y}_k - \overline{env}_k \leq 0 \\ & -\underline{y}_k + \underline{env}_k \leq 0 \end{aligned} \quad (3.3.33)$$

En substituant les bornes supérieure et inférieure de l'intervalle $[env]_k$ qui ont été calculées dans l'équation 3.3.32 dans l'équation 3.3.33 -la condition d'inclusion de la

sortie dans l'enveloppe englobante- et l'équation 3.3.24 -le critère d'optimisation-, le problème d'identification se traduit par :

$$\begin{aligned}
 \forall k, \quad & -\lambda_e \leq 0 \\
 & -\lambda_\theta \leq 0 \\
 & +\bar{y}_k - \overline{env}_k \leq 0 \\
 & -\underline{y}_k + \underline{env}_k \leq 0 \\
 J &= \frac{2}{M} \sum_{k=1}^M (Sum(|\theta_c^T | \lambda_{\varphi_k}) + Sum(\lambda_\theta^T | \varphi_{k,c}|) \\
 & + Sum(Diag(\lambda_\theta^T \lambda_{\varphi_k})) + \lambda_e)
 \end{aligned} \tag{3.3.34}$$

Comme avant, les paramètres à optimiser sont θ_c , λ_θ , e_c et λ_e . Le problème de l'équation 3.3.34 est un problème d'optimisation non-linéaire sous un ensemble de contraintes non-linéaires par rapport aux paramètres.

Entrée et sortie observées incertaines

Pendant l'expérience sur le système, il arrive que l'entrée et la sortie du système soient connues avec imprécision. Sur le même principe que précédemment, ces observations sont encadrées par des supports adéquats :

$$\begin{aligned}
 \forall k, \quad & u_k \in [u]_k \\
 & y_k \in [y]_k
 \end{aligned}$$

Donc, le jeu de données devient $DS = \{[u]_k, [y]_k\}$.

Puisque toutes les entrées et les sorties dans le jeu de données sont des intervalles, pour toutes les configurations (modèle statique, réponse impulsionnelle à horizon fini ou relation entrée-sortie), le vecteur de régression est un vecteur intervalle. Par conséquent, le problème d'identification du modèle est identique à l'équation 3.3.34 et se traduit par un problème d'optimisation non-linéaire sous un ensemble de contraintes non-linéaires par rapport aux paramètres θ_c , λ_θ , e_c et λ_e .

◇

Le tableau 3.1 montre la nature du critère d'optimisation en fonction du modèle et des grandeurs incertaines.

Méthodes d'optimisation du critère de précision

Dans ce travail, pour résoudre les problèmes d'optimisation 3.3.29, 3.3.31 et 3.3.34 par rapport aux paramètres θ_c , λ_θ , e_c et λ_e , on utilise la fonction $fmincon(\cdot)$ du logiciel

Structure Observation	Modèle statique	Modèle réponse impulsionnelle $n_a = 0, n_b = HG$	Modèle entrée-sortie $n_a = HG, n_b = HG$
entrée et sortie certaine	linéaire	linéaire	linéaire
entrée certaine et sortie incertaine	linéaire	linéaire	non-linéaire
entrée et sortie incertaines	non-linéaire	non-linéaire	non-linéaire

TAB. 3.1 – Caractéristiques du critère d’optimisation en fonction de la structure du modèle et de la présence des incertitudes sur les observations

Matlab. Cette fonction trouve le minimum d’une fonction $F(X)$ à plusieurs variables. Elle résout un problème posé sous la forme suivante :

$$\begin{aligned} \min F(x) \text{ soumise à } & A \times X \leq B, A_{eq} \times X = B_{eq} && \text{contraintes linéaires} \\ & C(X) \leq 0, C_{eq}(X) = 0 && \text{contraintes non-linéaires} \\ & BI \leq X \leq BS && \text{bornes des paramètres} \end{aligned}$$

Dans ce travail, $X = [\theta_c \ \lambda_\theta \ e_c \ \lambda_e]$ et la fonction F est le critère d’optimisation 3.3.24.

La fonction $fmincon(\cdot)$ utilise une des deux méthodes suivantes en fonction du problème d’optimisation sous contraintes à traiter :

- La méthode ”Subspace Trust Region” qui est basée sur la méthode de Newton. Cette méthode est expliquée dans [Coleman & Li, 1996] et [Coleman & Li, 1994].
- La méthode SQP¹⁶. Cette méthode est expliquée dans les travaux de [Fletcher & Powell, 1963] et [Goldfarb, 1970].

La fonction $fmincon(\cdot)$ utilise la première méthode si l’utilisateur fournit le gradient de la fonction à optimiser et *si seulement* les bornes supérieures et inférieures des paramètres sont définis ou *s’il existe seulement* des contraintes égalité. $fmincon(\cdot)$ choisit cette méthode par défaut et si les conditions ci-dessus ne sont pas satisfaites, elle utilise la deuxième méthode. Dans le cas où le critère d’optimisation et les contraintes sont linéaires, cette méthode garantit que la solution est globalement optimale. Le tableau 3.2 montre la complexité du problème d’optimisation pour les différentes structures du modèle. Dans ce tableau, k représente le nombre des éléments dans le jeu de données d’identification.

¹⁶Sequential Quadratic Programming

Modèle	Nombre de paramètres à optimiser	Nombre de contraintes à respecter
Statique	4	$2 \times (k - HG)$
Réponse Impulsionnelle $n_a = 0, n_b = HG$	$2 \times HG + 2$	$2 \times (k - HG)$
Relation entrée-sortie $n_a = HG, n_b = HG$	$4 \times HG + 2$	$2 \times (k - HG)$

TAB. 3.2 – La complexité du problème d’optimisation.

3.3.6 Effet de Mémoire

Une des propriétés des modèles dynamiques est *l’effet de mémoire* qui vient du fait que le vecteur de régression du modèle contient des entrées et des sorties décalées dans le temps sur un horizon fini. En considérant ce vecteur dans les deux configurations suivantes :

$$\begin{aligned}
 & \text{si } n_a = 0, \quad n_b = HG, \quad \varphi_k = [u_{k-1} \dots u_{k-HG}] \\
 & \text{si } n_a = HG, \quad n_b = HG, \quad \varphi_k = [y_{k-1} \dots y_{k-HG} \quad u_{k-1} \dots u_{k-HG}]
 \end{aligned}$$

on constate que pendant la procédure d’identification des paramètres d’un modèle dynamique, les paramètres ne peuvent pas être identifiés avant l’instant $k < HG$ puisque avant cet instant, les données disponibles ne sont pas assez nombreuses pour construire le vecteur de régression du modèle et lier l’entrée et la sortie du système. Ce problème apparaît aussi pendant la simulation d’un modèle dynamique. Au début de la simulation, la sortie de système ne peut pas être prédite avant l’instant $k < HG - 1$ par la structure de modèle choisie. Autrement dit, le modèle n’est pas valide dans l’intervalle de temps :

$$\begin{aligned}
 & k = 1 \text{ à } k = HG \quad \text{en identification} \\
 & k = 1 \text{ à } k = HG - 1 \quad \text{en simulation}
 \end{aligned}$$

Pour diminuer ce problème dû à l’effet de mémoire, une solution est de réduire l’horizon glissant du modèle. Néanmoins, cette solution n’est pas convenable en général puisqu’en réduisant cet horizon, la qualité du modèle diminue et l’enveloppe englobante sera une enveloppe plus large. Inversement, pour obtenir une enveloppe précise, on doit choisir l’horizon glissant en fonction du comportement du système et plus long est l’horizon glissant, plus le nombre des données nécessaires est important. Il faut aussi

se rappeler que même pour un modèle dynamique avec un horizon minimal $HG = 1$, l'effet de mémoire est toujours présent. Néanmoins, pour un modèle statique, puisqu'il n'utilise pas des entrées et des sorties décalées dans le vecteur de régression, il est valide pour tous les instants.

Pour avoir un modèle qui soit valide partout, on propose dans ce document d'utiliser un banc de modèles constitué d'un modèle statique et d'un ensemble de modèles dynamiques dont les horizons glissants varient de $HG = 1$ à $HG = N$. Le modèle statique et les modèles dynamiques s'appellent des *modèles auxiliaires*, le modèle qui contient les modèles auxiliaires s'appelle le *modèle F*. N représente l'horizon glissant nécessaire pour représenter le comportement du système et il est choisi en fonction de la dynamique du système. Il s'agit de l'ordre du modèle que l'on aurait choisi classiquement si on travaillait sur un unique modèle à la place d'un banc de modèles. Si les modèles dynamiques auxiliaires sont des modèles de types réponse impulsionnelle à horizon fini, on appellera cette structure *modèle FFIR* et si ce sont des modèles de type relation entrée-sortie, on l'appellera *modèle FIO*. La structure d'un modèle F est montrée sur la figure 3.13.

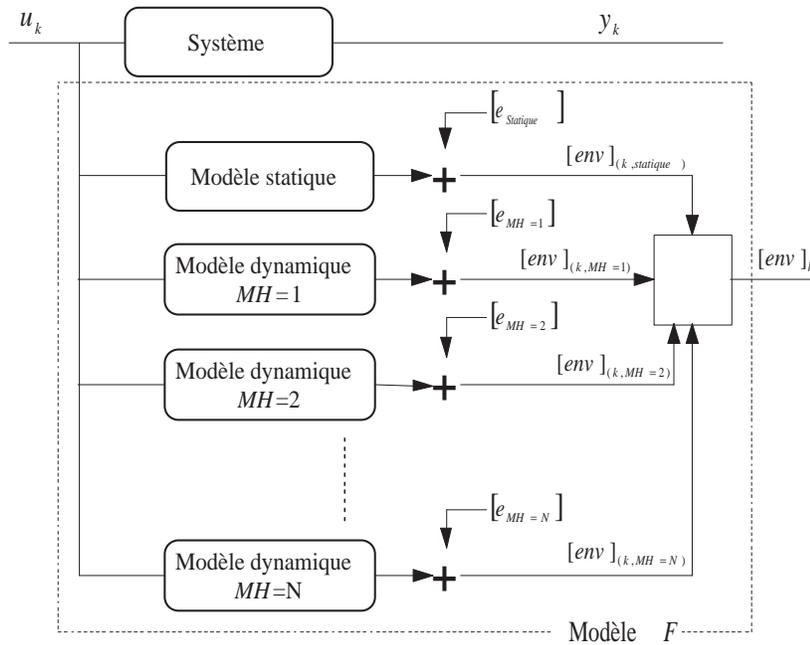


FIG. 3.13 – Structure d'un modèle F .

Les supports des paramètres de tous les modèles auxiliaires et de leurs erreurs additives respectives seront identifiés individuellement en utilisant la méthode d'identification présentée précédemment. Dans ce cas et pour tous les modèles auxiliaires, on aura :

$$\forall k > HG, y_k \in [env]_{(k,HG)} = [\hat{y}]_{(k,HG)} + [e_{HG}], \quad HG \in \{0, 1, 2, \dots, N\} \quad (3.3.35)$$

La condition $k > HG$ détermine le domaine de validité d'un modèle auxiliaire HG

en identification, dans le sens où un modèle auxiliaire est valide qu'à partir du moment où on dispose de suffisamment de données. Puisque l'énumération dans le temps discret commence par $k = 1$, l'inégalité est stricte. Par exemple, le modèle dynamique avec $HG = 1$ est valide qu'à partir de $k > 1$. Par contre, pour un modèle statique, l'horizon glissant est nul et peut être représenté par $HG = 0$. Ce modèle est valide pour toutes les valeurs de $k > 0$.

De l'équation 3.3.35, on peut conclure qu'à l'instant k , la sortie du système est aussi englobée par l'intersection des domaines $[env]_{(k,HG)}$ calculés. Si $k \leq N$, on aura :

$$\forall k \leq N, y_k \in [env]_k = \bigcap_{HG=0}^k [env]_{(k,HG)}, \quad HG \in \{0, 1, 2, \dots, N\} \quad (3.3.36)$$

A partir de l'instant $k > N$, tous les modèles auxiliaires sont valides. Donc :

$$\forall k > N, y_k \in [env]_k = \bigcap_{HG=0}^N [env]_{(k,HG)}, \quad HG \in \{0, 1, 2, \dots, N\} \quad (3.3.37)$$

L'intérêt de procéder à l'intersection des enveloppes est d'utiliser le modèle auxiliaire valide de plus grand ordre pour gagner en précision.

Dans un modèle mono-mode, l'effet de mémoire n'est pas gênant puisqu'une fois l'horizon glissant du modèle passé, les données sont assez nombreuses pour identifier ou simuler le modèle. Cependant, dans un modèle multi-mode qui commute fréquemment d'un mode à un autre, cet effet nous empêche de calculer l'enveloppe de la sortie HG instants après chaque commutation. Le modèle F est plutôt utilisé dans le contexte multi-mode. Le modèle F proposé dans ce document nous aide au moins à englober la sortie de système par une enveloppe pendant le passage de l'horizon glissant du modèle même si cette enveloppe n'est pas optimale. Le problème de non-validité des modèles dynamiques pendant le passage de l'horizon glissant du modèle a été aussi traité dans [Shahriari et al., 2006] et la solution proposée ci-dessus a été utilisée pour un système multi-mode dans le cas où la commutation est déterminée.

Définition 3.3.3. *L'horizon glissant d'un modèle F est défini comme l'ordre le plus grand des modèles auxiliaires ; c'est-à-dire N .*

Remarque. Si le jeu de données qui est utilisé pour identifier les paramètres des modèles auxiliaires n'est pas assez riche au niveau des informations sur le comportement du système, les paramètres ne seront pas calculés correctement. La conséquence de ce fait est que pendant la simulation¹⁷, il peut arriver des instants pour lesquels l'intersection des domaines $[env_{HG}]_k$ des modèles auxiliaires soit vide et donc, on n'arrive plus à calculer l'enveloppe englobante de la sortie du système à ces instants. Ce phénomène sera illustré dans l'exemple 3.4.3 ◇

¹⁷On étudiera la simulation d'un modèle pour prédire l'enveloppe englobante de la sortie du système dans la section 3.4.

3.3.7 Exemple Numérique

On illustre cette partie par deux exemples numériques. L'objectif du premier exemple est de représenter la dynamique d'un système par un modèle relation entrée-sortie à intervalle. Le deuxième exemple traite le problème de non-validité du modèle dynamique en début d'identification et de simulation pour le même système en utilisant un modèle *FFIR*.

Exemple 3.3.1

L'objectif de cet exemple est de représenter le comportement d'un système par un modèle intervalle. Le système à modéliser est un système du deuxième ordre. On suit les étapes d'identification en considérant la boucle d'identification de la figure 3.7.

Système "physique" à modéliser. Les observations sur le système sont recueillies en utilisant le modèle déterministe suivant :

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned}$$

où

$$A = \begin{bmatrix} -0,1275 & 0,3165 \\ -1,2662 & -0,3807 \end{bmatrix}, B = \begin{bmatrix} 1,1275 \\ 1,2662 \end{bmatrix}, C = [1 \ 0] \quad (3.3.38)$$

Jeu de données. Pour obtenir le jeu de données, on conçoit une expérience sur le système pour laquelle l'entrée varie et on observe les sorties du système. Le signal d'entrée est un signal SBPA¹⁸. En pratique, la sortie y_k est inconnue et on observe $y_k + \epsilon_k$ où ϵ_k représente l'erreur de mesure. Dans cet exemple, l'erreur ϵ_k est une variable aléatoire bornée et on suppose qu'on connaît a priori son domaine de variation qui est entre -0,1 et 0,1. Donc, on encadre la valeur observée $y_k + \epsilon_k$ par l'intervalle $[-0,1 \ 0,1]$ pour garantir que la sortie réelle y_k est prise en compte dans les calculs. Cet intervalle est ensuite utilisé comme la sortie du système dans le jeu de données d'identification *DS* :

$$[y]_k \leftarrow y_k + \epsilon_k + [-0,1 \ 0,1]$$

On fait de même pour simuler l'erreur de mesure et la perturbation sur l'entrée et on obtient :

$$[u]_k = u_k + \epsilon'_k + [-0,01 \ 0,01]$$

Donc, le jeu de données d'identification obtenu est de la forme $DS = \{[u]_k, [y]_k\}$ qui est le cas le plus général traité dans la section précédente.

¹⁸Signal Binaire Pseudo Aléatoire

Structure de modèle. La structure choisie est une structure *ARX* de la forme A.0.1 :

$$y_k = \theta_k^T \times \varphi_k + e_k$$

Pour prendre en compte l'incertitude paramétrique, on utilise le modèle *ARX* intervalle avec le vecteur de paramètres $[\theta]$ et l'erreur additive $[e]$. En considérant l'information a priori sur le système, on sait que le comportement du système peut être représenté par un modèle relation entrée-sortie du deuxième ordre. Donc, on choisit l'ordre du modèle comme suit :

$$n_a = 2, n_b = 2$$

Sémantique d'identification. On utilise la sémantique d'identification définie dans 3.3.21 et 3.3.22. Les supports du modèle intervalle doivent être identifiés tels que :

$$\forall k, y_k \in [env]_k$$

Puisque la sortie observée est incertaine, l'égalité doit être satisfaite pour toutes les valeurs de $y_k \in [y]_k$:

$$\forall k, [y]_k \subseteq [env]_k$$

Calcul des paramètres du modèle intervalle. Les entrées et les sorties observées sont incertaines. Donc, le problème d'identification se traduit par le problème d'optimisation 3.3.34. A l'aide de la fonction *fmincon*(.), θ_c , λ_c , e_c et λ_e sont calculés tels qu'ils optimisent le critère d'optimisation. Après les avoir calculés, on obtient les valeurs suivantes pour les supports :

$$[e] = [-0.0421, 0.0903], e_c = 0,0241 \lambda_e = 0,0663$$

$$[\theta] = \begin{bmatrix} [-0,4670 & -0,4670] \\ [-0,4481 & -0,4481] \\ [1,0928 & 1,1109] \\ [0,8022 & 0,8022] \end{bmatrix}, \theta_c = \begin{bmatrix} -0,4670 \\ -0,4482 \\ 1,1019 \\ 0,8023 \end{bmatrix}, \lambda_\theta = \begin{bmatrix} 0,0000 \\ 0,0000 \\ 0,0090 \\ 0,0000 \end{bmatrix}$$

Les rayons d'incertitudes sur λ_θ et λ_e s'adaptent au mieux pour expliquer les observations en minimisant le critère d'optimisation. Dans cet exemple, l'incertitude sur θ est presque nulle et l'incertitude paramétrique est expliquée par le terme e .

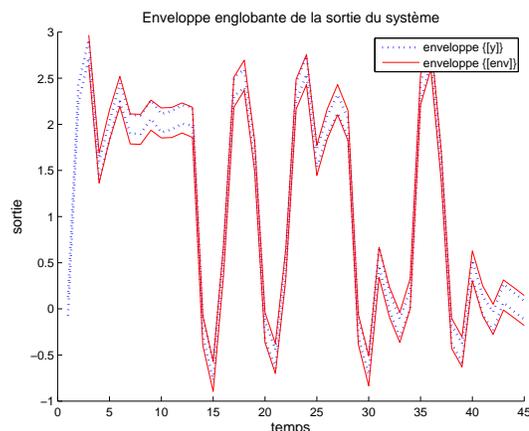


FIG. 3.14 – Sortie observée et enveloppe englobante pour un modèle relation entrée-sortie du deuxième ordre.

Pour calculer le domaine $[env]_k$ à l'instant k par le prédicteur à un pas 3.3.20, le jeu de données DS est utilisé. La sortie du système ainsi que l'enveloppe englobante $\{[env]\}$ sont montrées sur la figure 3.14. Puisqu'on a utilisé un modèle du deuxième ordre et à cause de l'effet de mémoire, l'enveloppe ne peut pas être calculée avant l'instant $k < 2$.

Pour mesurer la qualité de l'enveloppe, on avait défini deux indicateurs ; le rayon moyen de l'enveloppe qui est aussi le critère d'optimisation (équation 3.3.24) et la précision de l'enveloppe (équation 3.3.25). Ces deux indicateurs pour l'enveloppe de cet exemple valent :

$$\begin{aligned} J &= 0,2292 \\ P &= 0,7952 \end{aligned}$$

Pour montrer l'effet d'un mauvais choix de la structure de modèle sur l'incertitude paramétrique du modèle intervalle, on identifie le même système par un modèle ARX du première ordre $-n_a = 1$ et $n_b = 1$. On a :

$$\begin{aligned} [e] &= [-0,8892 \ 0,3659], \quad e_c = -0,2617, \quad \lambda_e = 0,6276 \\ [\theta] &= \begin{bmatrix} [-0,2859 \ -0,2859] \\ [1,5404 \ 1,5404] \end{bmatrix}, \quad \theta_c = \begin{bmatrix} -0,2859 \\ 1,5404 \end{bmatrix}, \quad \lambda_\theta = \begin{bmatrix} 0,0000 \\ 0,0000 \end{bmatrix} \end{aligned}$$

On constate une augmentation de l'incertitude sur l'erreur additive par rapport au modèle du deuxième ordre. On utilise le prédicteur à un pas de l'équation 3.3.20 et le jeu de données DS pour calculer l'enveloppe $\{[env]\}$. Cette enveloppe est montrée sur la figure 3.15. Les indicateurs de la qualité de l'enveloppe valent :

$$J = 1,2403$$

$$P = 0,2893$$

L'imprécision de l'enveloppe englobante sur la figure 3.15 par rapport à celle de la figure 3.14 est due à une structure de modèle volontairement mal adaptée. Cela montre l'effet de ce mauvais choix de la structure de modèle sur l'incertitude des paramètres et que la première structure est plus adaptée pour représenter le comportement du système que la seconde structure.

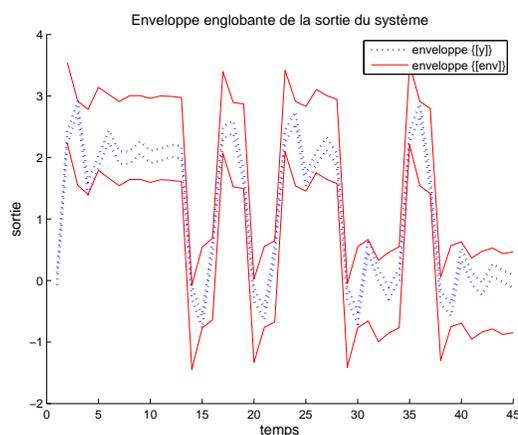


FIG. 3.15 – Sortie observée et enveloppe englobante pour un modèle relation entrée-sortie du première ordre.

Exemple 3.3.2

L'objectif de cet exemple est de valider la méthode pour surmonter le problème lié à l'effet de mémoire proposée dans la section 3.3.6. On utilise le même système et le même jeu de données que pour l'exemple 3.3.1. La structure choisie est une structure *ARX*. Concernant l'information a priori sur le système, le comportement du système peut être représentée par un modèle *FIR* d'ordre 10 ; c'est à dire :

$$n_a = 0, n_b = 10$$

Donc, le modèle F consistera en 11 modèles auxiliaires : un modèle statique et 10 modèles *FIR* dont les horizons glissants varient de $HG = 1$ à $HG = 10$. Puisque les entrées et les sorties observées sont incertaines, le problème d'identification du modèle F se traduit par le problème d'optimisation 3.3.34 appliqué sur chaque modèle auxiliaire. Après avoir identifié les modèles auxiliaires individuellement, les enveloppes englobantes de tous les modèles auxiliaires sont calculées par le prédicteur à un pas de l'équation 3.3.20 en

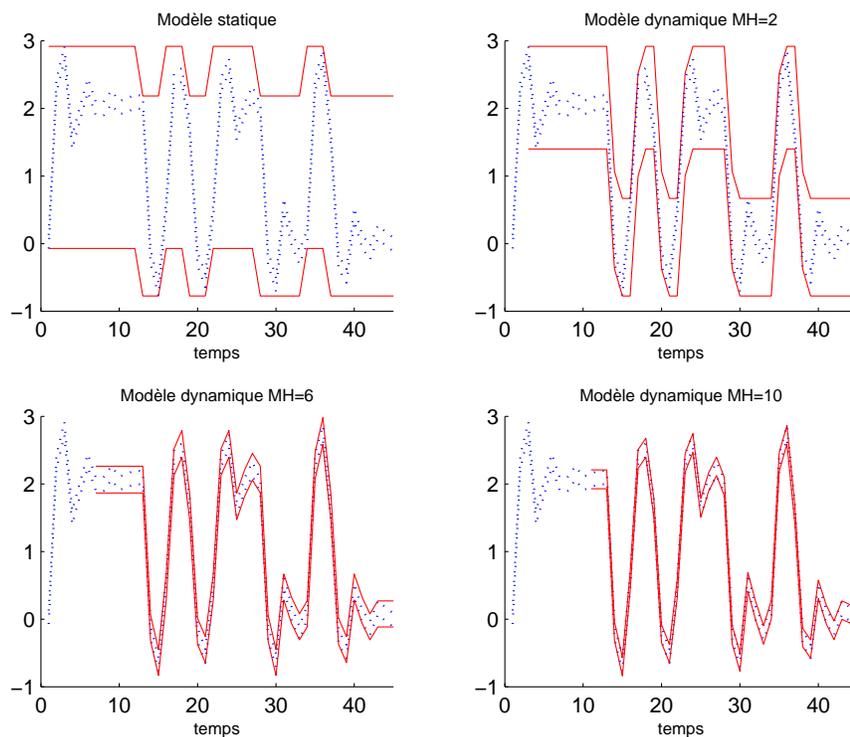


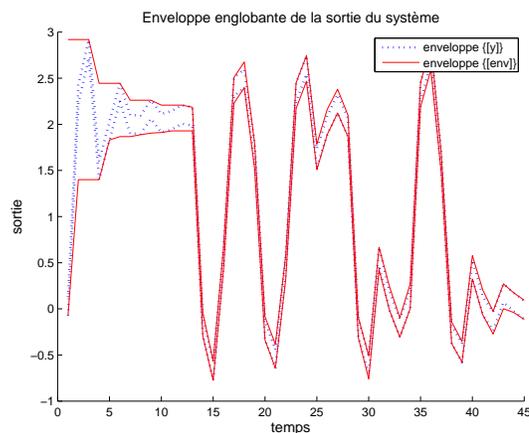
FIG. 3.16 – Sortie observée et enveloppes englobantes des modèles auxiliaires pour une structure *FFIR* d’ordre 10

utilisant le jeu de données *DS*. Certaines d’entre elles ont été sélectionnées pour être montrées sur la figure 3.16.

L’intersection de ces enveloppes détermine l’enveloppe de la sortie du modèle *FFIR* qui est montrée sur la figure 3.17. Sur cette figure, on voit que le rayon de l’enveloppe du modèle *FFIR* se réduit dans le temps pendant le passage de l’horizon glissant. Plus particulièrement, on constate une enveloppe très large à l’instant $k = 1$, car on est dans le pire de cas ; celui d’un régime transitoire expliqué par un modèle statique.

Les supports des paramètres du modèle dynamique avec l’horizon glissant $HG = 10$ sont donnés ci-dessous. Les incertitudes sur les paramètres et l’erreur additive sont faibles et elles sont comparables avec celles sur les paramètres du modèle relation entrée-sortie d’ordre deux.

$$[e] = [-0,0596 \ 0,0980], \quad e_c = 0,0192, \quad \lambda_e = 0,0788$$


 FIG. 3.17 – Sortie observée et enveloppe englobante du modèle *FFIR*.

$$\theta = \begin{bmatrix} [1, 1165 \ 1, 1267] \\ [0, 2520 \ 0, 2520] \\ [-0, 6337 \ -0, 6337] \\ [0, 1996 \ 0, 2126] \\ [0, 1660 \ 0, 1660] \\ [-0, 1765 \ -0, 1754] \\ [0, 0285 \ 0, 0285] \\ [0, 0402 \ 0, 0402] \\ [-0, 0188 \ -0, 0188] \\ [-0, 0016 \ -0, 0016] \end{bmatrix}, \quad \theta_c = \begin{bmatrix} 1, 1216 \\ 0, 2521 \\ -0, 6338 \\ 0, 2062 \\ 0, 1661 \\ -0, 1760 \\ 0, 0286 \\ 0, 0402 \\ -0, 0188 \\ -0, 0016 \end{bmatrix}, \quad \lambda_\theta = \begin{bmatrix} 0, 0051 \\ 0, 0000 \\ 0, 0000 \\ 0, 0065 \\ 0, 0000 \\ 0, 0005 \\ 0, 0000 \\ 0, 0000 \\ 0, 0000 \\ 0, 0000 \end{bmatrix}$$

Les indicateurs de la qualité de l'enveloppe calculée sont comme suit :

$$J = 0,3174$$

$$P = 0,7280$$

Ces indicateurs montrent une enveloppe moins bonne que l'exemple précédent. Cela est dû au démarrage difficile. Si on ignore les 10 premiers instants et si on re-calcule les indicateurs de la qualité de l'enveloppe pour les restants, on obtient les valeurs suivantes :

$$J = 0,2205$$

$$P = 0,8021$$

ce qui correspond à une qualité presque identique à celle de l'enveloppe de l'exemple 3.3.1. Cela montre que la structure du modèle est bien choisie et le modèle représente correctement le comportement du système.

3.3.8 Modèle intervalle et paire d'inégalités

Au premier regard, calculer l'enveloppe de la sortie du système par un modèle intervalle paraît similaire au fait d'encadrer la sortie du système par deux bornes \hat{y}_1 et \hat{y}_2 (en statique, deux droites) où la première détermine la borne supérieure de la sortie et la deuxième détermine la borne inférieure ; c'est-à-dire :

$$\hat{y}_2 < y_k < \hat{y}_1$$

L'idée était d'encadrer au mieux le jeu de données à traiter, il faut optimiser les 4 paramètres de ces 2 droites afin que les distances entre les points et les droites frontières soient les plus faibles possibles. Cependant, ce sont deux approches différentes. En utilisant un modèle intervalle, les bornes supérieure et inférieure de l'enveloppe englobante ne sont pas deux droites indépendantes. On le constate dans les équations 3.3.27 et 3.3.32 qui déterminent ces deux bornes. Pour $\varphi = 0$, la distance entre les droites est minimale et est égale à $2 \times \text{rayon}([e])$. Dans cette approche, l'espace de sortie est divisé en trois sous-espaces : au-dessus de la borne supérieure, au-dessous de la borne inférieure et le sous-espace entre ces deux bornes qui détermine, dans notre cas, l'enveloppe de la sortie du système. Cette enveloppe est montrée sur la figure 3.18 pour un ensemble des points.

Contrairement à l'approche ensembliste, si on encadre les sorties du système par une paire de droites, celles-ci peuvent s'intersecter en un point. Dans cette approche, l'espace de sortie est divisé en quatre sous-espaces P_1 , P_2 , P_3 et P_4 :

$$\begin{aligned} P_1 &= \{p \mid p < \hat{y}_1 \wedge p < \hat{y}_2\} \\ P_2 &= \{p \mid p < \hat{y}_1 \wedge p > \hat{y}_2\} \\ P_3 &= \{p \mid p > \hat{y}_1 \wedge p < \hat{y}_2\} \\ P_4 &= \{p \mid p > \hat{y}_1 \wedge p > \hat{y}_2\} \end{aligned}$$

où la sortie du système doit se trouver dans le sous-espace P_2 . Le résultat pour l'encadrement du même ensemble de points que la figure 3.18 par la deuxième approche est montré sur la figure 3.19. En utilisant cette approche, les bornes peuvent s'intersecter et le domaine de solution se limite à seul sous-espace de l'espace de sortie. Par l'approche ensembliste, l'enveloppe calculée est un véritable tube qui correspond mieux à l'objectif de ce travail.

3.4 Simulation de Modèles Intervalle

Après avoir calculé les paramètres d'un modèle intervalle, ce modèle peut être utilisé dans diverses applications ; par exemple le diagnostic ou la détection de défauts. Ces deux applications se situent dans le cas où pendant la simulation, on a accès aux sorties du système et on peut utiliser le prédicteur à un pas de l'équation 3.3.20 pour calculer

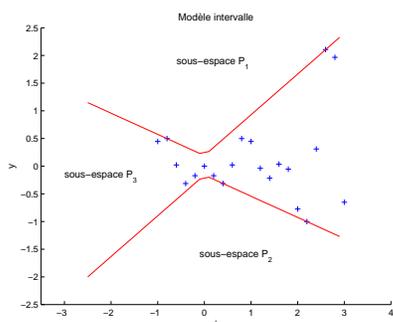


FIG. 3.18 – Modèle intervalle.

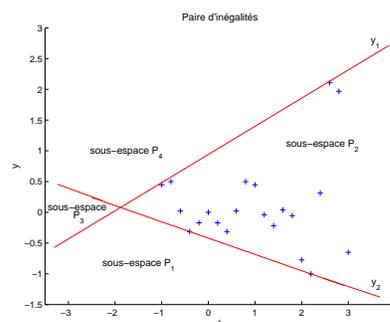


FIG. 3.19 – Paire de droites.

l'enveloppe englobante de la sortie du système puis tester par exemple si la sortie mesurée appartient bien à la sortie intervalle estimée.

Une autre application pour un modèle intervalle est l'analyse de sûreté. L'objectif de cette analyse est de prédire l'enveloppe englobante des variables caractérisant l'état du système sur un horizon de prédiction pour une séquence des entrées qui est donnée par $U = \{[u]_1 [u]_2 \dots [u]_{ph}\}$. ph s'appellera dans la suite *l'horizon de prédiction*. Dans ce document, la variable qu'on cherche à englober est la sortie du système. L'analyse de sûreté est une analyse pour laquelle on n'a pas accès aux sorties du système pendant la simulation. Dans ce cas, pour calculer l'enveloppe englobante de la sortie, on propose d'utiliser un simulateur de modèles intervalle. Ce simulateur doit calculer l'enveloppe englobante de la sortie sur l'horizon ph et il sera conçu par la suite. En considérant la figure 2.7, ce simulateur consiste en trois parties :

Modèle intervalle. On a déjà expliqué dans la section précédente comment on peut identifier un modèle F intervalle à partir des observations de l'entrée et de la sortie du système. Le modèle intervalle est utilisé pour prendre en compte l'incertitude paramétrique du modèle et la structure F est utilisée pour surmonter le problème lié à l'effet de mémoire.

Sémantique de simulation. La sémantique de simulation détermine les propriétés de l'enveloppe englobante. Comme la sémantique d'identification, elle est composée des quantificateurs \forall , \exists , et \neg sur les paramètres et les variables du modèle. Elle est définie en fonction de l'objectif de simulation.

Méthode numérique pour calculer l'enveloppe. Après avoir défini la sémantique de simulation, l'enveloppe englobante doit être calculée en fonction de cette sémantique par une méthode convenable. Cette enveloppe peut être parfois déterminée analytiquement en précisant ses bornes supérieure et inférieure -ce qui est le cas de ce travail- ou par des méthodes numériques comme le pavage ou la méthode de Monte-Carlo.

Dans la suite, on va définir la sémantique de simulation concernant l'objectif de

l'analyse de sûreté ainsi que la méthode pour calculer l'enveloppe englobante de la sortie du système sur l'horizon de prédiction ph .

3.4.1 Sémantique de Simulation

Puisque dans l'objectif de l'analyse de sûreté, on souhaite que l'enveloppe englobante inclut toutes les valeurs possibles de la sortie du système (englobe tous les comportements possibles du système), pour le modèle auxiliaire avec l'horizon glissant $HG \in \{0, 1, \dots, N\}$, la sémantique de simulation est définie comme suit :

$$[P.env]_{(k,HG)} = \{\hat{y}_{(k,HG)} + e_{(k,HG)} \mid \forall \theta_{(k,HG)} \in [\theta_{HG}], \forall e_{(k,HG)} \in [e_{HG}]\} \quad (3.4.1)$$

où

$$\hat{y}_{(k,HG)} = \theta_{(k,HG)}^T \times \varphi_{(k,HG)}$$

On utilise la notation $[P.env]_k$ pour faire la distinction entre l'enveloppe englobante $[env]_k$ à l'instant k correspondant à une analyse pour laquelle on a accès aux sorties du système et l'enveloppe englobante $[P.env]_k$ à l'instant k correspondant à une analyse pour laquelle on n'a pas accès aux sorties observées.

Dans le cas où les éléments du vecteur de régression sont des intervalles ($[\varphi_{HG}]_k$ au lieu de $\varphi_{k,HG}$), l'enveloppe doit être calculée pour toutes les valeurs possibles de ce vecteur :

$$[P.env]_{(k,HG)} = \{\hat{y}_{(k,HG)} + e_{(k,HG)} \mid \forall \varphi_{(k,HG)} \in [\varphi_{HG}]_k, \forall \theta_{(k,HG)} \in [\theta_{HG}], \forall e_{(k,HG)} \in [e_{HG}]\} \quad (3.4.2)$$

où

$$\hat{y}_{(k,HG)} = \theta_{(k,HG)}^T \times \varphi_{(k,HG)}$$

Les expressions 3.4.1 et 3.4.2 déterminent la sémantique de simulation dans le contexte d'une analyse de la sûreté.

3.4.2 Méthode numérique pour calculer l'enveloppe

En considérant l'équation 3.2.1 qui définit l'image d'une paire d'intervalles par rapport aux opérations mathématiques et les expressions des équations 3.4.1 et 3.4.2 qui déterminent l'intervalle $[P.env]_{(k,HG)}$ à l'instant k , on constate que la sémantique des

opérations mathématiques sur des intervalles est identique avec la sémantique de simulation. Donc, le domaine $[P.env]_{(k,HG)}$ peut être calculé directement par :

$$[P.env]_{(k,HG)} = [\theta_{HG}]^T \times [\varphi_{HG}]_k + [e_{HG}]$$

En substituant les formes normalisées des vecteurs $[\theta_{HG}]$, $[\varphi_{HG}]_k$ et $[e_{HG}]$, on obtient :

$$\begin{aligned} [P.env]_{(k,HG)} &= (\theta_{(HG,c)} + \lambda_{\theta_{HG}} \times [v])^T \times (\varphi_{(k,HG,c)} + \lambda_{\varphi_{(k,HG)}} \times [v]) \\ &+ (e_{c_{HG}} + \lambda_{e_{HG}} \times [-1 \ 1]) \end{aligned} \quad (3.4.3)$$

où

$$[v]_{HG \times 1} = [[-1 \ 1] \dots [-1 \ 1]]^T$$

et les bornes supérieure et inférieure de $[P.env]_{(k,HG)}$ sont :

$$\begin{aligned} \overline{P.env}_{(k,HG)} &= (\theta_{(HG,c)}^T \varphi_{(k,HG,c)} + e_{(HG,c)}) + (Sum(|\theta_{(HG,c)}^T| \lambda_{\varphi_{(k,HG)}}) \\ &+ Sum(\lambda_{\theta_{HG}}^T |\varphi_{(k,HG,c)}|) + Sum(Diag(\lambda_{\theta_{HG}}^T \lambda_{\varphi_{(k,HG)}})) + \lambda_{e_{HG}}) \\ \underline{P.env}_{(k,HG)} &= (\theta_{(HG,c)}^T \varphi_{(k,HG,c)} + e_{(HG,c)}) - (Sum(|\theta_{(HG,c)}^T| \lambda_{\varphi_{(k,HG)}}) \\ &+ Sum(\lambda_{\theta_{HG}}^T |\varphi_{(k,HG,c)}|) + Sum(Diag(\lambda_{\theta_{HG}}^T \lambda_{\varphi_{(k,HG)}})) + \lambda_{e_{HG}}) \end{aligned} \quad (3.4.4)$$

L'équation 3.4.4 détermine analytiquement l'intervalle $[P.env]_{(k,HG)}$ en fonction des paramètres du modèle $\theta_{(HG,c)}$, $\lambda_{\theta_{HG}}$, $e_{(HG,c)}$ et $\lambda_{e_{HG}}$, et le vecteur de régression $\varphi_{(k,HG,c)}$ et $\lambda_{\varphi_{(k,HG)}}$. L'enveloppe englobante finale du modèle F à l'instant k est déterminée par les équations issues de 3.3.36 et 3.3.37 :

$$\forall k \leq N, [P.env]_k = \bigcap_{HG=0}^k [P.env]_{(k,HG)} \quad HG \in \{0, 1, 2, \dots, N\}$$

$$\forall k > N, [P.env]_k = \bigcap_{HG=0}^N [P.env]_{(k,HG)} \quad HG \in \{0, 1, 2, \dots, N\}$$

Pour une étude plus approfondie, deux configurations *FIO* et *FFIR* de modèle F seront analysées séparément. Mais avant, on présente le théorème suivant pour un modèle relation entrée-sortie :

Théorème 3.4.1. *On suppose que les supports $[\theta]$ et $[e]$ d'un modèle relation entrée-sortie intervalle sont identifiés en considérant la sémantique d'identification des équations 3.3.21 et 3.3.22. Le modèle est un modèle d'ordre HG ; c'est-à-dire :*

$$n_a = HG, \quad n_b = HG$$

A l'instant k , le vecteur de régression est de la forme suivante :

$$[\varphi]_k = [-[y]_{k-1} \dots -[y]_{k-n_a} \quad [u]_{k-1} \dots [u]_{k-n_b}]$$

Dans l'objectif d'une prédiction de l'enveloppe englobante, on substitue, à une ou plusieurs sorties observées $[y]_{k-i}$ auxquelles on n'a pas accès, les domaines prédits $[P.env]_{k-i}$ respectifs et on obtient un nouveau vecteur de régression $[\varphi]'_k$. Si

$$\forall i < k, [y]_{k-i} \subseteq [P.env]_{k-i}$$

dans ce cas :

$$\forall k, [y]_k \subseteq [P.env]_k$$

où

$$[P.env]_k = [\theta]^T \times [\varphi]'_k + [e]$$

Autrement dit, la sortie du système reste incluse dans le domaine $[P.env]_k$ en remplaçant les sorties mesurées du régresseur par les sorties intervalle estimées.

Démonstration. A l'instant k , les intervalles $[P.env]_{k-i}$ sont déterminés tels que :

$$\forall i < k, [y]_{k-i} \subseteq [P.env]_{k-i}$$

Par conséquent, si une ou plusieurs sorties observées $[y]_{k-i}$ du vecteur de régression $[\varphi]_k$ sont remplacées par les domaines prédits $[P.env]_{k-i}$ respectifs, $[\varphi]_k$ sera un sous-ensemble du nouveau vecteur de régression $[\varphi]'_k$:

$$[\varphi]_k \subseteq [\varphi]'_k$$

La fonction $[env]_k = [\theta]^T \times [\varphi]_k + [e]$ est une fonction monotone par inclusion. A partir de la définition 3.2.5, on a :

$$[\varphi]_k \subseteq [\varphi]'_k \Rightarrow [env]_k \subseteq [P.env]_k \quad (3.4.5)$$

De plus, les paramètres $[\theta]$ et $[e]$ sont identifiés tels que :

$$\forall k, [y]_k \subseteq [env]_k \quad (3.4.6)$$

Des équations 3.4.5 et 3.4.6, on conclut que :

$$[y]_k \subseteq [P.env]_k \quad (3.4.7)$$

□

3.4.3 Prédiction de l'enveloppe englobante : Modèle *FFIR*

On suppose que le modèle *FFIR* est déjà identifié. La séquence des entrées dont on souhaite calculer l'enveloppe englobante est donnée par $U = \{[u]_1 [u]_2 \dots [u]_{ph}\}$ et sont différente de celle utilisée pour identifier les paramètres du modèle. La prédiction commence à l'instant $k = 1$. A cet instant, seul le modèle auxiliaire statique est valide. Donc, la prédiction se base sur la sortie du modèle statique :

$$\begin{aligned} [P.env]_{(1,0)} &= [\theta_0]^T \times [\varphi_0]_1 + [e_0] \\ [P.env]_1 &= [P.env]_{(1,0)} \end{aligned}$$

où

$$[\theta_0] = [b_{HG=0}], \quad [\varphi_0]_1 = [u]_1$$

A l'instant $k = 2$, puisque le vecteur de régression $[\varphi_1]_2 = [u]_1$ est disponible, le premier modèle dynamique $HG = 1$ est aussi valide. Par conséquent :

$$\begin{aligned} [P.env]_{(2,0)} &= [\theta_0]^T \times [\varphi_0]_2 + [e_0] \\ [P.env]_{(2,1)} &= [\theta_1]^T \times [\varphi_1]_2 + [e_1] \\ [P.env]_2 &= [P.env]_{(2,0)} \cap [P.env]_{(2,1)} \end{aligned}$$

où

$$\begin{aligned} [\theta_0] &= [b_{HG=0}] & [\varphi_0]_2 &= [u]_2 \\ [\theta_1] &= [b_{HG=1}] & [\varphi_1]_2 &= [u]_1 \end{aligned}$$

et à l'instant $k = 3$:

$$\begin{aligned}
 [P.env]_{(3,0)} &= [\theta_0]^T \times [\varphi_0]_3 + [e_0] \\
 [P.env]_{(3,1)} &= [\theta_1]^T \times [\varphi_1]_3 + [e_1] \\
 [P.env]_{(3,2)} &= [\theta_2]^T \times [\varphi_2]_3 + [e_2] \\
 [P.env]_3 &= [P.env]_{(3,0)} \bigcap [P.env]_{(3,1)} \bigcap [P.env]_{(3,2)}
 \end{aligned}$$

où

$$\begin{aligned}
 [\theta_0] &= [b_{HG=0}] & [\varphi_0]_3 &= [u]_3 \\
 [\theta_1] &= [b_{HG=1}] & [\varphi_1]_3 &= [u]_2 \\
 [\theta_2] &= [[b_{1,HG=2}] [b_{2,HG=2}]]^T & [\varphi_2]_3 &= [[u]_2 [u]_1]^T
 \end{aligned}$$

A chaque instant, un nouveau modèle auxiliaire est validé et il intervient dans le calcul de la sortie du modèle *FFIR*. A l'instant $k = N + 1$, tous les modèles auxiliaires sont valides et à partir de cet instant jusqu'à l'instant $k = ph$, on aura :

$$\begin{aligned}
 [P.env]_{(k,HG)} &= [\theta_{HG}]^T \times [\varphi_{HG}]_k + [e_{HG}], \quad HG \in \{1, \dots, N\} \\
 [P.env]_k &= \bigcap_{HG=0}^N [P.env]_{(k,HG)}
 \end{aligned}$$

L'enveloppe englobante de la sortie du système sur l'horizon de prédiction ph sera :

$$\{[P.env]\} = \{[P.env]_1, \dots, [P.env]_{ph}\}$$

Dans un modèle *FFIR*, en considérant la méthode proposée pour prédire l'enveloppe englobante n'ayant pas accès aux sorties du système, on constate que :

$$\forall k, [env]_k = [P.env]_k \Rightarrow \{[env]\} = \{[P.env]\}$$

◇

L'erreur de prédiction dans la procédure de prédiction de l'enveloppe englobante de la sortie du système par un modèle intervalle est définie comme suit :

Définition 3.4.1. *L'erreur de prédiction est définie comme le pourcentage des instants échantillonnés où la sortie réelle du système se trouve en dehors de l'enveloppe englobante prédite sur l'horizon de prédiction.*

Exemple 3.4.1

Pour illustrer la méthode de prédiction sur un exemple, on choisit le modèle

FFIR qui a été identifié dans l'exemple 3.3.2. Une séquence des entrées $U = \{u_1 u_2 \dots u_{93}\}$ autre que celle utilisée en identification est appliquée au modèle *FFIR* et l'enveloppe englobante de la sortie du système est prédite. La prédiction se fait sur un horizon $ph = 93$. Pour valider la prédiction, la même séquence des entrées est appliquée au système 3.3.38 qui correspond au modèle ayant généré le jeu de données pour l'identification et représente le système physique. La sortie du système avec une incertitude de mesure telle que

$$[y]_k = y_k + \epsilon_k + [-0, 1 \ 0, 1]$$

ainsi que l'enveloppe de la sortie sont montrées sur la figure 3.20. ϵ_k est une variable aléatoire bornée et on sait a priori qu'elle prend des valeurs entre -0,1 et 0,1. Aux instants 3, 8, 20 et 27, la sortie du système se trouve en dehors de l'enveloppe englobante. Cela, par la définition 3.4.1, montre une erreur de prédiction égale à 4,3%. Les indicateurs de la qualité de l'enveloppe englobante -le rayon moyen et la précision de l'enveloppe- sont comme suit :

$$J = 0,1278$$

$$P = 0,8800$$

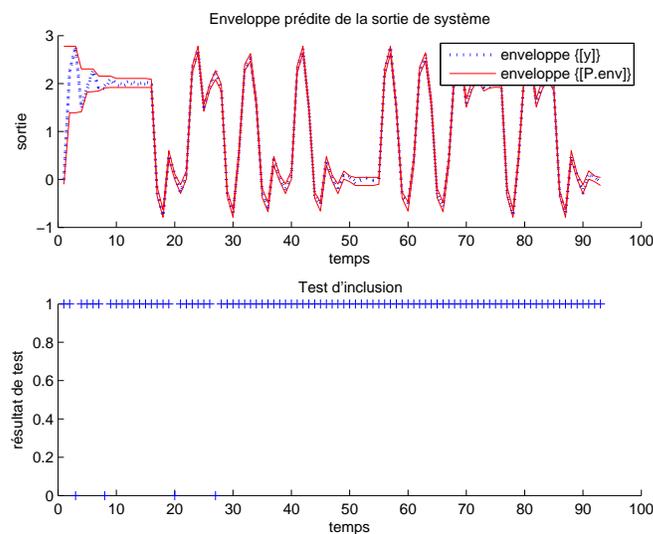


FIG. 3.20 – Prédiction de l'enveloppe englobante de la sortie du système par le modèle *FFIR*.

3.4.4 Prédiction de l'enveloppe englobante : Modèle *FIO*

Comme dans le cas précédent, on suppose que le modèle *FIO* est déjà identifié. La séquence des entrées dont on souhaite calculer l'enveloppe englobante est donnée par

Chapitre 3. Modèle Intervalle

$U = \{[u]_1 [u]_2 \dots [u]_{ph}\}$ et est différente de celle utilisée pour identifier les paramètres du modèle. La prédiction commence à l'instant $k = 1$. A cet instant, seul le modèle auxiliaire statique est valide. Donc, la prédiction se base sur la sortie du modèle statique :

$$\begin{aligned} [P.env]_{(1,0)} &= [\theta_0]^T \times [\varphi_0]_1 + [e_0] \\ [P.env]_1 &= [P.env]_{(1,0)} \end{aligned}$$

où

$$[\theta_0] = [b_{HG=0}], \quad [\varphi_0]_1 = [u]_1$$

A l'instant $k = 2$, pour construire le vecteur de régression $[\varphi_1]_2 = [[y]_1 [u]_1]^T$, l'entrée $[u]_1$ est disponible. Cependant, puisqu'on effectue une analyse de sûreté, on n'a pas accès à la sortie $[y]_1$. On substitue $[y]_1$ par sa prédiction $[P.env]_1$ qui a été calculée dans l'étape précédente et on obtient $[\varphi_1]_2 = [[P.env]_1 [u]_1]^T$. En considérant le théorème 3.4.1, on peut conclure que :

$$[y]_2 \subseteq [P.env]_{(2,1)}$$

où

$$[P.env]_{(2,1)} = [\theta_1]^T \times [\varphi_1]_2' + [e_1] \quad (3.4.8)$$

Par conséquent, l'enveloppe de la sortie du système $[P.env]_2$ à l'instant $k = 2$ se trouve aussi dans l'intersection de $[P.env]_{(2,0)}$ et $[P.env]_{(2,1)}$:

$$\begin{aligned} [P.env]_{(2,0)} &= [\theta_0]^T \times [\varphi_0]_2 + [e_0] \\ [P.env]_{(2,1)} &= [\theta_1]^T \times [\varphi_1]_2' + [e_1] \\ [P.env]_2 &= [P.env]_{(2,0)} \cap [P.env]_{(2,1)} \end{aligned}$$

où

$$\begin{aligned} [\theta_0] &= [b_{HG=0}] & [\varphi_0]_2 &= [u]_2 \\ [\theta_1] &= [[a_{HG=1} [b_{HG=1}]]^T & [\varphi_1]_2' &= [-[P.env]_1 [u]_1]^T \end{aligned}$$

et pour l'instant $k = 3$:

$$\begin{aligned} [P.env]_{(3,0)} &= [\theta_0]^T \times [\varphi_0]_3 + [e_0] \\ [P.env]_{(3,1)} &= [\theta_1]^T \times [\varphi_1]_3' + [e_1] \\ [P.env]_{(3,2)} &= [\theta_2]^T \times [\varphi_2]_3' + [e_2] \\ [P.env]_3 &= [P.env]_{(3,0)} \cap [P.env]_{(3,1)} \cap [P.env]_{(3,2)} \end{aligned}$$

où

$$\begin{aligned}
 [\theta_0] &= [b_{HG=0}] & [\varphi_0]_3 &= [u]_3 \\
 [\theta_1] &= [[a_{HG=1}] [b_{HG=1}]]^T & [\varphi_1]'_3 &= [-[P.env]_2 [u]_2]^T \\
 [\theta_2] &= [[a_{1,HG=2}] [a_{2,HG=2}] [b_{1,HG=2}] [b_{2,HG=2}]]^T & [\varphi_2]'_3 &= [-[P.env]_2 - [P.env]_1 [u]_2 [u]_1]^T
 \end{aligned}$$

A l'instant $k = N + 1$, tous les modèles auxiliaires sont valides et à partir de cet instant jusqu'à l'instant $k = ph$, on aura :

$$\begin{aligned}
 [P.env]_{(k,HG)} &= [\theta_{HG}]^T \times [\varphi_{HG}]'_k + [e_{HG}], \quad HG \in \{1, \dots, N\} \\
 [P.env]_k &= \bigcap_{HG=0}^N [P.env]_{(k,HG)}
 \end{aligned}$$

L'enveloppe englobante de la sortie du système sera :

$$\{[P.env]\} = \{[P.env]_1 \dots [P.env]_{ph}\}$$

Pour un modèle *FIO*, l'enveloppe $\{[P.env]\}$ qui est obtenue en utilisant les sorties estimées du système (on n'a pas accès aux sorties du système) englobe l'enveloppe $\{[env]\}$ qui serait obtenue en utilisant les sorties observées.

$$\forall k, [env]_k \subseteq [P.env]_k \Rightarrow \{[env]\} \subseteq \{[P.env]\}$$

Remarque. Remplacer les sorties observées $[y]_k$ par les domaine prédits $[P.env]_k$ dans le vecteur de régression est une source de surestimation. Dans cette méthode itérative proposée et à cause de l'effet d'enveloppement, cette substitution peut conduire à l'explosion du domaine de l'enveloppe englobante. \diamond

Exemple 3.4.2

Pour illustrer cette méthode sur un exemple, on identifie un modèle *FIO* d'ordre deux en utilisant le jeu de données des exemples 3.3.1 et 3.3.2. Après avoir identifié le modèle, la séquence des entrées de l'exemple 3.4.1 qui est différente de celle utilisée en identification, est appliquée au modèle *FIO* et l'enveloppe englobante est prédite sur l'horizon de prédiction $ph = 93$. La sortie du système qui a été calculée dans l'exemple 3.4.1 ainsi que l'enveloppe prédite de la sortie du système par le modèle *FIO* sont montrées sur la figure 3.21. A l'instant 3, la sortie du système se trouve en dehors de l'enveloppe englobante. Cela, par la définition 3.4.1, correspond à une erreur de prédiction égale à 1,07%. Dans cet exemple, le rayon moyen et la précision de l'enveloppe englobante sont égaux à :

$$\begin{aligned}
 J &= 0,5815 \\
 P &= 0,5590
 \end{aligned}$$

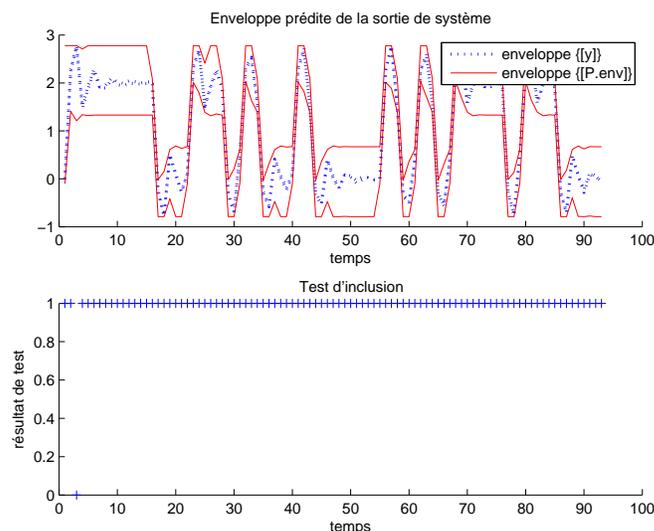


FIG. 3.21 – Prédiction de l’enveloppe englobante de la sortie de système par le modèle *FIO*.

Exemple 3.4.3

L’objectif de cet exemple est de montrer l’effet de la qualité du jeu de données DS sur la prédiction obtenue par le modèle. Pour cela, on identifie un modèle *FFIR* pour le système de l’exemple 3.3.1 avec un horizon $N = 10$, comme il a été expliqué dans l’exemple 3.3.2 sauf qu’ici, on utilise un jeu de données d’identification moins riche par rapport à l’exemple 3.3.2. L’enveloppe englobante est ensuite prédite pour la même séquence des entrées que l’exemple 3.4.1. Le résultat de la prédiction est montrée sur la figure 3.22.

On constate qu’à plusieurs instants, l’intersection des domaines $[P.env]_{(k,HG)}$ est vide ; c’est-à-dire $[P.env]_k = \phi$. Autrement dit, les modèles auxiliaires du modèle *FFIR*, à certains instants, ne prédisent pas correctement l’enveloppe englobante de la sortie du système et ils calculent des intervalles pour la sortie du système qui ne sont pas cohérents. Cela sont du fait que les paramètres des modèles auxiliaires ne sont pas correctement identifiés. Cela est aussi en accord avec la remarque de la section 3.3.6.

3.4.5 Résultats et Discussions

Dans les exemples 3.3.1, 3.3.2, 3.4.1 et 3.4.2, le système décrit par des équations 3.3.38 est modélisé par plusieurs modèles. Les résultats obtenus pour les indicateurs de la qualité de l’enveloppe englobante de ces exemples sont résumés dans le tableau 3.3. Considérons les résultats des exemples 3.3.1 et 3.3.2 :

- Pour une analyse pour laquelle les sorties du système sont disponibles, les modèles *FFIR* et *FIO* englobent la sortie du système par des enveloppes de qualité pratiquement identiques.

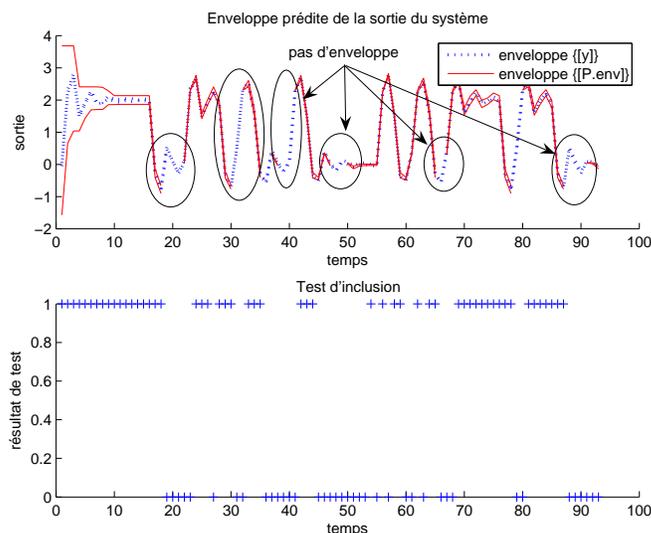


FIG. 3.22 – Prédiction de l’enveloppe englobante par le modèle $FFIR$.

- Le rayon de l’enveloppe englobante du modèle $FFIR$ diminue plus lentement par rapport à celui du modèle FIO . C’est dû à l’horizon glissant plus long du modèle $FFIR$ par rapport de celui du modèle FIO . On verra dans le chapitre suivant que pour un modèle multi-mode, la vitesse de la réduction du rayon de l’enveloppe englobante est importante puisqu’une commutation rapide ne laisse que peu de temps pour avoir une enveloppe précise et cela dégrade la précision de la prédiction.

Compte tenu des méthodes proposées dans les sections 3.4.3 et 3.4.4 et les résultats des exemples 3.4.1 et 3.4.2, on illustre que :

- Dans une analyse pour laquelle on n’a pas accès aux sorties du système, puisque dans un modèle FIO on remplace les sorties observées par celles estimées, l’enveloppe englobante prédite par ce modèle est moins précise que celle du modèle $FFIR$.
- Dans le modèle $FFIR$, puisque le vecteur de régression contient seulement les entrées, l’effet d’accumulation de l’incertitude n’apparaît pas. De ce fait, les enveloppes calculées par un modèle $FFIR$ dans le cas où on a accès aux sorties du système et dans le cas où elles ne sont pas disponibles sont identiques.

3.5 Conclusion

Dans ce chapitre, les procédures d’identification et de simulation des modèles intervalle ont été étudiées. Les modèles intervalle ont été utilisés dans ce travail pour gérer l’incertitude paramétrique du modèle. Après avoir défini l’objectif de l’identification et

	Sortie mesurée du système disponible	Sortie mesurée du système indisponible
Modèle <i>IO</i> première ordre	$J = 1, 2403$ $P = 0, 2893$	× ×
Modèle <i>IO</i> deuxième ordre	$J = 0, 2292$ $P = 0, 7952$	× ×
Modèle <i>FFIR</i> $HG = 10$	$J = 0, 3174$ $P = 0, 7280$	$J = 0, 1278$ $P = 0, 8800$
Modèle <i>FIO</i> $HG = 2$	$J = 0, 2886$ $P = 0, 7493$	$J = 0, 5815$ $P = 0, 5590$

TAB. 3.3 – La qualité de la prédiction de l’enveloppe englobante de la sortie du système pour une prédiction à un pas compte tenue que l’on ait accès ou non aux sorties du système.

la structure de modèle, une approche numérique a été proposée pour identifier les paramètres du modèle intervalle. Dans cette approche, un jeu de données expérimental est indispensable.

Une des propriétés des modèles dynamiques est l’effet de mémoire. Cet effet est le résultat direct de l’horizon glissant des modèles dynamiques choisis. Pour avoir un modèle qui soit toujours valide, le modèle *F* a été proposé. Il consiste en un modèle statique et *N* modèles dynamiques avec des horizons glissants qui varient entre $HG = 1$ à $HG = N$. Ces derniers s’appellent des modèles auxiliaires. La sortie du modèle *F* à chaque instant est l’intersection des sorties des modèles auxiliaires valides à cet instant.

Pour simuler un modèle intervalle, on a besoin de deux autres informations : la sémantique de simulation qui est définie en fonction de l’objectif de simulation et la méthode de calcul de l’enveloppe englobante déterminée par la sémantique de simulation. Après avoir déterminé la sémantique dans le contexte d’une analyse de sûreté, deux méthodes ont été proposées pour calculer cette enveloppe -une méthode pour le modèle *FFIR* et une méthode pour le modèle *FIO*- sur un horizon de prédiction donnée. Dans le cas où les observations de la sortie du système sont disponibles pendant la simulation, la qualité de l’enveloppe englobante calculée par un modèle *FFIR* ou un modèle *FIO* est presque identique. Par contre, dans le cas où les observations de la sortie du système ne sont pas disponibles, l’enveloppe englobante calculée par un modèle *FIO* n’est pas autant précise que celle d’un modèle *FFIR*. C’est à cause de l’accumulation de l’imprécision qui se produit en cours de calcul et en remplaçant les sorties observées par celles estimées dans le vecteur de régression du modèle *FIO*. Dans un modèle *FFIR*, puisque les éléments

de son vecteur de régression sont des entrées, cette accumulation de l'imprécision ne se produit jamais. Les procédures d'identification et de simulation ont été ensuite illustrées sur des exemples et les résultats obtenus ont été analysés.

Sur le tableau 3.1, on constate qu'en fonction de la structure de modèle et dans certains cas, le critère d'optimisation est non-linéaire par rapport aux paramètres du modèle. La méthode d'optimisation utilisée dans ce travail garantit seulement une solution optimale pour un critère d'optimisation linéaire. Pour résoudre un problème d'optimisation non-linéaire, les méthodes d'optimisation globale doivent être envisagées. Ces méthodes sont souvent très lentes par rapport à la méthode utilisée. Pour un modèle mono-mode, cette lenteur peut être tolérée. Néanmoins, pour la modélisation multi-mode et pendant le partitionnement du régime opératoire du système, le temps de calcul augmente avec le nombre des points du jeu de données (voir section 4.5.2). Pour éviter un temps de calcul considérable, pendant ce travail, les méthodes d'optimisation globales ne sont pas appliquées au problème d'optimisation des paramètres de modèle.

Dans le chapitre suivant, les problèmes de l'identification et de la simulation multi-mode seront étudiés.

Chapitre 4

Modèle Multi-Mode

Je sais pourquoi tant de gens aiment couper du bois. C'est une activité où l'on voit tout de suite le résultat.

Albert Einstein

4.1 Introduction

Dans ce chapitre, qui consiste en sept sections, le problème d'identification et de simulation d'un modèle multi-mode intervalle sera traité. Après avoir expliqué les objectifs de l'identification d'un modèle multi-mode et les raisons pour lesquelles cette approche est choisie, dans la section trois, l'identification multi-mode sera étudiée dans le cas où le système est un système multi-mode dans lequel le mode opératoire est déterminé (commutation déterminée). Le modèle sera ensuite utilisé, dans la section quatre, pour prédire l'enveloppe englobante de la sortie du système.

L'identification d'un modèle multi-mode est plus difficile à réaliser quand le mode opératoire est indéterminé (commutation indéterminée). Dans ce cas, trois problèmes supplémentaires doivent être résolus : partitionner le régime opératoire du système en plusieurs zones locales et représenter le comportement du système dans ces zones par des modèles locaux, déterminer les domaines de validité des modèles locaux et étudier la possibilité d'agrèger des modèles de comportement similaires. Ces problèmes seront étudiés dans la section cinq. Dans la section six, on expliquera comment le modèle identifié dans la section précédente peut être utilisé pour prédire l'enveloppe englobante de la sortie du système. On illustrera les méthodes d'identification et de simulation par un exemple numérique dans la section sept. Ce chapitre se terminera par une conclusion.

4.2 Objectif

En ingénierie, une approche standard pour résoudre les problèmes complexes est la stratégie de diviser-pour-régner¹. Un problème complexe d'une manière ou d'une autre est alors divisé en un nombre fini de sous-problèmes plus simples qui peuvent être résolus individuellement tels que leurs solutions individuelles conduisent à la solution du problème original. La clé du succès de cette approche est de trouver les bons axes pour faire le découpage (partitionnement dans notre cas). Considérons le problème consistant à développer un modèle pour représenter le comportement d'un système complexe. Au lieu de chercher à modéliser en un coup l'ensemble du système, celui-ci peut être décomposé de différentes manières [Murray-Smith & Johansen, 1997b] :

- Décomposition en composants physiques
- Décomposition basée sur les phénomènes
- Décomposition du régime opératoire

Dans ce document, la complexité du système est au niveau de son comportement et on souhaite traiter le problème de l'identification et de la simulation des systèmes multi-mode ou non-linéaires dans le contexte de l'analyse de sûreté. Ce problème a été traité pour un système linéaire mono-mode (un système simple) dans le chapitre précédent. Dans le contexte de la sûreté, le fait d'englober le comportement d'un système multi-mode ou non-linéaire par un modèle linéaire intervalle peut introduire beaucoup de pessimisme sur les supports des paramètres du modèle et par conséquent, l'enveloppe englobante prédite pour la sortie du système ne sera pas précise. Dans ce chapitre, on proposera des méthodes pour diviser le régime opératoire du système en plusieurs zones locales et déterminer les modèles individuels pour chaque zone.

Très souvent, un modèle linéaire mono-mode intervalle n'est pas capable de prédire l'enveloppe englobante de la sortie d'un système non-linéaire ou multi-mode avec une précision demandée. Un système statique et non-linéaire par rapport à l'entrée qui a été représentée par un modèle linéaire mono-mode intervalle est montrée sur la figure 4.1.

Pour obtenir un modèle plus précis et atteindre la précision désirée, le régime opératoire du système sera divisé en plusieurs zones telles que les zones peuvent être représentées par des modèles linéaires intervalle avec la précision désirée. Cette solution est très proche de celle consistant à linéariser le système autour de plusieurs points de fonctionnement. La même courbe que la figure 4.1 est représentée par un modèle linéaire multi-mode intervalle sur la figure 4.2.

L'application des modèles multi-mode n'est pas limitée aux systèmes non-linéaires. Cette approche peut aussi être utilisée pour représenter des systèmes commutés. Un exemple de tels systèmes est une voiture dans laquelle la position de l'accélérateur est l'entrée et la vitesse de la voiture est la sortie du système. Chaque mode opératoire est déterminé par la position du levier de vitesse et la commutation aura lieu quand celle-ci

¹Divide-and-Conquer strategy

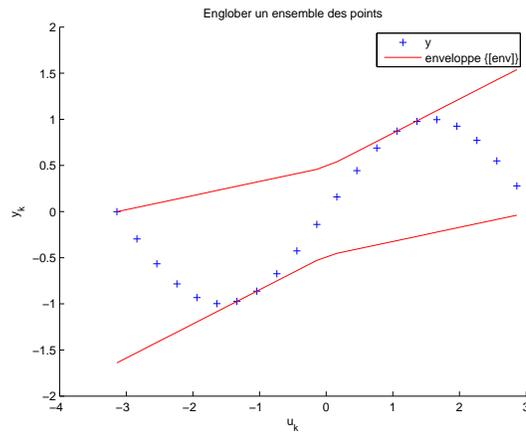


FIG. 4.1 – Modèle linéaire mono-mode intervalle

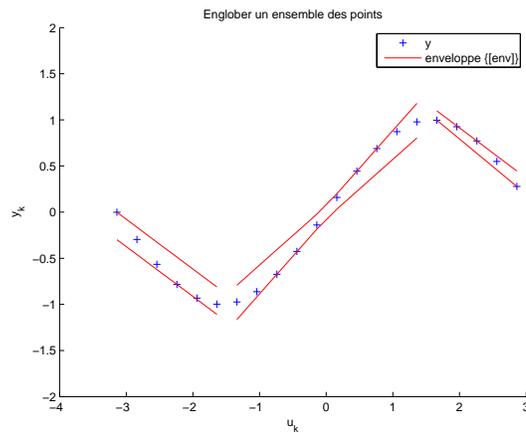


FIG. 4.2 – Modèle linéaire multi-mode intervalle

est changée. Dans un système commuté, la commutation est déterminée soit par un signal externe soit par un certain nombre de conditions sur l'état du système. Dans l'exemple de la boîte de vitesse, si elle est manuelle, c'est le conducteur qui détermine la commutation en changeant la position du levier -le signal externe- et si elle est automatique, c'est un certain nombre de conditions qui la déterminera. Pendant la procédure d'identification, dans le premier cas, la commutation se produit quand le signal externe change d'état. Cependant dans le deuxième cas, l'instant et les conditions de la commutation doivent être détectés et déterminés.

4.3 Identification Multi-Modes : commutation déterminée

Dans cette section, on étudiera le problème d'identification d'un système multi-mode dans lequel la commutation est déterminée par un signal externe. Le système consiste en plusieurs *modes opératoires* et il se comporte différemment dans chaque mode. Le mode actif est déterminé par un signal qui s'appelle le *signal de commutation*. La sortie du système à chaque instant est déterminée par le mode opératoire actif. Le schéma de ce système est donné sur la figure 4.3.

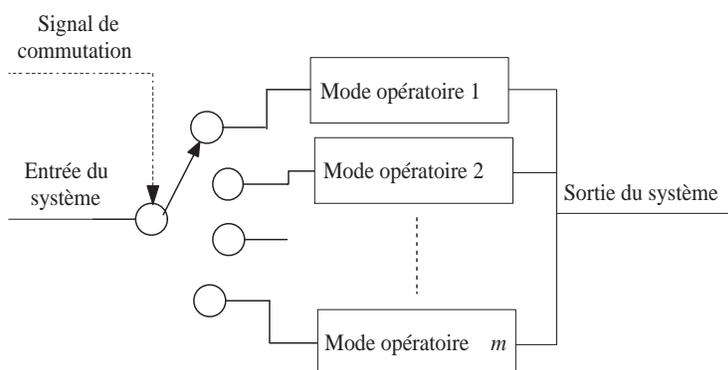


FIG. 4.3 – Structure d'un système multi-mode en présence du signal de commutation

En présence du signal de commutation, le problème d'identification d'un modèle multi-mode se traduit par plusieurs problèmes d'identification de modèles mono-mode; le comportement pour chaque mode opératoire est représenté individuellement. Le modèle qui représente le comportement d'un mode opératoire s'appelle le *modèle local* et l'ensemble des modèles locaux constitue le *modèle multi-mode*.

4.3.1 Jeu de données

Pour commencer la procédure d'identification, on a besoin de données expérimentales pour chacun des modes opératoires. On suppose que des observations sur le système $DS = \{([u]_k, [y]_k)\}$ sont disponibles.

Pour identifier les paramètres du modèle local F_i^2 , on doit extraire les données qui correspondent à ce mode opératoire du jeu de données DS . En prenant en compte le signal de commutation, le jeu de données DS peut être divisé en sous-ensembles DS_i . Le sous-ensemble DS_i contient des observations sur le système qui correspondent au mode opératoire i . Il doit être suffisamment riche pour représenter le comportement de ce mode

²Comme on l'a déjà vu dans la section 3.3.6, un modèle F est un banc de modèles auxiliaires constitué d'un modèle statique et d'un certain nombre de modèles dynamiques dont les horizons glissants varient de $HG = 1$ à $HG = N$.

opérateur. Ceci est une condition nécessaire pour garantir la qualité de l'identification des paramètres du modèle local F_i .

En pratique, un système multi-mode change alternativement de modes opératoires et peut se retrouver plusieurs fois dans le même mode. Dans ce cas, les échantillons qui correspondent au mode i et qui sont dispersés sur l'axe du temps seront rassemblés dans le sous-ensemble DS_i qui sera ensuite utilisé pour l'identification des paramètres.

Pour le modèle local F_i , l'horizon glissant du modèle statique est nul. De ce fait, le modèle statique i est toujours valide pendant que le système se trouve dans ce mode. Soit $IW_{i,j,s}$ une fenêtre temporelle sur laquelle les paramètres du modèle statique du mode i peuvent être identifiés, on a :

$$\begin{aligned} IW_{i,j,s} &= \{k \mid k_j^{debut,i} \leq k \leq k_j^{fin,i}\} \\ IW_{i,s} &= \bigcup_j IW_{i,j,s} \end{aligned} \quad (4.3.1)$$

où $k_j^{debut,i}$ représente l'instant où le système entre d'un mode quelconque dans le mode i et $k_j^{fin,i}$ représente le dernier instant où le système se trouve dans le mode i avant la prochaine commutation. j est un indice pour énumérer les fenêtres temporelles $IW_{i,j,s}$ et $IW_{i,s}$ représente l'ensemble de celles-ci. Par définition de $IW_{i,s}$ dans 4.3.1, on a :

$$DS_i = \{(u_k, y_k) \mid k \in IW_{i,s}\}$$

Cependant, les modèles auxiliaires dynamiques ne peuvent pas être identifiés pour tous les échantillons de DS_i . L'identification des paramètres du modèle auxiliaire avec un horizon glissant HG ne commence qu'à partir de HG échantillons après chaque commutation. Autrement dit, après chaque commutation, il faut attendre d'avoir plus de HG échantillons pour pouvoir identifier les paramètres du modèle auxiliaire HG du mode actif i . Soit $IW_{i,j,HG}$, une fenêtre sur laquelle les paramètres du modèle auxiliaire HG de mode i peuvent être identifiés ; on a :

$$\begin{aligned} IW_{i,j,HG} &= \{(u_k, y_k) \mid k_j^{debut,i} + HG \leq k < k_j^{fin,i}\} \\ IW_{i,HG} &= \bigcup_j IW_{i,j,HG} \end{aligned}$$

Ces fenêtres sont montrées sur la figure 4.4.

Par exemple, si le système passe m fois dans le mode opératoire i , les paramètres du modèle statique peuvent être identifiés sur $IW_{i,s} = \{IW_{i,1,s}, \dots, IW_{i,m,s}\}$, les paramètres du modèle dynamique avec $HG = 1$ sur $IW_{i,HG=1} = \{IW_{i,1,1}, \dots, IW_{i,m,1}\}$, les paramètres du modèle dynamique avec $HG = 2$ sur $IW_{i,HG=2} = \{IW_{i,1,2}, \dots, IW_{i,m,2}\}$ etc.

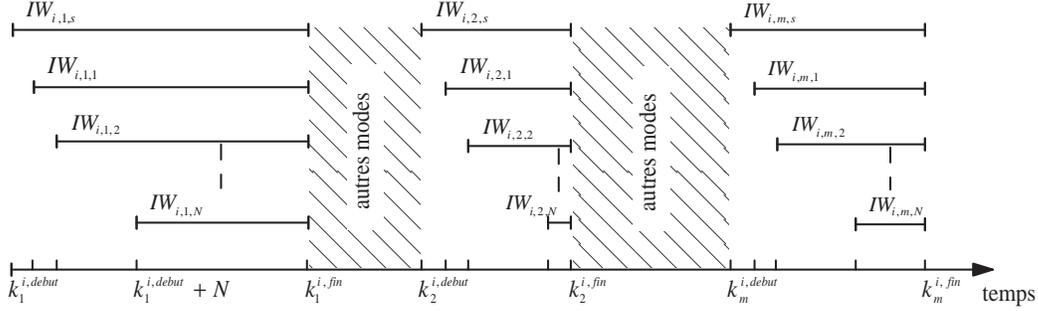


FIG. 4.4 – Fenêtres temporelles pour l’identification des modèles auxiliaires.

4.3.2 Structure de modèle

La structure de chaque modèle local est choisie comme le préconise la section 3.3.2. Pour surmonter l’effet de mémoire et sur le même principe qui a été expliqué dans la section 3.3.6, le comportement de chaque mode opératoire sera représenté par un modèle F (FIO ou $FFIR$) et l’horizon glissant du modèle F_i est choisi en fonction de sa connaissance a priori sur le comportement du système dans ce mode. Les horizons glissants des modèles locaux F_i peuvent varier d’un mode à un autre. Cependant, pour simplifier la présentation, ils seront supposés identiques par la suite. Pour prendre en compte les incertitudes paramétriques, les modèles auxiliaires sont des modèles intervalle.

4.3.3 Sémantique d’identification

Pour tous les i , la sémantique d’identification définie par les équations 3.3.21 et 3.3.22 sera utilisée pour identifier le modèle F_i qui représentera le comportement du mode opératoire i .

4.3.4 Critère d’optimisation

A partir de l’équation 3.3.24 et pour tous les i , le modèle local F_i est identifié indépendamment des autres modèles locaux tel que l’enveloppe de ce modèle soit l’enveloppe la plus précise permettant d’expliquer les données DS_i .

Définition 4.3.1. *La précision de l’enveloppe englobante d’un modèle multi-mode est définie comme la précision la plus mauvaise des enveloppes de modèles locaux; c’est-à-dire :*

$$P = \min_i(P_i)$$

où

$$P_i = \exp(-J_i)$$

4.3.5 Calcul des paramètres du modèle

La méthode d'identification d'un modèle F a été expliquée dans les sections 3.3.5 et 3.3.6 du chapitre précédent. Dans un modèle multi-mode avec une commutation déterminée, la procédure d'identification doit être répétée pour tous les modes opératoires i . A la fin de la phase de l'identification, pour tous les modes opératoires i , on aura un modèle F_i qui représente le comportement du système dans ce mode. Dans la prochaine section, on expliquera comment ce modèle multi-mode peut être utilisé pour prédire l'enveloppe englobante de la sortie du système.

4.4 Simulation Multi-Modes : commutation déterminée

Après avoir identifié le modèle multi-mode, ce modèle peut être utilisé dans diverses applications pour analyser le système quand la sortie du système est disponible ou non-disponible. Un exemple pour le premier cas est la détection de défauts où on mesure par exemple la sortie du système et où on l'utilise pour la comparer avec la sortie prédite par le modèle. Un exemple pour le deuxième cas est l'analyse de sûreté où on veut prédire la sortie du système sur un horizon de prédiction pour vérifier que la trajectoire (ou l'ensemble des trajectoires dans le cas ensembliste) satisfait des propriétés données. Le schéma du modèle multi-mode est montré sur la figure 4.5.

Dans cette section, comme pour la section 3.4, l'objectif est de prédire l'enveloppe englobante de la sortie de système pour une séquence des entrées qui est donnée par $U = \{[u]_1 [u]_2 [u]_3 \dots [u]_{ph}\}$. ph s'appelle, comme précédemment, l'horizon de prédiction. Avec la séquence des entrées U , la séquence

$$AM = \{am_1, am_2, am_3, \dots, am_{ph}\}, \quad am_i = 1, 2, \dots, m$$

détermine le *mode actif* à chaque instant.

Ensuite, on rappellera brièvement, la sémantique de simulation et puis, on expliquera comment l'enveloppe englobante de la sortie du système peut être obtenue par ce modèle multi-mode.

4.4.1 Sémantique de simulation

Comme on l'a déjà expliqué, pour l'analyse de sûreté, on souhaite englober toutes les valeurs possibles de la sortie du système par une enveloppe. Pour un modèle mono-mode, la sémantique de simulation est donnée par les équations 3.4.1 et 3.4.2. Pour le mode opératoire actif i , la même sémantique de simulation sera utilisée pour calculer l'enveloppe englobante de la sortie du système.

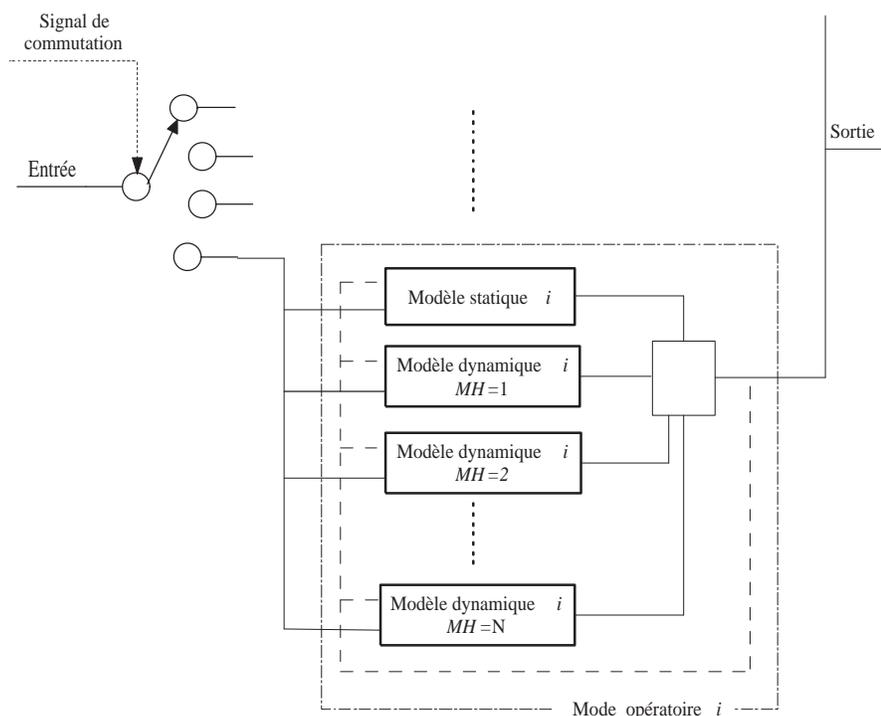


FIG. 4.5 – Schéma du modèle multi-mode quand la commutation est déterminée.

4.4.2 Méthode numérique pour calculer l'enveloppe

Dans le cas où la sortie du système est également disponible pour la séquence des entrées U et pour le mode opératoire actif i , on utilise le prédicteur à un pas de l'équation 3.3.20 et l'enveloppe englobante de la sortie à l'instant k est déterminée par les équations 3.3.36 et 3.3.37.

Le calcul de l'enveloppe englobante de la sortie a été expliqué pour un modèle mono-mode $FFIR$ dans la section 3.4.3 et un modèle mono-mode FIO dans la section 3.4.4 dans le cas où la sortie du système n'est pas disponible pour la séquence des entrées U . **La différence entre la prédiction de l'enveloppe englobante par un modèle mono-mode et par un modèle multi-mode quand la commutation est déterminée** est que pour le deuxième, en fonction du signal de commutation, le modèle commute à des instants précis d'un mode à un autre donné. Après chaque commutation, le modèle F_i actif est choisi.

A cause du manque d'information sur les entrées et les sorties précédentes du système dans ce nouveau mode et juste après la commutation, les modèles auxiliaires du modèle F_i du mode récemment actif avec les horizons glissants élevés ne sont pas valides. Par conséquent, on commence la prédiction par le modèle statique et puis, les modèles dynamiques interviennent au fur et à mesure aux instants suivants. Grâce au modèle F_i , on est toujours capable de calculer l'enveloppe englobante même si cette enveloppe n'est pas précise pendant les premiers instants après les commutations.

Dans le cas où la mesure de la sortie du système n'est pas disponible, il est suggéré d'utiliser la structure *FFIR* pour les modèles locaux. Comme expliqué dans la section 3.4.4, pour calculer l'enveloppe englobante par un modèle *FIO*, les sorties observées sont remplacées par des sorties estimées dans les vecteurs de régression des modèles auxiliaires. Surestimer la sortie observée par le domaine estimé sera une source d'accumulation de l'imprécision des modèles après chaque itération. On obtient donc une enveloppe moins précise. Dans certains cas, l'accumulation de l'imprécision peut aussi faire exploser le domaine de calcul. Contrairement à un modèle *FIO*, l'enveloppe calculée par une structure *FFIR* dépend seulement des entrées. De ce fait, aucune accumulation d'imprécision ne se produira.

Dans le travail de [Shahriari et al., 2006], l'identification et la simulation d'un système multi-mode en présence du signal de commutation a été étudié.

Exemple 4.4.1

L'objectif de cet exemple est de prédire l'enveloppe englobante de la sortie d'un système multi-mode dont la commutation est déterminée par un signal de commutation. Le schéma du système est donné sur la figure 4.6.

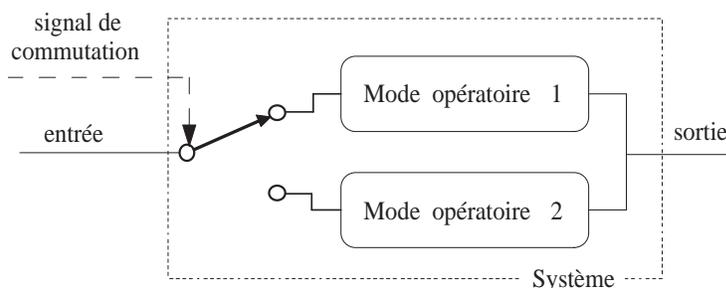


FIG. 4.6 – Système

Système. Le système consiste en deux modes opératoires. Le comportement de chaque mode est décrit par les équations suivantes :

$$\begin{aligned}
 x_{k+1} &= A_i \times x_k + B_i \times u_k, \quad y_k = C_i \times x_k, \quad i \in \{1, 2\} \\
 A_1 &= \begin{bmatrix} 0.4315 & 0.3158 \\ -0.7105 & -0.1369 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0.5685 \\ 0.7105 \end{bmatrix}, \quad C_1 = [1 \ 0] \\
 A_2 &= \begin{bmatrix} -0.1275 & 0.3165 \\ -1.2662 & -0.3807 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1.1275 \\ 1.2662 \end{bmatrix}, \quad C_2 = [1 \ 0]
 \end{aligned}
 \tag{4.4.1}$$

Les pôles du premier système sont

$$p_1 = -0.9000 + 1.2000i, \quad p_2 = -0.9000 - 1.2000i$$

et les pôles du deuxième système sont

$$p_1 = -0.4000 + 1.9596i, \quad p_2 = -0.4000 - 1.9596i$$

Jeu de données. On conçoit une expérience sur le système et on obtient le jeu de données DS . En pratique, la valeur de y_k est connue avec imprécision et la sortie du système est mesurée avec une erreur ϵ_k où ϵ_k est une variable aléatoire et on sait a priori qu'elle varie entre $-0, 1$ et $0, 1$. Pour qu'on soit sûr que la sortie y_k est prise en compte dans les calculs, la sortie mesurée $y_k + \epsilon_k$ est encadrée par l'intervalle $[-0, 1 \ 0, 1]$ qui sera ensuite utilisé dans le jeu de données DS :

$$[y]_k \leftarrow \epsilon_k + [-0, 1 \ 0, 1]$$

L'entrée reste certaine. Donc, le jeu de données d'identification est de la forme $DS = [(u_k, [y]_k)]$.

Identification des modèles locaux. La structure $FFIR$ est choisie pour les modèles locaux. Considérant l'information a priori sur le système, l'horizon glissant des modèles F_1 et F_2 est fixé à 10. Ensuite, les paramètres des modèles locaux sont identifiés en appliquant la méthode proposée dans la section 4.3.

Prédiction de l'enveloppe englobante de la sortie. L'objectif est de prédire l'enveloppe englobante de la sortie $\{[P.env]\}$ sur un horizon de prédiction ph pour une séquence des entrées données. Les entrées sont certaines. Le système reste dans le premier mode de $k = 1$ à $k = 93$ et il commute ensuite dans le deuxième mode dans lequel il reste jusqu'à la fin de l'horizon de prédiction. L'enveloppe $\{[P.env]\}$ est calculée, en considérant le mode actif, en appliquant la méthode proposée dans la section 3.4.3. La même séquence des entrées est appliquée sur le système 4.4.1 et la sortie observée est construite. L'enveloppe englobante prédite et la sortie observée du système sont montrées sur la figure 4.7.

Aux instants 65, 102, 112 et 115, la sortie observée en dehors de l'enveloppe prédite, ce qui correspond à une erreur de prédiction égale à 2,1% d'après la définition 3.4.1. On voit aussi l'effet du manque d'information qui conduit à une enveloppe moins précise au début de la prédiction ainsi qu'après la commutation. Puisque la structure choisie est le $FFIR$, l'effet d'enveloppement n'existe pas et la largeur de l'enveloppe prédite se réduit pas à pas.

4.5 Identification Multi-Modes : commutation indéterminée

Dans la section précédente, on a traité le problème de l'identification et de la simulation d'un système multi-mode dans lequel le mode actif est déterminé par un signal

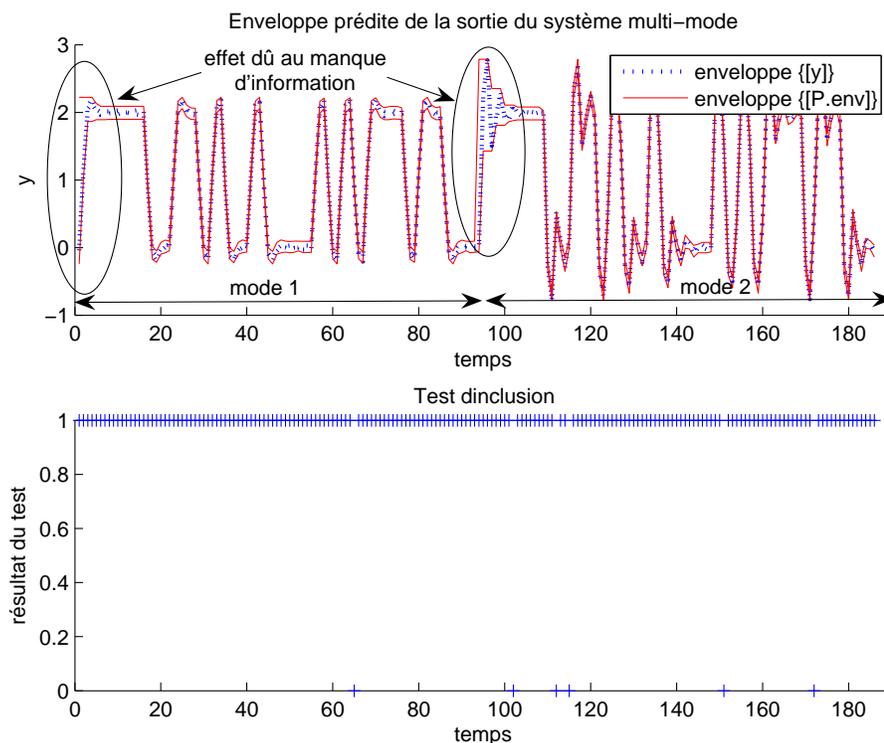


FIG. 4.7 – Prédiction de l’enveloppe englobante de la sortie ; commutation déterminée.

de commutation. Dans cette section, on étudiera ce problème en l’absence de ce signal. Pour cela, 5 problèmes supplémentaires doivent être résolus :

Déterminer les variables caractéristiques. Les variables caractéristiques sont des variables qui représentent le comportement du système et qui sont un sous-ensemble des variables du modèle.

Partitionner le régime opératoire du système en zones locales. Le régime opératoire du système (un système commuté ou un système non-linéaire) doit être partitionné en zones locales telles que le comportement du système dans chaque zone puisse être représenté par un modèle linéaire local F_i . Les modèles locaux sont identifiés individuellement en utilisant les données des zones locales.

Déterminer les domaines de validité des modèles locaux. Le domaine de validité d’un modèle local est le domaine dans lequel le modèle explique ce comportement du système. Ce domaine sera déterminé dans *l’espace du domaine de validité*.

Agréger les modèles correspondant à des comportements similaires .

Développer un simulateur pour ce modèle ayant pour but de prédire l'enveloppe englobante de la sortie du système dans le contexte d'une analyse de sûreté.

Dans les prochaines sections, on expliquera en détail les 5 démarches ci-dessus.

4.5.1 Variables caractéristiques

Dans ce document, puisqu'on traite le problème de l'identification par l'approche boîte-noire et que les variables qui représentent le comportement du système sont inconnues a priori, on suppose que toutes les variables du modèle (les éléments du vecteur de régression) sont les variables caractéristiques. Selon cette définition, partitionner le régime opératoire du système peut aussi être interprété comme partitionner l'espace du vecteur de régression du modèle en plusieurs zones locales telle que chaque zone puisse être représentée par un modèle linéaire avec une précision désirée.

4.5.2 Partitionnement du régime opératoire

Dans cette section, deux méthodes seront proposées pour partitionner le régime opératoire du système. La première méthode est basée sur la *précision* de l'enveloppe englobante de la sortie et la deuxième sur la *cohérence* entre la prédiction par le modèle et la sortie observée du système.

Dans les deux méthodes, le découpage des données et l'identification des paramètres des modèles locaux se font en même temps. Au préalable, la structure des modèles locaux doit être déterminée. Le modèle F à intervalle est notre choix dans ce document pour représenter le comportement du système dans les zones locales de fonctionnement. Les horizons glissants des modèles locaux sont choisis en fonction de la connaissance a priori sur le comportement du système dans chaque zone. Pour simplifier l'étude, les horizons glissants des modèles locaux seront pris identiques.

Partitionner en utilisant la précision de l'enveloppe englobante

Dans le cas des systèmes non-linéaires, l'objectif est de partitionner le régime opératoire du système en plusieurs zones et de représenter le système dans chaque zone par un modèle linéaire tel que chaque modèle local produit une enveloppe englobante pour la sortie du système avec une précision désirée. De ce fait, la précision de l'enveloppe englobante est le critère permettant de diviser le jeu de donnée DS en sous-ensembles DS_i s.

Remarque. Après chaque commutation, durant le passage de l'horizon glissant du modèle F , l'enveloppe englobante de la sortie a une largeur importante. On ne considère pas cette partie de l'enveloppe dans le calcul du rayon moyen de l'enveloppe. Le calcul de la précision du modèle local par le partitionnement ne prend donc en compte que la partie de l'enveloppe située après un nombre de points égal à l'horizon glissant du modèle local. Donc, aucune commutation ne sera détectée pendant les N instants après

chaque commutation. Dans le cas d'un système commuté, le système doit rester au moins N instants dans un mode opératoire avant de le quitter pour que les modes opératoires puissent être détectés correctement. \diamond

On suppose que la précision désirée est une valeur pré-définie égale à pre . On suppose aussi que la dernière commutation a eu lieu à l'instant noté $début$ et qu'on reprend la procédure de découpage à partir de cet instant. Avant l'instant $début$, $i - 1$ modes opératoires ont été trouvés et donc, à l'instant d'échantillonnage $début$, on entre dans le mode opératoire numéro i .

Dans un premier temps, les paramètres du modèle F_i sont identifiés en utilisant les données

$$DS^{debut,debut+N} = \{([u]_k, [y]_k) \mid debut \leq k \leq debut + N\}$$

A cet instant, l'enveloppe de la sortie du système a une précision $P = 1$. Ensuite, le point suivant est ajouté au jeu de données et les paramètres du modèle F_i sont re-identifiés en utilisant les données $DS^{debut,debut+N+1}$. Si la précision de l'enveloppe de ce nouveau modèle satisfait la précision désirée, le point $([u]_{debut+N+1}, [y]_{debut+N+1})$ peut être intégré dans le mode courant. On continue cette procédure jusqu'à l'instant $debut + q$ où $q > N$ pour lequel si on ajoute le point $([u]_{debut+q}, [y]_{debut+q})$ dans le jeu de données du mode courant, la précision de l'enveloppe ne satisfait plus la précision désirée. Cela signifie que ce point ne peut pas être intégré dans le mode i . Donc,

$$DS_i = DS^{debut,debut+q-1}$$

détermine le jeu de données du mode opératoire i . Les paramètres du modèle F_i sont les paramètres identifiés en utilisant l'ensemble $DS^{debut,debut+q-1}$ et un nouveau mode commence à l'instant $k = debut + q$. Cette procédure doit être répétée jusqu'à ce qu'on arrive à la fin du jeu de données initial DS . L'algorithme 1 résume cette méthode en quelques lignes. Cette méthode est aussi illustrée sur la figure 4.8.

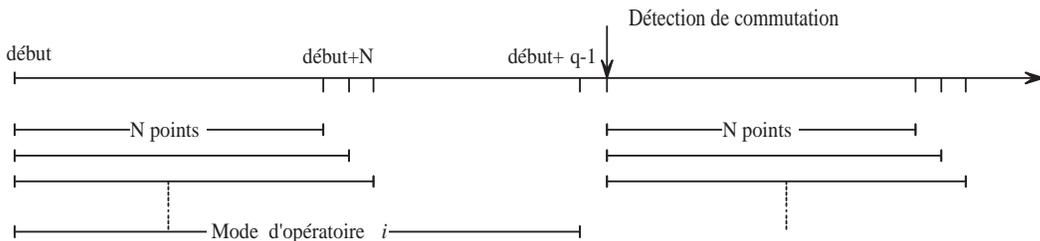


FIG. 4.8 – Partitionner le régime opératoire ; méthode précision de l'enveloppe englobante.

Algorithme 1

Pre : la précision désirée pour l'enveloppe de la sortie

DS : le jeu de données expérimental

N : l'horizon glissant du modèle F_i

1. **Commencer**

2. $debut \leftarrow 1, fin \leftarrow N, i \leftarrow 1$

3. **Tant que** $fin <$ (le nombre d'éléments de DS)

(a) **Identifier** les paramètres du modèle F_i en utilisant $DS^{debut,fin}$

(b) **Si** (la précision de l'enveloppe du modèle F_i) $< pre$

i. **Prendre** $DS^{debut,fin-1}$ les observations du i ème mode opératoire. **Utiliser** les paramètres identifiés en utilisant $DS^{debut,fin-1}$ pour le modèle F_i .

ii. $i \leftarrow i + 1$

iii. $debut \leftarrow fin$

iv. $fin \leftarrow debut + N$

v. **Aller** à la ligne 3

(c) **Sinon**

i. $fin \leftarrow fin + 1$

ii. **Aller** à la ligne 3

4. **Fin**

Partitionner en utilisant la cohérence de la sortie observée avec la sortie prédite

Une autre méthode pour partitionner le régime opératoire du système est d'utiliser le test de cohérence entre l'enveloppe prédite par le modèle et la sortie du système. Avant d'expliquer cette méthode, on a besoin de donner la définition de ce qu'on appellera par la suite la *cohérence de la sortie observée avec la sortie prédite* qui sont dans notre cas à un instant donné toutes les deux des intervalles.

Définition 4.5.1. *Un intervalle $[a]$ est cohérent avec l'intervalle $[b]$ si et seulement si pour un $\delta \geq 0$ donné, on a :*

$$\delta \geq 0, [a] \approx [b] \Leftrightarrow [a] \subseteq [\underline{b} - \delta \bar{b} + \delta]$$

δ s'appelle la *tolérance* sur la cohérence de l'intervalle $[a]$ avec l'intervalle $[b]$. Dans le cas général, si $[a]$ est cohérent avec $[b]$, il n'en est pas de même de $[b]$ avec $[a]$. La cohérence d'un point avec un intervalle est un cas particulier pour lequel le rayon de l'intervalle $[a]$ est nul ; autrement dit $[a] = a$ et on aura :

$$\delta \geq 0, a \approx [b] \Leftrightarrow a \in [\underline{b} - \delta \bar{b} + \delta]$$

Définition 4.5.2. La tolérance sur la cohérence de l'intervalle $[a]$ avec l'intervalle $[b]$ est définie comme la valeur minimale de δ telle que :

$$[a] \subseteq [\underline{b} - \delta \bar{b} + \delta]$$

Le point commun entre la première méthode et cette méthode est que les points sont ajoutés un par un au jeu de données du mode actif. Pourtant, un autre test est utilisé pour diviser DS ; ce test est le test de cohérence de la sortie du système avec la sortie prédite.

On suppose, comme avant, que la dernière commutation a eu lieu à l'instant d'échantillonnage *début* et on commence la procédure à partir de cet instant. On suppose aussi qu'avant l'instant *début*, $i - 1$ modes opératoire ont été trouvés et donc, à cet instant, on rentre dans le mode opératoire i .

Pour commencer la procédure, les paramètres du nouveau modèle F_i sont identifiés sur un horizon d'identification initial IIH en utilisant le jeu de données $DS_i = DS^{debut, debut+IIH}$. Cet horizon doit être assez long pour permettre au modèle F_i d'être représentatif du comportement du système. La longueur de cet horizon est **pré-déterminée** par l'utilisateur en fonction de sa connaissance sur le jeu de données ainsi que sur le comportement du système et la structure de modèle choisie.

Après avoir identifié les paramètres du modèle F_i , on fait une prédiction à un pas et on calcule $[env]_{debut+IIH+1}$. La cohérence entre la sortie observée $[y]_{debut+IIH+1}$ et la prédiction $[env]_{debut+IIH+1}$ montre si le modèle est capable de bien prédire la sortie du système ou non. Si la sortie observée $[y]_{debut+IIH+1}$ est cohérente avec la sortie estimée $[env]_{debut+IIH+1}$, le point $([u]_{debut+IIH+1}, [y]_{debut+IIH+1})$ est ajouté dans le jeu de données du mode i et les paramètres du modèle F_i sont re-identifiés en utilisant $DS^{debut+IIH+1}$. Cette procédure est poursuivie jusqu'à l'instant $debut + q$, $q > IIH$ pour lequel la sortie observée n'est plus cohérente avec l'enveloppe prédite. Cette incohérence montre que le modèle n'est plus capable de prédire le comportement du système. Dans ces conditions, on définit le jeu de données du mode i par $DS_i = DS^{debut+q-1}$ et un nouveau mode commence à l'instant $k = debut + q$. Cette procédure est poursuivie jusqu'à ce qu'on arrive à la fin du jeu de données DS . La méthode proposée est illustrée sur la figure 4.9 et l'algorithme 2 l'explique pas à pas.

Algorithme 2

IIH : l'Horizon d'Identification Initial

δ : la tolérance sur la cohérence de la sortie observée avec la sortie prédite

DS : le jeu de données expérimental

N : l'horizon glissant du modèle F

1. Commencer

2. $debut \leftarrow 1$, $fin \leftarrow N$, $i \leftarrow 1$

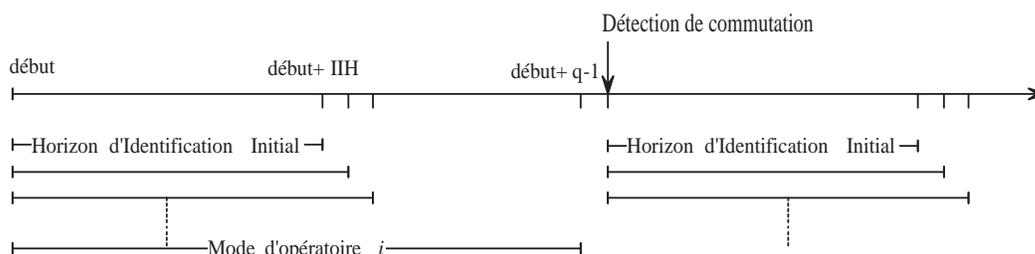


FIG. 4.9 – Partitionner le régime opératoire ; méthode cohérence entre la sortie observée et la sortie prédite.

3. Tant que $fin < (\text{le nombre des éléments de } DS)$

- (a) **Identifier** les paramètres du modèle F_i en utilisant $DS^{debut,fin}$
- (b) **Si** $[y]_{fin+1} \not\approx [env]_{fin+1}$
 - i. **Prendre** $DS^{debut,fin-1}$ les observations du $i^{\text{ème}}$ mode opératoire. **Utiliser** les paramètres identifiés en utilisant $DS^{debut,fin-1}$ pour le modèle F_i .
 - ii. $i \leftarrow i + 1$
 - iii. $debut \leftarrow fin$
 - iv. $fin \leftarrow debut + IIIH$
 - v. **Aller** à la ligne 3
- (c) **Sinon**
 - i. $fin \leftarrow fin + 1$
 - ii. **Aller** à la ligne 3

4. Fin

Dans cette méthode, pour les systèmes commutés et pendant l'expérience sur le système, le temps entre deux commutations consécutives doit comprendre au moins $IIIH$ instances puisqu'aucune commutation ne peut être détectée pendant l'horizon d'identification initial $IIIH$.

Pour utiliser cette méthode, deux paramètres supplémentaires doivent être choisis par rapport à la première approche :

L'horizon d'identification initial $IIIH$. Il doit être choisi tel qu'en utilisant les observations des entrées et des sorties du système sur cet horizon, le modèle local représente presque bien le comportement du système dans cette zone spécifique. Déterminer la longueur de cet horizon est un problème classique en automatique et on a besoin des informations sur le comportement du système ainsi que le jeu de données expérimental pour déterminer cet horizon.

La valeur de la tolérance δ . En pratique, malgré les précautions d'usage, le modèle local qui est identifié sur l'horizon IIH ne représente pas tous les comportements du système dans la zone spécifiée. La valeur δ sert à donner une tolérance au modèle local afin de lui permettre de s'adapter dans une certaine mesure aux points suivants qui sont dans le même mode opératoire et qui ne peuvent pas être expliqués par le modèle déjà identifié. Cependant, elle doit être assez petite pour que la méthode puisse détecter la commutation au bon moment et au bon endroit. Une approximation sur la valeur adéquate de δ peut être obtenue à partir d'une analyse préliminaire sur le jeu de données DS . Cependant, aucune solution n'est présentée pour trouver la valeur précise de ce paramètre.

Après chaque itération, la tolérance δ peut permettre une augmentation graduelle de l'incertitude sur les paramètres du modèle. Cela vient du fait qu'à l'instant k , si la sortie observée $[y]_k$ n'est pas incluse dans l'intervalle prédit par le modèle, on essaiera d'expliquer $[y]_k$ avec la tolérance δ et si ça se produit, on ajoutera un point dans le jeu de données du mode courant qui n'était pas expliqué par le modèle individuellement. Pour l'expliquer, il faut ajouter de l'incertitude sur les paramètres du modèle. Cette augmentation graduelle de l'incertitude sur les paramètres du modèle peut entraîner l'explosion progressive et incontrôlable du rayon de l'enveloppe englobante. Les deux paramètres supplémentaires à choisir ainsi que l'augmentation progressive de l'incertitude sur les paramètres du modèle conduisent à préférer la première méthode pour partitionner le régime opératoire du système en zones locales.

Par ces deux méthodes de partitionnement, on ne peut pas déterminer si le système entre dans un nouveau mode opératoire ou s'il rentre dans un mode déjà identifié. De ce fait, on peut avoir plusieurs modèles pour un comportement identique. Ce problème sera traité dans la section 4.5.4 en agrégeant les modèles de comportements similaires.

Exemple 4.5.1

Par exemple, on applique les méthodes de découpage de jeu de données sur un système mono-entrée mono-sortie.

Système. Le système à modéliser est décrit sous la forme suivante :

$$y_k = \sin(u_k) \tag{4.5.1}$$

pour

$$u_1 = -4, \quad u_{k+1} = u_k + 0, \quad 1 \leq k$$

Jeu de données. Dans cet exemple, l'entrée du système est certaine et la sortie est incertaine. Pour simuler l'erreur de mesure et pour être sûr que la sortie du système y_k est prise en compte dans les calculs, on ajoute l'intervalle $[-0,1 \ 0,1]$ sur la sortie observée $\epsilon_k + y_k$ où ϵ_k est

une variable aléatoire bornée entre $-0,1$ et $0,1$ représentant l'erreur de mesure. L'intervalle $[-0,1 \ 0,1]$ nous permet de définir un support où se trouve la valeur réelle (mais inconnue en pratique) de y_k :

$$[y]_k \leftarrow y_k + \epsilon_k + [-0,1 \ 0,1]$$

L'entrée reste certaine. Donc, le jeu de donnée est de la forme $DS = \{(u_k, [y]_k)\}$.

Structure de modèle. L'horizon glissant des modèles locaux est fixé à $N = 0$; c'est-à-dire que le modèle local F_i contient seulement un modèle statique.

Analyse des données. On fait une analyse préliminaire des données DS pour avoir une idée de la précision de l'enveloppe englobante à chaque instant ainsi que la tolérance sur la cohérence de la sortie observée avec la sortie prédite si aucune commutation ne se produit.

Dans un premier temps, on calcule la précision de l'enveloppe. A l'instant k , elle est égale à la précision de l'enveloppe englobante pour le jeu de données $DS^{1,k}$. Ensuite, la même procédure est répétée pour calculer la tolérance δ . A l'instant k , la tolérance δ_k sur la cohérence de la sortie observée $[y]_k$ avec la sortie prédite $[env]_k$ est égale à la valeur minimale de δ_k telle que $[y]_k \approx [env]_k$. Les courbes de ces deux analyses sont montrées sur la figure 4.10. Cette analyse préliminaire des données permet de choisir la valeur de la précision de l'enveloppe et/ou la valeur de la tolérance δ si elle ne peuvent pas être déterminées à partir de l'information a priori sur le système.

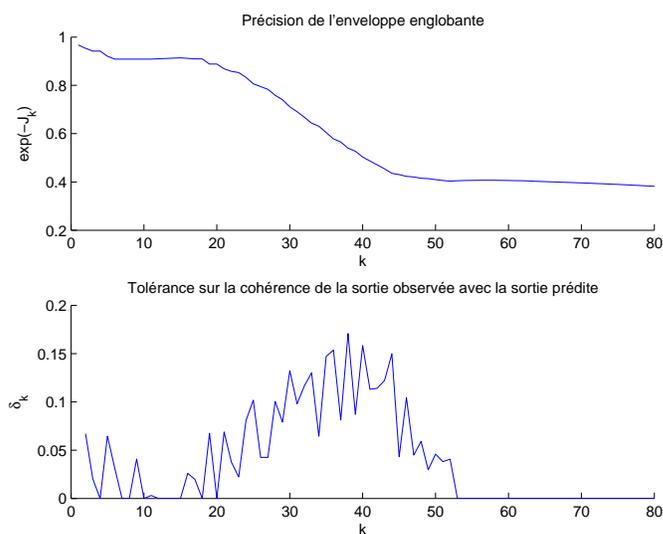


FIG. 4.10 – Analyse préliminaire des observations.

Méthode 1 : précision de l'enveloppe englobante. On fixe la précision désirée (le rayon moyen maximal de l'enveloppe englobante) à $pre = 0,8$ et on applique la méthode sur le jeu de données DS . Les trois zones détectées sont montrées sur la figure 4.11. Les précisions des modèles locaux et les rayons moyens des enveloppes $\{[env]\}$ des modèles locaux sont comme suit :

$$\text{zone 1 : } J_1 = 0,1979 \quad P_1 = 0,8204$$

$$\text{zone 2 : } J_2 = 0,2006 \quad P_2 = 0,8182$$

$$\text{zone 3 : } J_3 = 0,1903 \quad P_3 = 0,8267$$

La précision de tous les modèles locaux est supérieure à 0,8. En considérant la définition 4.3.1, la précision du modèle multi-mode est égale à :

$$P = \min_i(P_i) = 0,8182$$

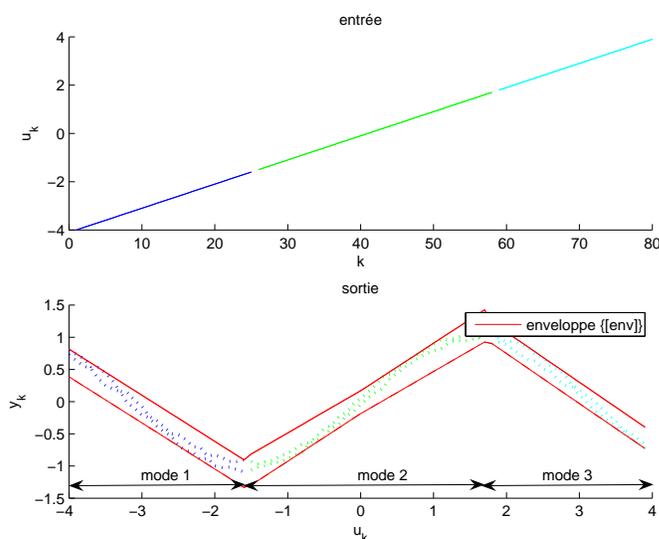


FIG. 4.11 – Partitionnement de régime opératoire du système (première méthode).

Méthode 2 : cohérence de la sortie observée avec la sortie prédite.

Sur la figure 4.12, le même système est découpé en appliquant la deuxième méthode. IIH est égal à 2 et la valeur de la tolérance δ est fixée à 0,12. Par la deuxième méthode, on détecte aussi trois zones opératoires. Les mesures de qualité de l'enveloppe sont :

zone 1 : $J_1 = 0,2251$ $P_1 = 0,7984$

zone 2 : $J_2 = 0,1887$ $P_2 = 0,8280$

zone 3 : $J_3 = 0,1965$ $P_3 = 0,8216$

et donc, la précision du modèle multi-mode est égale à :

$$P = \min_i(P_i) = 0,7984$$

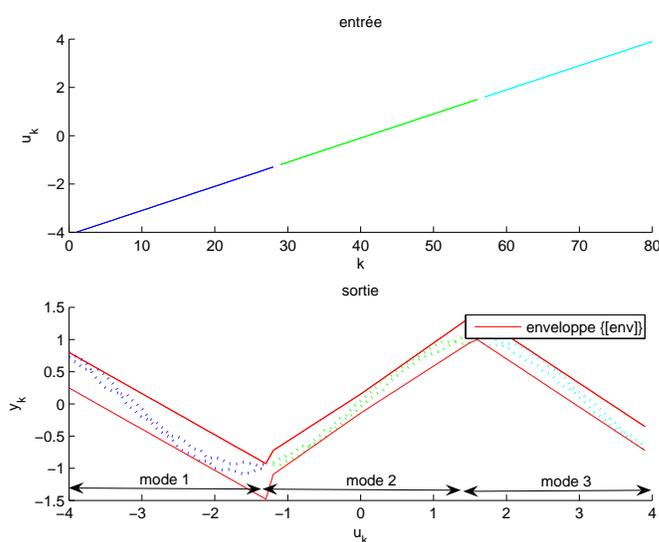


FIG. 4.12 – partitionnement de régime opératoire du système (deuxième méthode).

4.5.3 Détermination des domaines de validité des modèles locaux

On a vu que le domaine de validité d'un modèle local est le domaine dans lequel le modèle explique le comportement du système. Pour déterminer ces domaines, dans un premier temps, on définira l'espace dans lequel ces domaines peuvent être déterminés et ensuite, on présentera une technique pour calculer les frontières de ces domaines. On étudiera également la possibilité de diminuer les dimensions de l'espace de validité pour simplifier les calculs pendant la simulation.

Espace de validité d'un modèle

L'espace $\chi_{dimension}$ est un espace dans lequel le domaine de validité d'un modèle donné peut être déterminé. L'indice *dimension* explicite les dimensions de cet espace. Un point

dans l'espace $\chi_{dimension}$ est noté par le symbole ξ . On précise que pour un modèle donné, cet espace n'est pas unique.

Dans ce document, on a supposé que toutes les variables du modèle sont les variables caractéristiques et donc, si les variables du modèle expliquent le comportement du système, alors le domaine de validité du modèle peut être déterminé dans l'espace de ces variables; c'est-à-dire dans l'espace du vecteur de régression du modèle. De ce fait et dans un premier temps, $\chi_{dimension}$ est défini comme l'espace du vecteur de régression du modèle :

$$\chi_{dimension} = \chi_{\varphi}$$

On verra par la suite que la dimension de cet espace peut être réduite si le domaine de validité du modèle est une enveloppe convexe linéaire dans l'espace d'état observable.

Domaine de validité d'un modèle local

Après avoir divisé le jeu de données en sous-ensembles DS_i , l'espace χ_{φ} sera partitionné en domaines H_{φ}^i tels que le domaine H_{φ}^i détermine le domaine de validité du modèle local i . Pour cela, on utilise DS_i .

L'ensemble des sorties du système dans le sous-ensemble DS_i détermine une zone locale du régime opératoire du système. On a vu pendant la procédure de partitionnement que le comportement du système (la sortie du système) dans cette zone peut être représentée assez précisément par le modèle local i .

Chaque point $y_k \in DS_i$ de cette zone est une estimation "précise" de l'image de φ_k par rapport à la fonction du modèle où le vecteur φ_k contient des éléments du sous-ensemble DS_i . Les éléments de DS_i conduisent donc à un certain nombre de points φ_k dans l'espace χ_{φ} qui définissent une région dans cet espace. Donc, la région H_{φ}^i constituée par ces points peut être considérée comme une estimation du domaine de validité du modèle i . Cela est montré sur la figure 4.13.

Définition 4.5.3. *Le domaine de validité H_{φ}^i du modèle local i est défini par l'enveloppe convexe autour des points φ_k ou des boîtes $[\varphi]_k$ ³ dans l'espace χ_{φ} . Le vecteur de régression φ_k contient des éléments de DS_i .*

Dans le cas où les observations sont certaines, les φ_k sont des points dans l'espace χ_{φ} et H_{φ}^i est l'enveloppe convexe définie par ces points. Cependant, si les observations sont incertaines, le vecteur de régression est un vecteur intervalle qui représente une boîte dans l'espace χ_{φ} . Puisque la valeur exacte de ce vecteur est un point inconnu dans la boîte $[\varphi]_k$, pour garantir que ce point est pris on compte, H_{φ}^i sera l'enveloppe convexe autour des boîtes $[\varphi]_k$.

³Cela arrive dans le cas où les observations sont incertaines.

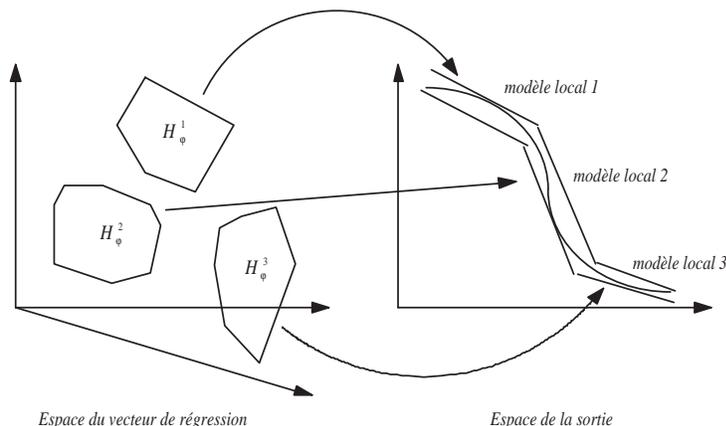


FIG. 4.13 – Espace de vecteur de régression et les domaines de validité des modèles locaux.

Tout d’abord, on commence par le cas le plus simple, le modèle statique, et on déterminera son domaine de validité. Ensuite, on déterminera le domaine de validité des modèles dynamiques relation entrée-sortie et réponse impulsionnelle à horizon fini. Dans la dernière partie, cette définition sera généralisée pour les modèles *FIO* et *FFIR*.

Modèle statique. Dans un modèle statique, le vecteur de régression ne contient que l’entrée du modèle à l’instant courant. Ce vecteur a un seul élément et par conséquent, l’espace χ_φ a une seule dimension. Le domaine de validité du modèle i est l’enveloppe convexe H_φ^i des entrées u_k qui sont dans l’ensemble DS_i . Ce domaine peut être déterminé par un ensemble d’inégalités. Dans l’exemple 4.5.1, les domaines de validité des modèles locaux détectés par la première méthode de découpage (figure 4.11) sont déterminés par :

$$\begin{aligned} \text{Mode 1 : } H_\varphi^1 &= \{u \mid -4 \leq u < -1, 6\} \\ \text{Mode 2 : } H_\varphi^2 &= \{u \mid -1, 6 \leq u < 1, 7\} \\ \text{Mode 3 : } H_\varphi^3 &= \{u \mid 1, 7 \leq u < 4\} \end{aligned}$$

et les domaines de validité des modèles locaux détectés par la deuxième méthode de découpage (figure 4.12) sont déterminés par :

$$\begin{aligned} \text{Mode 1 : } H_\varphi^1 &= \{u \mid -4 \leq u < -1, 3\} \\ \text{Mode 2 : } H_\varphi^2 &= \{u \mid -1, 3 \leq u < 1, 6\} \\ \text{Mode 3 : } H_\varphi^3 &= \{u \mid 1, 6 \leq u < 4\} \end{aligned}$$

Modèle relation entrée-sortie. On a vu dans la section 3.3.2 qu’un modèle dynamique *ARX* peut être interprété comme un modèle statique si on considère l’espace

du vecteur de régression du modèle comme l'espace d'entrée de celui-ci. Par conséquent, le domaine de validité d'un modèle dynamique relation entrée-sortie est aussi défini comme le domaine de validité d'un modèle statique sauf que dans ce cas, le vecteur de régression contient des entrées et des sorties décalées dans le temps. Par exemple, pour un modèle du premier ordre, l'espace $\chi_\varphi = \chi_{\{y_{k-1}, u_{k-1}\}}$ a deux dimensions, pour un modèle du deuxième ordre, l'espace $\chi_\varphi = \chi_{\{y_{k-1}, y_{k-2}, u_{k-1}, u_{k-2}\}}$ a quatre dimensions etc. Dans le cas où l'ordre du système est élevé, le calcul de l'enveloppe convexe est très souvent lourd à réaliser. Pour cette raison, on étudiera par la suite la possibilité de réduire la dimension de l'espace χ .

La notion de commutation d'un modèle multi-mode est classique dans le cas d'un modèle d'état. Dans le cas d'un modèle à horizon fini, à cause de l'effet de mémoire, des précautions sont à prendre, ce qui s'est traduit dans ce document par l'utilisation d'un banc de modèles à horizons glissants croissants. Nous avons donc cherché à étudier le lien qui peut exister entre les domaines de validité d'un modèle d'état et d'un modèle relation entrée-sortie et un modèle réponse impulsionnelle.

Selon l'hypothèse où le domaine de validité d'un modèle relation entrée-sortie se traduit par une enveloppe convexe linéaire dans l'espace d'état observable (figure 4.14), on arrive alors à réduire la dimension de l'espace χ . On commence par un système du premier ordre où l'entrée u_k et la sortie y_k du système sont des scalaires. A priori, l'espace de validité du modèle est l'espace du vecteur de régression $\{y_{k-1}, u_{k-1}\}$. Si on décrit le comportement du système sous la forme d'une représentation d'état observable, on aura :

$$\begin{aligned} x_{k+1} &= a \times x_k + b \times u_k \\ y_k &= x_k \end{aligned} \quad (4.5.2)$$

où $a, b \in R$ sont les paramètres et $x \in R$ représente l'état du système. Selon l'hypothèse ci-dessus, le domaine de validité du modèle i dans l'espace $\chi_{\{x\}}$ est l'enveloppe convexe $H_{\{x\}}^i$. Ce domaine peut être déterminé par un ensemble de contraintes inégalité linéaires ; c'est-à-dire :

$$H_{\{x\}}^i = \{x \mid \forall j, \alpha_{(i,j)} \times x + \gamma_{(i,j)} \leq 0\} \quad (4.5.3)$$

où le j est l'indice de la contrainte. Le domaine de validité $H_{\{x\}}^i$ est défini par les coefficients $\alpha_{(i,j)}, \gamma_{(i,j)} \in R$. Par exemple, si

$$-5 \leq x \leq 7$$

est le domaine de validité du modèle local i , ce domaine peut être déterminé par l'ensemble de contraintes inégalité comme suit :

$$\begin{aligned} -x + 5 &\leq 0 \\ x - 7 &\leq 0 \end{aligned}$$

où

$$\alpha_{(i,1)} = -1, \quad \gamma_{(i,1)} = 5$$

$$\alpha_{(i,2)} = 1, \quad \gamma_{(i,2)} = -7$$

De l'équation 4.5.2, on a :

$$x_k = y_k$$

et en le substituant dans 4.5.3, on peut calculer l'image du domaine de validité $H_{\{x\}}^i$ dans l'espace $\chi_{\{y_k\}}$:

$$H_{\{y_k\}}^i = \{y_k \mid \forall j, \alpha_{(i,j)} \times y_k + \gamma_{(i,j)} \leq 0\} \quad (4.5.4)$$

Les équations 4.5.3 et 4.5.4 montrent que l'enveloppe H^i peut être déterminée à la fois dans l'espace $\chi_{\{y_{k-1}, u_{k-1}\}}$ (l'espace de régression), $\chi_{\{x\}}$ (l'espace d'état) et $\chi_{\{y_k\}}$ (l'espace entrée-sortie réduit).

Pour continuer, on suppose que le système est un système de deuxième ordre. A priori, l'espace de validité du modèle est l'espace du vecteur de régression ; c'est-à-dire $\chi_\varphi = \chi_{\{y_{k-2}, y_{k-1}, u_{k-2}, u_{k-1}\}}$. Si on décrit le comportement du système sur la forme d'une représentation d'état observable, on aura :

$$\begin{aligned} \begin{bmatrix} x_{(1,k+1)} \\ x_{(2,k+1)} \end{bmatrix} &= \begin{bmatrix} 0 & a_2 \\ 1 & a_1 \end{bmatrix} \times \begin{bmatrix} x_{(1,k)} \\ x_{(2,k)} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \times u_k \\ y_k &= [0 \ 1] \times \begin{bmatrix} x_{(1,k)} \\ x_{(2,k)} \end{bmatrix} \end{aligned} \quad (4.5.5)$$

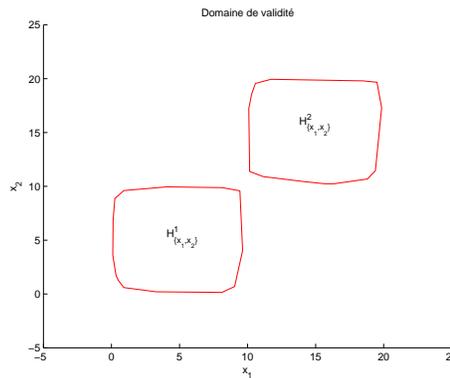


FIG. 4.14 – Domaines de validité dans l'espace d'état.

Selon l'hypothèse ci-dessus, le domaine de validité du modèle i dans l'espace $\chi_{\{x_1, x_2\}}$ est l'enveloppe convexe $H_{\{x_1, x_2\}}^i$. Ce domaine de validité peut être déterminée par un ensemble de contraintes inégalité linéaires dont chaque contrainte détermine une droite, qui est une frontière de $H_{\{x_1, x_2\}}^i$, dans l'espace $\chi_{\{x_1, x_2\}}$; c'est-à-dire :

$$H_{\{x_1, x_2\}}^i = \{(x_1, x_2) \mid \forall j, [\alpha_{(i,j)} \beta_{(i,j)}] \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \gamma_{(i,j)} \leq 0\} \quad (4.5.6)$$

où j est l'indice de la contrainte. La droite j est définie par les coefficients $\{\alpha_{(i,j)}, \beta_{(i,j)}, \gamma_{(i,j)}\}$ et l'ensemble des droites détermine $H_{\{x_1, x_2\}}^i$ dans l'espace d'état. En substituant l'équation 4.5.5 dans l'équation 4.5.6, on peut calculer l'image du domaine de validité $H_{\{x_1, x_2\}}^i$ dans l'espace $\chi_{\{y_k, y_{k-1}, u_{k-1}\}}$:

$$H_{\{y_k, y_{k-1}, u_{k-1}\}}^i = \{(y_k, y_{k-1}, u_{k-1}) \mid \forall j, [\alpha_{(i,j)} \beta_{(i,j)}] \times \begin{bmatrix} 0 & a_2 & b_1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} y_k \\ y_{k-1} \\ u_{k-1} \end{bmatrix} + \gamma_{(i,j)} \leq 0\} \quad (4.5.7)$$

et si dans l'équation 4.5.7, on substitue la valeur de y_k en fonction des entrées et des sorties décalées, on arrive à calculer l'image de $H_{\{x_1, x_2\}}^i$ dans l'espace $\chi_{\{y_{k-1}, y_{k-2}, u_{k-1}, u_{k-2}\}}$:

$$H_{\{y_{k-1}, y_{k-2}, u_{k-1}, u_{k-2}\}}^i = \{(y_{k-1}, y_{k-2}, u_{k-1}, u_{k-2}) \mid \forall j, [\alpha_{(i,j)} \beta_{(i,j)}] \times \begin{bmatrix} a_2 & 0 & b_1 & 0 \\ a_1 & a_2 & b_2 & b_1 \end{bmatrix} \times \begin{bmatrix} y_{k-1} \\ y_{k-2} \\ u_{k-1} \\ u_{k-2} \end{bmatrix} + \gamma_{(i,j)} \leq 0, \forall j\} \quad (4.5.8)$$

Les équations 4.5.6, 4.5.7 et 4.5.8 montrent que le domaine de validité H^i peut être déterminé à la fois dans l'espace $\chi_{\{y_{k-1}, y_{k-2}, u_{k-1}, u_{k-2}\}}$ (l'espace du vecteur de régression), $\chi_{\{x_1, x_2\}}$ (l'espace d'état) et $\chi_{\{y_k, y_{k-1}, u_{k-1}\}}$ (l'espace entrée-sortie réduit).

De manière générale, si on sait a priori que le domaine de validité d'un modèle relation entrée-sortie d'ordre N est une enveloppe convexe linéaire de la forme suivante :

$$H_{\{x_1, \dots, x_N\}}^i = \{(x_1, \dots, x_N) \mid \forall j, [\alpha_{1,(i,j)} \dots \alpha_{N,(i,j)}] \times \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} + \gamma_{(i,j)} \leq 0\}$$

dans l'espace d'état observable, ce domaine peut aussi être déterminé dans l'espace réduit

$$\mathcal{X}\{y_k, \dots, y_{k-N-1}, u_{k-1}, \dots, u_{k-N-1}\}$$

au lieu de l'espace du vecteur de régression

$$\mathcal{X}\{y_{k-1}, \dots, y_{k-N}, u_{k-1}, \dots, u_{k-N}\}$$

Modèle réponse impulsionnelle à horizon fini. Un modèle réponse impulsionnelle à horizon fini est un cas particulier d'un modèle *ARX*. Donc, en considérant les arguments donnés au début de cette section, le domaine de validité de ce modèle peut aussi être déterminé a priori dans l'espace du vecteur de régression du modèle ; c'est-à-dire :

$$\mathcal{X}\{u_{k-1}, u_{k-2}, \dots, u_{k-HG}\}$$

où *HG* est l'horizon glissant du modèle. On définit donc le domaine de validité H_φ^i du modèle local *i* par l'enveloppe convexe autour des points φ_k ou des boîtes $[\varphi]_k$ dans l'espace \mathcal{X}_φ . Le vecteur φ_k contient des éléments de DS_i .

Dans un modèle réponse impulsionnelle à horizon fini, par rapport à un modèle relation entrée-sortie, l'ordre de l'horizon glissant est plus élevé et par conséquent, le calcul de domaines de validité H_φ^i est plus complexe et nécessite un temps de calcul plus élevé. Ceci est un des principaux inconvénients d'un modèle réponse impulsionnelle à horizon fini par rapport à un modèle relation entrée-sortie dans le contexte de l'identification multi-mode. Pourtant, cet inconvénient peut être éliminé si on connaît l'ordre de la dynamique du système. Dans ce cas, au lieu de déterminer les domaines de validité des modèles locaux dans l'espace du vecteur de régression

$$\mathcal{X}\{u_{k-1}, u_{k-2}, \dots, u_{k-HG}\}$$

ces domaines peuvent être déterminés dans l'espace entrée-sortie

$$\mathcal{X}\{y_{k-1}, \dots, y_{k-N}, u_{k-1}, \dots, u_{k-N}\}$$

ou l'espace entrée-sortie réduit

$$\mathcal{X}\{y_k, \dots, y_{k-N-1}, u_{k-1}, \dots, u_{k-N-1}\}$$

où *N* est l'ordre de la dynamique du système. Cela permet d'utiliser un modèle réponse impulsionnelle à horizon fini avec ses avantages⁴ dans le contexte de

⁴On a déjà parlé des avantages d'un modèle *FFIR* par rapport d'un modèle *FIO* pour prédire l'enveloppe englobante de la sortie du système dans le chapitre précédent.

l'identification et de la simulation multi-mode en même temps que d'éviter l'inconvénient de déterminer le domaine de validité de ses modèles locaux dans un espace avec un grand nombre de dimensions.

Modèle FIO et $FFIR$. Un modèle FIO (respect. $FFIR$) consiste en un modèle statique et N modèles dynamiques relation entrée-sortie (respect. réponse impulsionnelle horizon fini) avec des horizons glissants qui varient entre $HG = 1$ et $HG = N$. Chacun des modèles auxiliaires a son propre domaine de validité.

Définition 4.5.4. On définit le domaine de validité d'un modèle FIO (resp. $FFIR$) par le domaine de validité de son modèle relation entrée-sortie (respect. réponse impulsionnelle à horizon fini) avec l'horizon glissant le plus long; c'est à dire : $HG = N$.

On choisit cette définition puisque le modèle auxiliaire avec l'horizon glissant $HG = N$ est le modèle le plus précis et qui a un poids plus important par rapport aux autres modèles auxiliaires.

Remarque. Il arrive parfois que les domaines de validité H_χ^i s'intersectent. Cela peut être dû aux deux raisons suivantes :

Incertitude sur les observations. Si les éléments du jeu de données sont incertains, dans l'espace χ , on calcule l'enveloppe convexe de boîtes au lieu de points. Le fait de mal connaître les points à expliquer génère de l'incertitude sur les domaines de validité. Cette prise en compte de l'incertitude génère une surestimation de ces domaines pouvant causer l'intersection des domaines H_χ^i . Ce phénomène est montré sur un exemple de la figure 4.15. On voit sur cette figure que les enveloppes pour les observations exactes ne s'intersectent pas. Cependant, les enveloppes pour les mêmes observations dans le cas où elles sont imprécises conduit à l'intersection non-vide des domaines de validité.

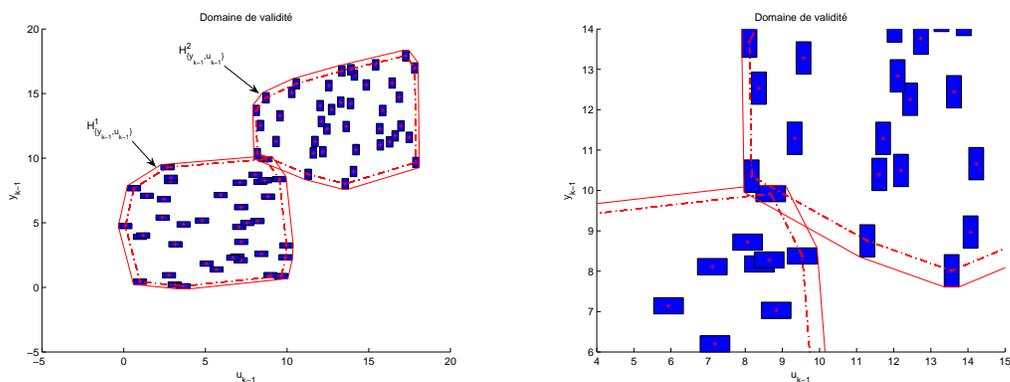


FIG. 4.15 – Intersection des domaines de validité à cause de l'imprécision des observations.

Détection de la commutation d'un système multi-mode à un mauvais endroit ou à un instant erroné. Pendant le partitionnement du régime opératoire du système (section 4.5.2), un mauvais choix ou des valeurs inadaptées de la précision désirée et de la tolérance sur la cohérence de la sortie observée avec la sortie prédite peuvent conduire à un mauvais partitionnement de DS . C'est surtout le cas pour un système multi-mode pour lequel un point appartient à un mode opératoire précis. Pendant le partitionnement, si le système entre dans un nouveau mode, les premiers points du nouveau mode, à cause de ce mauvais choix, peuvent être expliqués par le modèle du mode précédent et par conséquent, ils seront classés dans le jeu de données du mode précédent. Ce phénomène cause l'inclusion de régions supplémentaires dans les domaines de validité des modèles locaux qui ne correspondent pas à ce modèle (figure 4.23). Pour l'éviter, après avoir divisé DS en sous-ensembles DS_i , les N premiers et les N derniers points de tous les sous-ensembles DS_i sont extraits et les sous-ensembles écourtés seront utilisés pour identifier les paramètres des modèles F_i ainsi que déterminer leurs domaines de validité (la figure 4.24). Puisque le nombre de points dans les sous-ensembles DS_i est assez important, l'élimination de $2N$ points ne pose pas de problème. Cela est montré sur la figure 4.16.

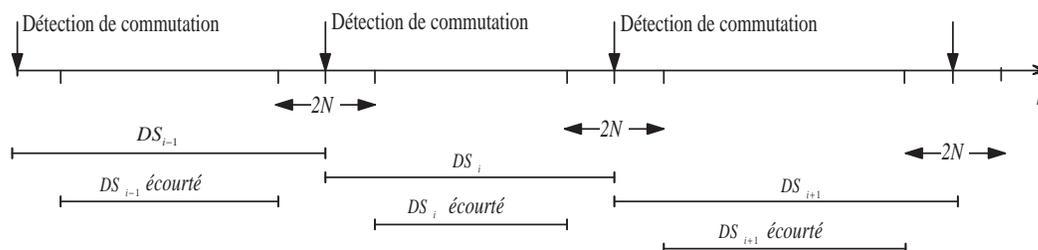


FIG. 4.16 – Éliminer les points autour des commutations détectées.

◇

4.5.4 Agrégation des modèles

Comme l'on a déjà expliqué dans la section 4.5.2, pendant l'expérience sur un système, celui-ci peut passer plusieurs fois dans le même mode opératoire i . Si la commutation est déterminée, toutes les mesures qui correspondent à ce mode sont agrégées et sont utilisées pour identifier les paramètres du modèle local F_i . Cependant, si la commutation est indéterminée, rassembler et agréger les mesures du mode opératoire i qui sont dispersées dans le temps⁵ est difficile à réaliser puisque pendant la procédure de partitionnement de l'ensemble DS et après avoir détecté un nouveau mode, on ne sait pas, a priori, si on va entrer dans un nouveau mode ou dans un mode qui a été déjà détecté. En principe, avoir plusieurs modèles pour un mode opératoire ne pose pas de problème. Pourtant

⁵Voir la figure 4.4

pour réduire le nombre des modèles locaux dans le modèle multi-mode, on étudiera la possibilité d'agréger des modèles similaires.

La méthode d'agrégation proposée dans ce document est basée sur le domaine de validité des modèles locaux. Après avoir identifié les modèles F_i et calculé leur domaines de validité, les deux domaines $\square H_\chi^{i1}$ et $\square H_\chi^{i2}$ qui représentent les domaines de validité du modèle F_{i1} et du modèle F_{i2} peuvent prendre trois positions différentes l'une par rapport à l'autre :

- $\square H_\chi^{i1}$ et $\square H_\chi^{i2}$ sont deux domaines de validité disjoints. Dans ce cas, F_{i1} et F_{i2} sont deux modèles différents car ils correspondent à deux modes différents.
- $\square H_\chi^{i1}$ est un sous ensemble de $\square H_\chi^{i2}$. Dans ce cas, si le modèle F_{i1} est valide, le modèle F_{i2} est aussi valide. Donc, les modèles F_{i1} et F_{i2} représentent un mode opératoire identique. On élimine le modèle F_{i1} et on agrège le jeu de donnée DS_{i1} avec celui du deuxième modèle DS_{i2} . Les paramètres du modèle F_{i2} sont ensuite re-identifiés en utilisant $DS_{i1} \cup DS_{i2}$. $\square H_\chi^{i2}$ déterminera le domaine de validité du nouveau modèle.
- sinon, $\square H_\chi^{i1}$ et $\square H_\chi^{i2}$ s'intersectent. Cependant, aucun n'est un sous-ensemble de l'autre. Dans ce cas, on considère qu'il s'agit de deux modèles qui représentent deux modes opératoires différents.

En utilisant ces trois règles, on est généralement capable d'agréger des modèles similaires et de diminuer le nombre de modèles locaux dans le modèle multi-mode.

◇

L'agrégation des modèles similaires est l'étape finale de la procédure d'identification multi-mode. On résume cette procédure comme suit :

1. On choisit, tout d'abord, la structure et l'ordre des modèles locaux F_i à partir de la connaissance a priori sur le système et les résultats des analyses préliminaires sur les observations.
2. On partitionne le régime opératoire du système en appliquant le premier ou le deuxième algorithme de découpage sur les données d'identification DS . Dans cette étape, les paramètres des modèles locaux F_i sont identifiés en parallèle avec la procédure de partitionnement.
3. En fonction de la dynamique du système, on détermine la dimension de l'espace χ et on calcule les domaines de validité des modèles locaux dans cet espace.
4. On étudie la possibilité d'agréger des modèles similaires en considérant les domaines de validité des modèles locaux.

Dans les exemples qui suivent, pour simplifier :

- On suppose qu'on connaît l'ordre de la dynamique du système. Par conséquent, on détermine les domaines de validité des modèles dans l'espace $\mathcal{X}\{y_k, \dots, y_{k-n_a-1}, u_{k-1}, \dots, u_{k-n_a-1}\}$ indépendamment de la structure qu'on choisit pour les modèles locaux (le modèle *FIO* ou le modèle *FFIR*).
- Après avoir déterminé les domaines de validité des modèles locaux, on surestime les domaines de validité H_χ^i par leurs boîtes englobantes $\square H_\chi^i$.

4.6 Simulation Multi-Modes : commutation indéterminée

Après avoir identifié le modèle multi-mode, ce modèle peut être utilisé dans diverses applications pour analyser le système comme le diagnostic et l'analyse de sûreté. Dans ce document, comme il a été expliqué dans les sections 3.4 et 4.4, l'objectif est de prédire l'enveloppe englobante de la sortie du système pour une séquence des entrées qui est donnée par $U = \{[u]_1 [u]_2 [u]_3 \dots [u]_{ph}\}$ où ph définit l'horizon de prédiction.

4.6.1 Sémantique de simulation

Comme l'on a déjà expliqué, dans le cadre de l'analyse de sûreté, on souhaite englober toutes les valeurs possibles de la sortie de système par une enveloppe "englobante". Pour un modèle mono-mode, la sémantique de simulation est donnée par les équations 3.4.1 et 3.4.2. En ce qui concerne le modèle multi-mode et le modèle courant valide i , la même sémantique de simulation sera utilisée pour calculer l'enveloppe englobante de la sortie du système.

4.6.2 Calcul de l'enveloppe englobante

Dans le cas où la commutation est indéterminée, la structure du modèle multi-mode est similaire à celle de la figure 4.5 à l'exception de l'absence du signal de commutation. Dans le cas où le signal de commutation existe, celui-ci détermine quel mode est actif et à chaque instant, le modèle qui correspond au mode actif détermine l'enveloppe englobante. Dans le cas où ce signal de commutation n'existe pas, la validité d'un modèle local est déterminée à l'aide des entrées et des sorties du modèle local ainsi que son domaine de validité. On définira tout d'abord la condition de validité d'un modèle local et on expliquera en suite, comment cette condition peut être vérifiée.

Dans la littérature, le problème de détection de mode actif est plus souvent traité dans le contexte de l'estimation d'état de système par des observateurs hybrides. Dans ce contexte, les travaux de [Balluchi et al., 2002], [Ragot et al., 2003], [Babaali & Egerstedt,

2005] et [Domlan, 2006] peuvent être cités. Dans ces travaux, le mode actif est déterminé en utilisant les mesures sur le système et à l'aide des outils de diagnostic. Pourtant, dans le contexte de la prédiction de comportement de système, les mesures sur le système (la sortie du système) ne sont pas disponibles et par conséquent, les méthodes proposées dans le cadre de l'estimation d'état ne peuvent pas être appliquées sans un certain nombre de modifications. De plus, les modèles utilisés dans ces travaux sont toujours des modèles déterministes, alors qu'on utilise un modèle intervalle pour représenter le comportement du système. Ces sont les raisons pour lesquelles on développe par la suite une méthode spécifique particulière pour déterminer le mode actif d'un modèle multi-mode intervalle.

Condition de validité d'un modèle local

On suppose que le domaine de validité du modèle local i est déterminé dans l'espace $\chi^{dimension}$ par H_χ^i . A l'instant k , ξ_k , qui est un point de l'espace $\chi^{dimension}$, contient des entrées et des sorties du système :

$$\begin{aligned}
 \chi\{u_k\} &\rightarrow \xi_k = [u_k] \\
 \chi\{u_{k-1}, u_{k-2}, \dots, u_{k-N}\} &\rightarrow \xi_k = [u_{k-1}, u_{k-2}, \dots, u_{k-N}] \\
 \chi\{y_{k-1}, \dots, y_{k-N}, u_{k-1}, \dots, u_{k-N}\} &\rightarrow \xi_k = [y_{k-1}, \dots, y_{k-N}, u_{k-1}, \dots, u_{k-N}] \\
 \chi\{y_k, \dots, y_{k-N-1}, u_{k-1}, \dots, u_{k-N-1}\} &\rightarrow \xi_k = [y_k, \dots, y_{k-N-1}, u_{k-1}, \dots, u_{k-N-1}] \quad (4.6.1)
 \end{aligned}$$

Dans le cas où les sorties y_k du système ne sont pas disponibles, les sorties prédites $[env]_k^i$ ou $[P.env]_k^i$ par le modèle local i les remplacent en fonction de ses disponibilités et on obtient ξ_k^i . A l'instant k , le modèle i est valide si et seulement si le vecteur ξ_k^i se trouve dans le domaine de validité H_χ^i , c'est-à-dire :

$$\xi_k^i \in H_\chi^i \quad (4.6.2)$$

Si le vecteur ξ_k^i contient les supports des variables bornées, c'est alors un vecteur intervalle $[\xi]_k^i$. Puisque la valeur exacte du vecteur ξ_k^i est une valeur inconnue dans le vecteur intervalle $[\xi]_k^i$, pour garantir que le modèle i est potentiellement actif, on surestime la validité du modèle et on conclut qu'il est valide si :

$$[\xi]_k^i \cap H_\chi^i \neq \phi \quad (4.6.3)$$

Autrement dit, le modèle i est valide à l'instant k s'il existe au moins un point commun entre $[\xi]_k^i$ et H_χ^i .

Définition 4.6.1. *Un modèle FIO (respect. FFIR) est valide si son modèle auxiliaire IO (respect. FIR) avec $HG=N$ est valide.*

Il faut noter que :

- Le test d’intersection entre le vecteur $[\xi]_k^i$ et le domaine de validité H_χ^i est souvent difficile à réaliser. Pour cette raison, on remplace par la suite le domaine de validité H_χ^i par la plus petite boîte la contenant $\square H_\chi^i$. Cette approximation extérieure introduit des régions qui ne sont pas dans le domaine de validité H_χ^i et par conséquent, le modèle i sera considéré actif à des endroits supplémentaires. Ceci est une source de surestimation de l’enveloppe englobante de la sortie du modèle multi-mode mais permet de réduire le temps de calcul pendant la simulation.
- Les vecteurs $[\xi]_k^i$ obtenus par la prédiction des modèles locaux peuvent faire valider plusieurs modèles locaux à l’instant k . Autrement dit, plusieurs modèles locaux peuvent être valides à chaque instant.
- On garantit que si le système est dans un mode opératoire, le modèle qui correspond à ce mode est valide. En revanche, si le modèle d’un mode est valide, on ne peut pas garantir que le système fonctionne réellement dans ce mode.

Déterminer les modèles actifs par les blocs ACK

Pour déterminer les modèles actifs à l’instant $k = 1$, on suppose que tous les modèles locaux sont potentiellement actifs. On réalise une prédiction sur l’horizon glissant des modèles locaux F_i comme il a été expliqué dans les sections 3.4.3 et 3.4.4 et on construit les vecteurs $[\xi]_{k+N}^i$ pour chacun de ces modèles. Le résultat de l’intersection du vecteur $[\xi]_{k+N}^i$ et du domaine de validité $\square H_\chi^i$ détermine si le modèle local F_i est valide ou non. Pour le modèle valide F_i , tant que cette intersection n’est pas vide, on continue de faire la prédiction aux instants suivants $k = 2, 3, \dots$. Pour les modèles non-valides, on répète la prédiction par le modèle à chaque instant k (recommencer la prédiction par le modèle statique à l’instant k suivit par les modèles dynamiques du banc) et dès que l’intersection $[\xi]_{k+N}^i$ et $\square H_\chi^i$ est non-vide, le modèle F_i devient actif. Pour réaliser cette vérification systématiquement, et pour déterminer si un mode est actif ou non, on propose une stratégie pour représenter le modèle valide par un automate à deux états. Le bloc pour le mode opératoire i s’appelle ACK_i .

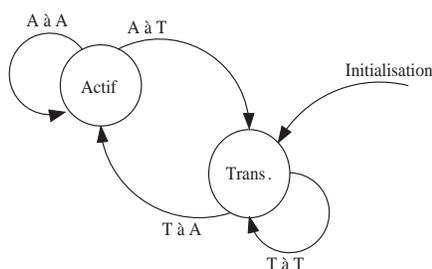


FIG. 4.17 – Automate à deux états ACK_i .

Chacun des blocs ACK_i est indépendant des autres et son état se met à jour à tous les instants d’échantillonnage en fonction des conditions sur les transitions. De ce fait

et à un instant donné, aucun, un ou plusieurs modes peuvent être déterminés actifs. Ci-après, les états et les transitions de cet automate sont expliqués.

Initialisation : Tous les blocs ACK_i entrent dans l'état transitoire au début de la simulation.

État actif : A l'instant k , si ACK_i est dans l'état actif, le modèle F_i est valide et il continue à prédire l'enveloppe englobante de la sortie du système.

État transitoire : A l'instant k , si ACK_i est dans l'état transitoire, le modèle F_i est inactif et la sortie prédite par le modèle F_i n'est pas prise en compte.

Transition transitoire-à-actif : A l'instant k , pour vérifier si le modèle F_i passe de l'état transitoire à l'état actif, on effectue une prédiction par le modèle F_i jusqu'à l'instant $k + N$ en supposant qu'on vient d'entrer dans le mode i et on construit le vecteur ξ_{k+N}^i (équation 4.6.1). Cette procédure a été expliquée dans la section 3.4.3 pour un modèle $FFIR$ et dans la section 3.4.4 pour un modèle FIO . Si l'intersection entre le vecteur ξ_{k+N}^i et H_X^i n'est pas vide, à l'instant $k + 1$ le bloc ACK_i passe à l'état actif. A cet instant, le modèle F_i commence à calculer l'enveloppe par son modèle statique qui sera poursuivit aux instants suivants par les modèles dynamiques du banc des modèles auxiliaires.

Une intersection non-vide entre le vecteur ξ_{k+N}^i et le domaine de validité H_X^i montre qu'avec la séquence des entrées donnée, il existe au moins une trajectoire pour la sortie du système qui peut activer le modèle auxiliaire $HG = N$ du modèle local i à l'instant $k + N$; ceci est en accord avec la définition 4.6.1. Autrement dit, il existe une cohérence entre la prédiction de la sortie par le modèle local F_i de l'instant $k + 1$ à l'instant $k + N$ et son domaine de validité. Cette cohérence nous permet de conclure que ce mode peut être potentiellement actif. Cependant, on ne garantit pas qu'il est réellement actif.

Transition transitoire-à-transitoire : Si la condition de la transition *transitoire-à-actif* n'est pas satisfaite, le bloc ACK_i reste dans l'état transitoire.

Transition actif-à-transitoire : Si à l'instant k , l'intersection entre le vecteur ξ_k^i et le domaine de validité de modèle F_i est vide, à l'instant $k + 1$ le bloc ACK_i passe à l'état transitoire.

Transition actif-à-actif : Si la condition de la transition *actif-à-transitoire* n'est pas satisfaite, le bloc ACK_i reste dans l'état actif.

Simulateur de modèle multi-mode

Le simulateur de modèle multi-mode contient deux types d'éléments :

- Les modèles locaux F_i .
- Les blocs ACK_i qui déterminent si un mode opératoire est actif ou non.

Le schéma de ce simulateur est donné sur la figure 4.18.

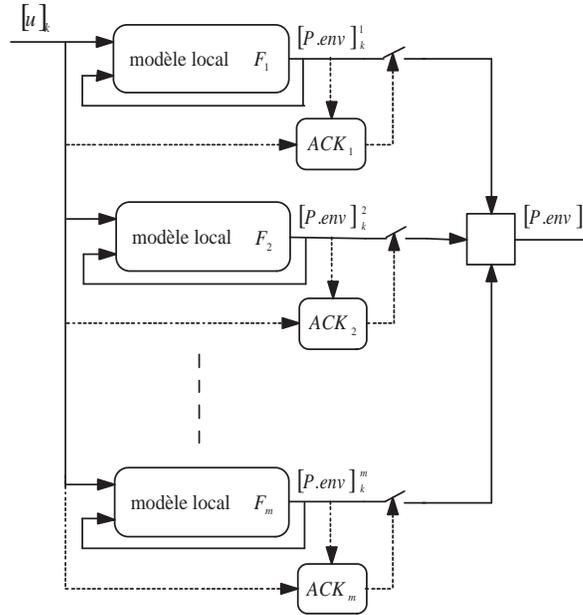


FIG. 4.18 – Simulateur du modèle multi-mode.

A l'instant k , tous les modèles locaux calculent $[P.env]_k^i$. Les blocs ACK_i déterminent les modes actifs et la sortie du simulateur est l'union des sorties des modes actifs ; c'est à dire :

$$[P.env]_k = \bigcup_{i \text{ actif}} [P.env]_k^i$$

L'enveloppe englobante de la sortie est :

$$\{[P.env]\} = \{[P.env]_1, [P.env]_2, \dots\}$$

4.7 Exemple : Système multi-mode

Dans cette section, les méthodes d'identification et de simulation d'un système multi-mode sont appliquées à un exemple.

Systeme

Le système à étudier est identique au système de l'exemple 4.4.1 sauf qu'ici, la commutation est déterminée par une condition sur la sortie du système; c'est-à-dire, si la sortie du système est inférieure à 3,7, le système fonctionne dans le premier mode opératoire et si la sortie est supérieure à 3,7, il fonctionne dans le deuxième mode. Le schéma du système est donné sur la figure 4.19. Le comportement du système dans le premier et le deuxième modes opératoires est décrit par les équations 4.4.1.

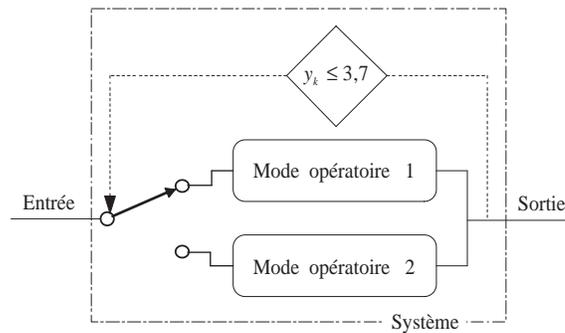


FIG. 4.19 – Système.

Jeu de données

Comme pour l'exemple 4.4.1, on conçoit une expérience sur le système et on obtient le jeu de données d'identification DS . Celui-ci doit être riche au niveau de l'information sur le comportement du système dans les modes opératoire 1 et 2. Pour simuler l'erreur de mesure et comme on l'a déjà expliqué, à l'instant k , la sortie du système est remplacée par l'intervalle

$$[y]_k = y_k + \epsilon_k + [-0,1 \ 0,1]$$

où ϵ_k , comme avant, est une variable aléatoire et on sait a priori qu'elle est bornée entre -0.1 et 0.1. L'entrée reste certaine. Donc, le jeu de données d'identification est de la forme $DS = \{(u_k, [y]_k)\}$.

Structure de modèle

Dans cet exemple, on utilise un modèle $FFIR$ intervalle avec $N = 10$ pour représenter le comportement du système. La structure F est utilisée pour surmonter le manque d'information dans les modèles dynamiques pour prédire l'enveloppe englobante au début de la prédiction et les modèles auxiliaires sont des modèles intervalles pour prendre en compte l'incertitude paramétrique du modèle.

Partitionner le régime opératoire du système

Dans la procédure d'identification multi-mode, après avoir obtenu le jeu de données d'identification et déterminé la structure du modèle, le régime opératoire du système est découpé en zones locales. Avant de commencer la procédure de découpage, on effectue une analyse préliminaire sur le jeu de données DS . Le résultat de cette analyse est montré sur la figure 4.20.

Pour découper le jeu de données, on utilise la première approche basée sur la précision de l'enveloppe englobante et comme l'exemple 4.5.1, la précision désirée est fixée à $pre = 0,8$. En appliquant cette méthode sur le jeu de données DS , on obtient deux sous-ensembles DS_1 qui contient les observations de $k = 1$ à $k = 131$ et DS_2 qui contient les observations de $k = 132$ à $k = 253$. Le résultat du découpage est montré sur la figure 4.21. Il faut se rappeler que les paramètres des modèles locaux $FFIR_1$ et $FFIR_2$ sont identifiés pendant le partitionnement.

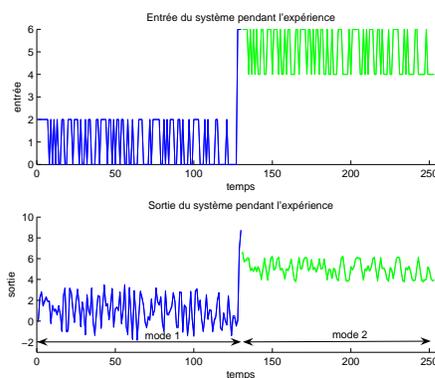
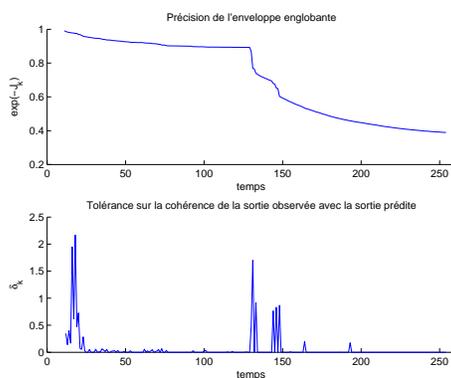


FIG. 4.20 – Analyse préliminaire sur les observations

FIG. 4.21 – Entrée et sortie du système.

Détermination des domaines de validité des modèles locaux

Dans l'étape précédente, le jeu de donnée DS a été divisé en deux sous-ensembles DS_1 et DS_2 . On sait que le système est un système du deuxième ordre. Donc, les domaines de validité des modèles locaux peuvent être déterminés dans l'espace $\chi\{y_{k-1}, y_{k-2}, u_{k-1}, u_{k-2}\}$. Cependant, cet espace peut être réduit à l'espace $\chi\{y_k, y_{k-1}, u_{k-1}\}$ puisque :

- la commutation est une condition sur une variable d'état du système, et
- les domaines de validité des modes opératoires sont des polytopes convexes dans l'espace d'état.

Le système passe du mode 1 (respect. mode 2) au mode 2 (respect. mode 1) quand la sortie est plus grande que 3,7 (respect. plus petite que 3,7). En considérant l'équation 4.4.1, la sortie du système est égale à

$$y_k = [0 \ 1] \times \begin{bmatrix} x_{(1,k)} \\ x_{(2,k)} \end{bmatrix}$$

Par conséquent, la condition de commutation peut être déterminée comme suit :

$$\begin{aligned} x_2 &\leq 3,7 & \text{mode 1} \\ x_2 &> 3,7 & \text{mode 2} \end{aligned}$$

Les domaines de validité des modes opératoires du système dans l'espace d'état sont montrés sur la figure 4.22.

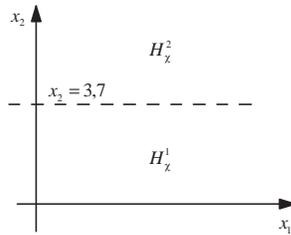


FIG. 4.22 – Système.

On calcule les enveloppes convexes pour les triplets (y_k, y_{k-1}, u_{k-1}) de DS_1 et de DS_2 dans l'espace $\chi\{y_k, y_{k-1}, u_{k-1}\}$ et on obtient H_χ^1 et H_χ^2 . Ces domaines de validité et leurs images dans l'espace $\{y_k, y_{k-1}\}$ sont montrés sur la figure 4.23.

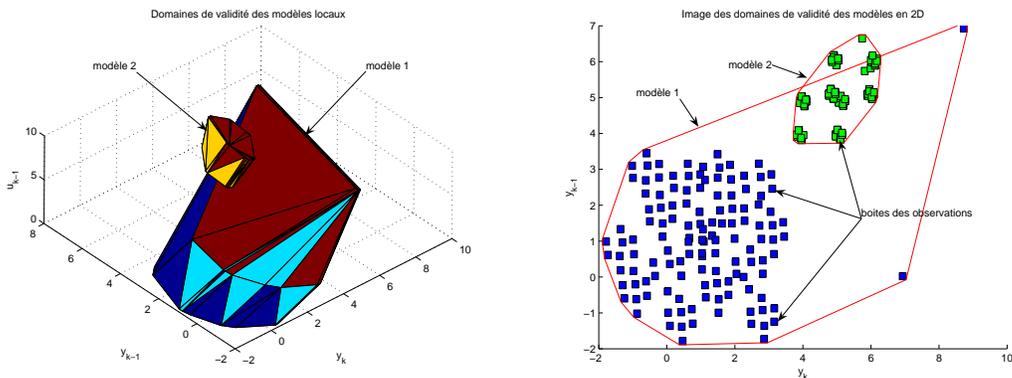


FIG. 4.23 – Domaines de validité avant d'éliminer les N points.

A la fin de la section 4.5.3, on a vu que pendant la procédure de découpage et à cause d'une valeur inadaptée de la précision désirée ou de la tolérance δ , un certain nombre des points d'un mode peuvent être classés dans le jeu de données d'un autre mode opératoire. C'est aussi le cas dans cet exemple. En regardant plus précisément la figure 4.21, on constate que la commutation n'est pas détectée au bon endroit et les points aux instants $k = 128, 129, 130$ du premier mode sont classés dans DS_2 . Pour éliminer ces points, on extrait les 10 points avant et après la commutation détectée des sous-ensembles DS_1 et DS_2 . Les domaines de validité H_χ^1 et H_χ^2 sont ensuite recalculés. Ces domaines et leurs images dans l'espace $\{y_k, y_{k-1}\}$ sont montrées sur la figure 4.24. Sur les images des domaines H_χ^1 et H_χ^2 dans l'espace $\{y_k, y_{k-1}\}$, certains points qui sont éliminés, sont montrés. Pour éviter la surestimation des supports de paramètres des modèles locaux, les paramètres des modèles $FFIR_1$ et $FFIR_2$ sont re-identifiés en utilisant les sous-ensembles DS_1 et DS_2 réduits. Pour simplifier le test de validité d'un modèle local, les domaines H_χ^1 et H_χ^2 sont ensuite remplacés par les boîtes $\square H_\chi^1$ et $\square H_\chi^2$. Ces boîtes sont montrées sur la figure 4.25. Dans cet exemple, l'agrégation des modes ne peut pas avoir lieu puisque $\square H_\chi^1$ et $\square H_\chi^2$ sont deux domaines disjoints.

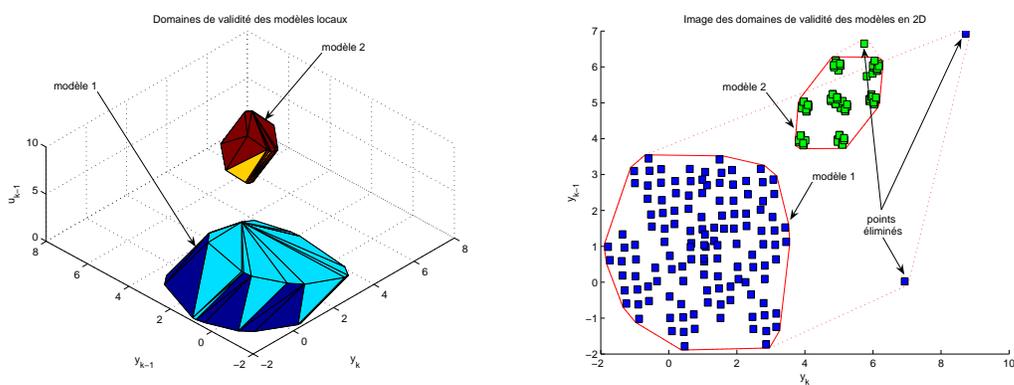


FIG. 4.24 – Domaines de validité après avoir éliminé les N points.

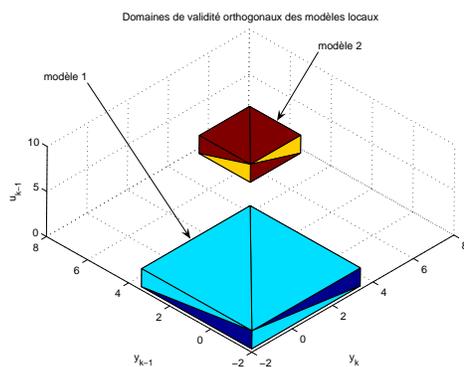


FIG. 4.25 – Approximations orthogonales des domaines de validité.

Remarque. On constate deux points sur les images en deux dimensions des domaines de validité des figures 4.23 et 4.24 qui sont ni proches de l'ensemble des points du mode 1 ni de l'ensemble des points du mode 2. Ceci est le résultat d'une erreur lors de la simulation des données du système 4.4.1. Au dernier instant où le système se trouve dans le mode 1, il reçoit une entrée le conduisant à se retrouver dans le mode 2 à l'instant suivant. Puisqu'on effectue la simulation en temps discret, on utilise la seule dynamique du mode 1 pour calculer la sortie à l'instant suivant alors que le système passe effectivement du mode 1 au mode 2 entre ces deux instants. Par conséquent, le premier point qui se trouve dans le mode 2 est un point qui est calculé par le mode 1 : il n'est ni proche de l'ensemble des points du mode 1 (puisque'il n'est pas un point de mode 1) et ni de l'ensemble des points du mode 2 (puisque'il a été calculé par rapport à la dynamique du mode 1). La distance du deuxième point du mode 2 des domaines de validité des modèles 1 et 2 est aussi la conséquence de cette erreur. \diamond

Prédiction de l'enveloppe englobante

Après avoir identifié les modèles locaux et déterminé leurs domaines de validité, on utilise ce modèle multi-mode pour prédire l'enveloppe englobante de la sortie du système pour une séquence des entrées donnée. Pour cela, on utilise le simulateur de la section 4.6.2 qui est montré sur la figure 4.18 et l'enveloppe englobante est calculée pour une séquence des entrées $U = \{[u]_1, \dots, [u]_{ph}\}$ ⁶ par la méthode expliquée dans la section 4.6.2. La séquence U est ensuite appliquée au système 4.4.1. Sur la figure 4.26 l'enveloppe prédite est comparée avec la sortie du système. Les mesures de la qualité de prédiction sont comme suit :

Erreur de la prédiction	$E = 8\%$
Largeur moyenne de l'enveloppe englobante	$J = 0,4258$
Précision de l'enveloppe englobante	$P = 0,6533$

Si on modifie la séquence des entrées telle que le système passe une seule fois du mode 1 à mode 2, on obtient les résultats suivants :

Erreur de la prédiction	$E = 8\%$
Largeur moyenne de l'enveloppe englobante	$J = 0,1554$
Précision de l'enveloppe englobante	$P = 0,8561$

ce qui montre l'effet des commutations successives sur la précision de l'enveloppe prédite par le modèle multi-mode.

Dans l'exemple 3.4.3, on a vu qu'un jeu de données insuffisamment riche peut faire que, dans certains cas, l'intersection des sorties des modèles auxiliaires soit vide et par conséquent, l'enveloppe prédite sera vide. Dans un modèle multi-mode, un autre phénomène qui peut aussi arriver est qu'à certains instants, aucun modèle n'est valide. Autrement dit, tous les blocs ACK_i sont dans l'état *transitoire*. Dans ce cas, l'enveloppe ne

⁶Cette séquence est différente de la séquence des entrées du jeu de données d'identification.

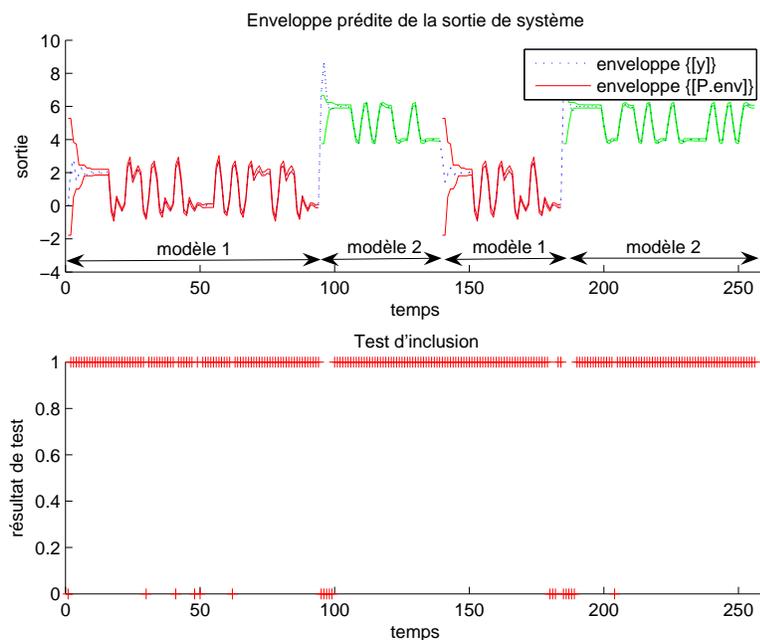


FIG. 4.26 – Prédiction de l’enveloppe de la sortie du système.

peut pas être calculée et on dit qu’elle est indéterminable. Pour illustrer ce phénomène, le système est re-identifié pour un jeu de donnée moins riche puis on prédit l’enveloppe englobante pour la séquence des entrées U . Le résultat de la prédiction est montrée sur la figure 4.27. Cela montre encore une autre fois l’importance de la qualité du jeu de données d’identification sur la qualité de la prédiction de l’enveloppe englobante du modèle multi-mode.

4.8 Conclusion

L’objectif de ce chapitre était de trouver et de proposer une solution pour identifier et simuler des systèmes multi-mode ou non-linéaires par des modèles linéaires multi-modes. Pour cela, on a étudié deux cas : celui où la commutation est déterminée par un signal externe et celui où la commutation est indéterminée. Dans la section trois, le problème d’identification d’un modèle multi-mode a été traité dans le cas où la commutation est déterminée. On a vu que ce problème s’est traduit par plusieurs problèmes d’identification de modèles mono-modes et dans la section quatre, ce modèle a été utilisé pour prédire l’enveloppe englobante de la sortie du système.

Si la commutation est indéterminée, par rapport au cas où la commutation est déterminée, trois problèmes supplémentaires doivent être résolus : partitionner le régime opératoire du système en zones locales et représenter chaque zone par un modèle local, déterminer le domaine de validité des modèles locaux et agréger les modèles similaires.

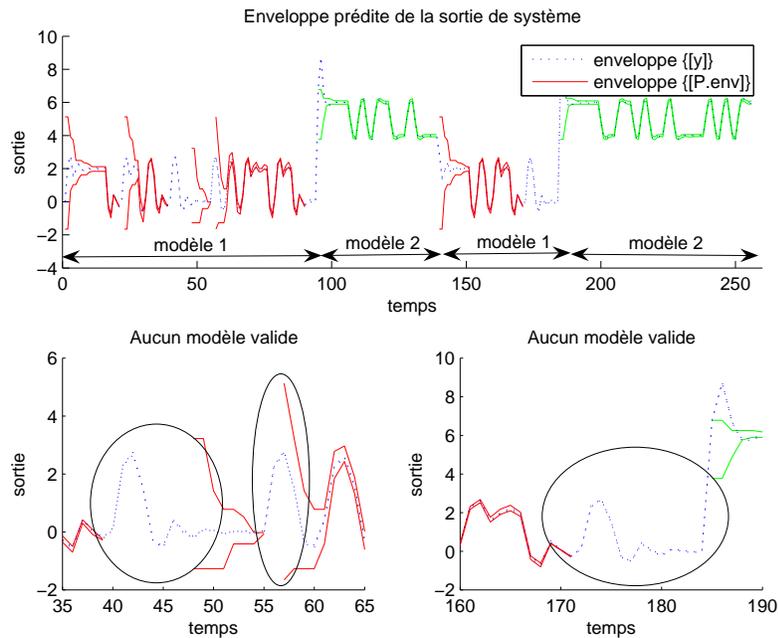


FIG. 4.27 – Effet d'un mauvais jeu de données de l'identification 2 sur la prédiction.

Ces problèmes ont été traités dans la section cinq. Dans la section six, un simulateur a été proposé pour le modèle multi-mode identifié dans la section précédente. Dans la section sept, les méthodes d'identification et de simulation ont été appliquées sur un exemple d'un système multi-mode.

Dans le chapitre suivant, la conclusion de ce travail de thèse est donnée.

Chapitre 5

Conclusion et Perspectives

L'imagination est plus importante que le savoir.

Albert Einstein

Conclusion

Dans ce document, on a proposé une méthode pour identifier un modèle multi-mode ensembliste dans lequel les paramètres sont des variables incertaines caractérisées par des intervalles. Ce modèle a été ensuite utilisé pour calculer l'enveloppe englobante de la sortie du système dans le cadre d'une analyse de sûreté pour vérifier si l'ensemble des trajectoires de la sortie du système reste dans une zone donnée. On a apporté dans ce travail des réponses aux quatre questions posées dans le chapitre d'introduction de ce document.

La première question concerne la structure du modèle. Les structures comme le modèle d'état ou le modèle relation entrée-sortie sont des modèles dits *récurifs* puisque l'état suivant et la sortie du système sont calculés en utilisant son état et sa sortie aux instants précédents. Dans une approche ensembliste, un calcul itératif est souvent évité puisqu'à cause de l'effet d'enveloppement et l'accumulation de l'imprécision après chaque itération, on peut être confronté à une prédiction imprécise (une enveloppe large) ou même à l'explosion du domaine de calcul. Pour résoudre ce problème, le modèle réponse impulsionnelle à horizon fini a été proposé dans ce travail. Dans ce modèle, la sortie du système est calculée directement à partir des entrées précédentes du système et par conséquent, l'accumulation de l'imprécision ne se produit pas. Un autre avantage de ce modèle par rapport aux modèles récurifs est que dans les modèles récurifs, connaître l'état initial du système est essentiel pour faire le calcul des trajectoires. Cependant, dans un modèle réponse impulsionnelle, le calcul se fait indépendamment de l'état initial du système et la sortie est calculée qu'à partir des entrées décalées dans le temps. C'est un avantage surtout dans un modèle multi-mode, où après chaque commutation,

l'état initial est inconnu par rapport au nouveau mode opératoire. Un modèle relation entrée-sortie a été aussi étudié dans ce document.

La deuxième question concerne les valeurs des paramètres du modèle. Généralement, par *identification de paramètres* d'un modèle déterministe, on veut dire une estimation optimale des valeurs des paramètres du modèle. Dans ce cas, les paramètres sont des scalaires et ils sont supposés très souvent invariants dans le temps. Cependant, dans un modèle ensembliste et dans notre cas, un modèle intervalle, on cherche les supports des paramètres du modèle. Les valeurs réelles des paramètres bougent dans leurs supports, mais elles n'en sortent jamais. L'objectif est de prendre en compte l'incertitude paramétrique du modèle et de donner au modèle la capacité d'expliquer tous les comportements possibles du système. De ce fait et à un instant donné, la valeur exacte d'un paramètre est un élément de l'ensemble des valeurs possibles de celui-ci. Cet ensemble est caractérisé par un support. La valeur de la variable est en général variant dans le temps, bornée et elle reste toujours dans son support. Les supports de paramètres dans un modèle intervalle sont des intervalles. L'objectif de trouver les supports des paramètres du modèle a été formalisé comme la sémantique d'identification des paramètres du modèle intervalle. Cette sémantique est un formalisme mathématique composé des paramètres du modèle à identifier, des variables du système (l'entrée et la sortie du système) ainsi que des quantificateurs \forall , \exists et \neg .

La troisième question concerne le domaine optimal des supports de chacun des paramètres du modèle. L'ensemble des valeurs possibles pour chacun des paramètres peut être englobé par différents intervalles; des intervalles avec les bornes proches ou des intervalles avec des bornes plus éloignées. Toutefois, les bornes supérieures et inférieures des supports des paramètres doivent être optimisées telles que l'enveloppe prédite par le modèle soit l'enveloppe la plus petite possible. Cette condition a conduit à la définition du critère d'optimisation des bornes supérieures et inférieures de supports des paramètres du modèle. Le problème d'identification d'un modèle intervalle a donc été traduit comme un problème d'optimisation sous un ensemble de contraintes. Ce problème a été ensuite résolu à l'aide d'un algorithme d'optimisation.

La dernière question posée dans l'objectif de ce travail était le problème de l'identification et de la simulation d'un modèle multi mode. Un modèle linéaire intervalle peut être toujours utilisé pour représenter le comportement d'un système non-linéaire. La conséquence est d'avoir une prédiction moins précise par rapport à un modèle non-linéaire avec une structure plus complexe. Pour bénéficier des avantages d'une structure simple pour le modèle ainsi qu'améliorer la précision de prédiction du modèle, la partie non-linéaire du régime opératoire du système a été partitionnée en plusieurs morceaux tels que chaque morceau puisse être expliqué par un modèle linéaire local en satisfaisant la précision de prédiction désirée. C'est aussi le cas pour un système multi-mode. Pendant ce travail, deux méthodes de partitionnement ont été proposées; une basée sur la précision de l'enveloppe englobante et l'autre basée sur la cohérence de la sortie observée avec la sortie prédite par le modèle. Une technique a été ensuite proposée pour déterminer les domaines de validité des modèles locaux dans l'espace de validé de modèles. Puis pour arriver à la dernière étape qui consistait en la prédiction de l'enveloppe de la sortie du système par le modèle multi-mode, un simulateur pour le modèle multi-mode intervalle a été développé et à l'aide de ce simulateur, on est arrivé à prédire l'ensemble

des trajectoires du système sur un horizon de prédiction.

Pour arriver à répondre à ces quatre questions posées dans l'objectif de ce travail, on a dû aussi résoudre plusieurs problèmes supplémentaires qui nous empêchaient d'atteindre notre objectif. Un bref bilan sur les travaux effectués dans les chapitres de ce document est présenté par la suite.

Chapitre un. Dans le chapitre un, l'objectif de ce travail a été présenté. Le but est de trouver un modèle mathématique du système qui peut prédire tous les comportements du système en prenant en compte l'incertitude paramétrique et non-paramétrique du modèle. Ce modèle sera ensuite utilisé pour effectuer une analyse de sûreté.

Chapitre deux. Dans ce chapitre, l'état de l'art de l'identification et de la simulation d'un modèle et l'identification et la simulation d'un modèle ensembliste et multi-mode a été étudié brièvement.

Chapitre trois. Dans le chapitre trois, on a traité le problème de l'identification et de la simulation d'un modèle intervalle. Dans un premier temps, l'outil intervalle a été présenté. Cet outil nous permet de gérer l'incertitude paramétrique du modèle. Identifier un modèle intervalle consiste à choisir la structure du modèle, déterminer la sémantique d'identification, définir le critère d'optimisation et calculer les supports des paramètres du modèle. Ces derniers ont été traités dans la section trois. Dans cette section, le problème d'invalidité d'un modèle dynamique au début de la procédure d'identification (respect. au début de la simulation) a été étudiée et la structure F a été proposée pour résoudre cet inconvénient. Dans la section quatre, le modèle a été utilisé pour calculer une enveloppe englobante qui inclut toutes les valeurs possibles de la sortie du système. Pour cela, la sémantique de la simulation a été définie et deux méthodes numériques (une méthode pour le modèle $FFIR$ et une méthode pour le modèle FIO) ont été proposées pour calculer cette enveloppe englobante. Une conclusion a été donnée dans la section cinq. Les contributions principales de ce chapitre peuvent être résumées comme suit :

- Généraliser les méthodes d'identification des supports de paramètres d'un modèle intervalle dans le cas où les entrées du système dans le jeu de données d'identification sont imprécises.
- Proposer la structure F qui est un banc de modèles constitué d'un modèle statique et d'un ensemble de modèles dynamiques dont les horizons glissants varient de $HG = 1$ à $HG = N$. Cette structure est utilisée pour résoudre le problème du manque d'informations pour un modèle dynamique au début des procédures d'identification et de simulation. Ce problème n'est pas gênant pour un modèle mono-mode. Cependant, il devient critique pour simuler un modèle multi-mode car, dans le cas d'un modèle réponse impulsionnelle, si le système commute fréquemment d'un mode à un autre avant le passage de l'horizon glissant du modèle, on n'arrive pas à calculer l'enveloppe de la sortie. Sans l'utiliser la structure F , le

calcul de l'enveloppe englobante est impossible pour un modèle relation entrée-sortie puisqu'après la commutation, on n'aura pas accès à l'état actuel du système dans le nouveau mode et la prédiction ne peut pas être réalisée.

- Proposer d'utiliser un modèle réponse impulsionnelle à horizon fini au lieu d'un modèle réponse entrée-sortie ou d'un modèle d'état pour éviter l'effet d'enveloppement pendant les calculs de l'enveloppe englobante de la sortie du système.

Chapitre quatre. Dans le chapitre quatre, le problème de l'identification et de la simulation d'un modèle multi-mode a été étudié. Ce modèle multi-mode a été utilisé pour améliorer la précision de la modélisation et de la prédiction de l'enveloppe englobante de la sortie du système. Après l'introduction et l'objectif de ce chapitre qui ont été présentés dans les sections un et deux, le problème de l'identification et de la simulation multi-mode a été considéré dans deux cas : le cas où la commutation est déterminée et le cas où la commutation est indéterminée. L'identification multi-mode dans le premier cas s'est traduite par plusieurs problèmes d'identification de modèles mono-modes. Celui-ci a été traité dans la section trois et dans la section quatre, ce modèle multi-mode a été utilisé pour calculer l'enveloppe englobante de la sortie du système. L'identification multi-mode dans le cas où la commutation est indéterminée consiste en trois problèmes supplémentaires : partitionner le régime opératoire du système, déterminer les domaines de validité de modèles locaux et agréger les modèles similaires qui ont été étudiés dans la section cinq. Pour prédire l'enveloppe englobante de la sortie du système par ce modèle multi-mode, un simulateur multi-mode a été proposé dans la section six. Ce simulateur contient deux types d'éléments : les modèles locaux qui ont été identifiés précédemment et les blocs ACK_i . Ces blocs nous permettent de déterminer les modes opératoires actifs. Ce chapitre se termine par un exemple pour illustrer les méthodes de l'identification et de la simulation multi-mode. On peut rappeler brièvement les contributions principales de ce chapitre comme suit :

- Proposer deux méthodes pour partitionner le régime opératoire du système en zones locales de fonctionnement.
- Déterminer les domaines de validité des modèles locaux dans l'espace des variables caractéristiques et étudier la possibilité de diminuer le nombre de dimensions de cet espace.
- Développer le simulateur du modèle multi-mode intervalle. Ce simulateur contient les modèles locaux ainsi que les blocs ACK_i qui déterminent la validité des modèles locaux pendant la prédiction.

Perspective

Ce travail de thèse peut être poursuivi selon les axes suivants :

Optimiser les paramètres du modèle dans le cas d'un critère non-linéaire.

Dans la section 3.3.5 et sur le tableau 3.1, la nature du critère d'optimisation en fonction du modèle et des grandeurs incertaines est montrée. Dans le cas où le critère est linéaire, la méthode proposée pour optimiser les paramètres trouve une solution qui est globalement optimale. Cependant, dans le cas où le critère est non-linéaire, il y a un risque de trouver un optimum seulement local. Ce problème peut être résolu en utilisant les méthodes d'optimisation globale dont on peut citer les méthodes d'optimisation intervalle [Jaulin et al., 2001*b*]. Cependant, ce type de méthodes sont souvent très lent lorsque le nombre de paramètres à optimiser est important et il faut étudier la faisabilité de les utiliser dans notre approche.

Identification d'un modèle intervalle pour d'autres applications. Dans ce travail et dans le cadre d'une analyse de sûreté, les paramètres du modèle intervalle ont été identifiés tels que le modèle puisse expliquer toutes les valeurs possibles de la sortie du système et l'englober par une enveloppe. La sémantique de l'identification peut être modifiée en fonction de l'objectif de la modélisation et pour que le modèle puisse être utilisé dans d'autres applications comme le diagnostic, la détection défauts ou la commande [Calm et al., 2006].

Optimiser le critère de partitionnement du régime opératoire. Dans la section 4.5.2, on a vu que pour partitionner le régime opératoire du système, la condition pour déterminer le passage d'un mode à un autre (la précision de la modélisation ou la tolérance sur la cohérence de la sortie observée avec la sortie prédite) est pré-déterminée. Le problème consistant à optimiser cette valeur n'est pas encore traité.

Trouver la dimension optimale pour la matrice de découplage. Dans la section 3.3.2, la structure du modèle est définie selon l'hypothèse que les paramètres sont indépendants. En ajoutant la matrice de couplage, la dépendance des paramètres du modèle peut être prise en compte et par conséquent, un modèle plus précis peut être obtenu. Une partie de ce problème a été déjà traité dans [Adrot et al., 2004] pour un modèle entrée-sortie dans le cas où la dimension de la matrice de couplage est déterminée. Pourtant, le problème d'optimisation de la dimension de cette matrice n'est pas encore étudié.

Simuler le modèle intervalle. Dans ce document, le modèle intervalle a été utilisé dans le cadre d'une analyse de sûreté pour trouver l'enveloppe englobante de la sortie du système. Par conséquent, la structure et la sémantique du simulateur utilisées ont été adaptées à cet objectif. Pour que ce simulateur puisse être utilisé dans d'autres applications comme le diagnostic, la détection défauts ou la commande, on a besoin de re-définir la sémantique de la simulation en fonction de l'objectif visé et de développer d'autres méthodes pour calculer cette enveloppe. Pour la sémantique de ce travail, l'enveloppe a été calculée directement par l'outil intervalle. Cependant, pour des autres sémantiques, on aura potentiellement besoin d'autres outils comme les intervalles modaux [Armengol, 1999].

Recouvrement des domaines de validité et agrégation de ces domaines.

Dans la section 4.5.4, on a déterminé trois règles pour agréger les modèles locaux

qui représentent un comportement similaire du système. Ces règles sont basées sur les domaines de validité des modèles locaux et on agrège le modèle i dans le modèle j seulement si $\square H^i \subset \square H^j$. Cependant, un recouvrement important de ces domaines peut aussi être interprété comme des modèles i et j qui représentent un comportement similaire du système. Modifier ces règles pour une agrégation plus efficace des modèles reste toujours un problème ouvert à étudier.

Annexe A

Modèles Mathématiques

Les fonctions de transfert G et H peuvent être déterminées de différentes façons comme des fonctions rationnelles en l'opérateur retard q . Par la suite, on présentera un certain nombre de telles configurations.

Modèle ARX. ¹ Probablement, la relation la plus simple entre l'entrée et la sortie observées peut être décrite par une équation aux différences comme suit :

$$y_k + a_1 y_{k-1} + \dots + a_{n_a} y_{k-n_a} = b_1 u_{k-1} + \dots + b_{n_b} u_{k-n_b} + e_k \quad (\text{A.0.1})$$

Puisque l'erreur e_k se présente sous la forme d'un unique terme additif, ce modèle s'appelle très souvent un *modèle erreur à d'équation*. Les paramètres du modèle sont :

$$\theta = [a_1 \ a_2 \ \dots \ a_{n_a} \ b_1 \ \dots \ b_{n_b}]$$

Si on introduit

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

et

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

on voit que :

$$G(q) = \frac{B(q)}{A(q)}, \quad H(q) = \frac{1}{A(q)}$$

Dans le cas où $n_a = 0$, ce modèle devient un modèle réponse impulsionnelle à horizon fini².

¹AutoRegression eXtra input

²Finite Impulse Response (FIR).

Si à l'instant k , l'erreur e_k est indéterminée, on utilisera la forme déterministe du modèle. Dans ce cas, on peut montrer que si y_s et u_s pour $s < k$ sont connus, la prédiction à un pas de la sortie de système par le modèle de l'équation 3.3.9 est calculée par l'expression suivante [Ljung, 1999] :

$$\widehat{y}(k, \theta) = H^{-1}(q)G(q)u_k + [1 - H^{-1}(q)]y_k \quad (\text{A.0.2})$$

On utilise θ dans l'expression $\widehat{y}(k, \theta)$ puisque la valeur prédite de la sortie dépend au vecteur de paramètres de modèle θ . Pour un modèle *ARX*, l'équation A.0.2 devient :

$$\widehat{y}(k, \theta) = B(q)u_k + [1 - A(q)]y_k \quad (\text{A.0.3})$$

On introduit la notation

$$\varphi_k = [-y_{k-1} \dots -y_{k-n_a} \quad u_{k-1} \dots u_{k-n_b}]^T$$

En utilisant le vecteur φ_k , l'équation A.0.3 peut être re-écrite :

$$\widehat{y}(k, \theta) = \theta^T \varphi_k \quad (\text{A.0.4})$$

Donc, la prédiction peut se faire par un produit scalaire entre le vecteur des données φ_k qui est connu et le vecteur des paramètres θ . En statistique, un tel modèle s'appelle *régression linéaire* et le vecteur φ_k s'appelle le *vecteur de régression*. La simplicité de ce modèle pour prédire la sortie de système en fait le premier choix pour des nombreuses applications. En considérant l'équation A.0.4, la sortie observée est égale à :

$$y_k = \widehat{y}(k, \theta) + e_k$$

Modèle ARMAX. ³ L'inconvénient du modèle A.0.1 est le manque de liberté pour décrire les propriétés de l'erreur e . On ajoute cette flexibilité en décrivant l'erreur comme une moyenne sur un horizon glissant⁴ :

$$y_k + a_1 y_{k-1} + \dots + a_{n_a} y_{k-n_a} = b_1 u_{k-1} + \dots + b_{n_b} u_{k-n_b} + e_k + c_1 e_{k-1} + \dots + c_{n_c} e_{k-n_c} \quad (\text{A.0.5})$$

Par

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

On aura donc :

$$A(q)y_k = B(q)u_k + C(q)e_k \quad (\text{A.0.6})$$

³AutoRegressive Moving Average eXtra input

⁴Moving Average

ou

$$G(q) = \frac{B(q)}{A(q)}, \quad H(q) = \frac{C(q)}{A(q)}$$

et les paramètres du modèle sont :

$$\theta = [a_1 \ a_2 \ \dots \ a_{n_a} \ b_1 \ \dots \ b_{n_b} \ c_1 \ \dots \ c_{n_c}]$$

Cette structure est devenue standard en automatique.

Considérant l'équation A.0.2, la prédiction à un pas de la sortie du système par le modèle *ARMAX* est :

$$\hat{y}(k, \theta) = \frac{B(q)}{C(q)}u_k + \left[1 - \frac{A(q)}{C(q)}\right]y_k$$

ou

$$C(q)\hat{y}(k, \theta) = B(q)u_k + [C(q) - A(q)]y_k$$

et on obtient

$$\hat{y}(k, \theta) = B(q)u_k + [1 - A(q)]y_k + [C(q) - 1][y_k - \hat{y}(k, \theta)]$$

On introduit l'erreur

$$e_{k,\theta} = y_k - \hat{y}(k, \theta)$$

et le vecteur

$$\varphi_{k,\theta} = [-y_{k-1} \ \dots \ -y_{k-n_a} \ u_{k-1} \ \dots \ u_{k-n_b} \ e_{k-1,\theta} \ \dots \ e_{k-n_c,\theta}]^T$$

Donc, la sortie de système à l'instant k peut être prédite par :

$$\hat{y}(k, \theta) = \theta^T \varphi_{k,\theta}$$

Puisque le vecteur de régression dépend du vecteur de paramètres θ , la prédiction dans un modèle *ARMAX* se fait par une régression non-linéaire. La sortie observée est égale à :

$$y_k = \hat{y}(k, \theta) + e_k$$

Remarque. Par la suite et pour simplifier les notations, on éliminera le terme θ dans l'expression $\hat{y}(k, \theta)$ et on le notera \hat{y}_k . Pourtant, il faut se rappeler que \hat{y}_k dépend toujours du vecteur de paramètres θ . \diamond

Modèles ARARX et ARARMAX. ^{5 6} Au lieu de décrire l'erreur dans l'équation A.0.5 comme une moyenne sur un horizon glissant, elle peut être décrite par une autorégression. Cela nous donne la structure suivante :

$$A(q)y_k = B(q)u_k + \frac{1}{D(q)}e_k \quad (\text{A.0.7})$$

avec

$$D(q) = 1 + d_1q^{-1} + \dots + d_{n_d}q^{-n_d}$$

Ce modèle s'appelle *ARARX*. Plus généralement, on peut utiliser une description *ARMA* pour l'erreur et on obtient le modèle *ARARMAX* :

$$A(q)y_k = B(q)u_k + \frac{C(q)}{D(q)}e_k \quad (\text{A.0.8})$$

qui regroupe les modèles A.0.1, A.0.6 et A.0.7 comme cas particuliers. La description ci-dessus forme une famille des modèles à erreur d'équation qui est montrée sur la figure A.1.

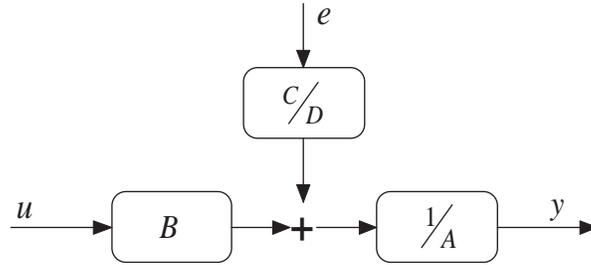


FIG. A.1 – Famille des modèles à erreur d'équation.

Modèle erreur de sortie. Dans les modèles à erreur d'équation présentés précédemment, les fonctions de transfert G et H ont un facteur en commun au dénominateur. D'un point de vue physique, il apparaît plus naturel que ces deux fonctions soient indépendantes. Si on suppose que la relation entre l'entrée et la sortie estimée \hat{y} peut être décrite par une équation linéaire aux différences, on obtient la description suivante :

$$\hat{y}_k + f_1\hat{y}_{k-1} + \dots + f_{n_f}\hat{y}_{k-n_f} = b_1u_{k-1} + \dots + b_{n_b}u_{k-n_b}$$

Avec

$$F(q) = 1 + f_1q^{-1} + \dots + f_{n_f}q^{-n_f}$$

⁵AutoRegressive AutoRegressive eXtra input

⁶AutoRegressive AutoRegressive Moving Average eXtra input

on peut écrire le modèle comme suit :

$$y_k = \frac{B(q)}{F(q)} + e_k \quad (\text{A.0.9})$$

Le vecteur de paramètres du modèle est :

$$\theta = [b_1 \ b_2 \ \dots \ b_{n_b} \ f_1 \ f_2 \ \dots \ f_{n_f}]$$

Modèle Box-Jenkins. Un développement naturel du modèle A.0.9 se fait en décrivant les propriétés de l'erreur comme un modèle *ARMA*. Donc, on obtient :

$$y_k = \frac{B(q)}{F(q)}u_k + \frac{C(q)}{D(q)}e_k \quad (\text{A.0.10})$$

On peut considérer l'équation A.0.10 comme l'extension la plus naturelle de l'équation 3.3.9 puisque les fonctions G et H sont paramétrisées comme des fonctions indépendantes et rationnelles en q . Le modèle de l'équation A.0.10 représente une famille de modèles à erreur de sortie.

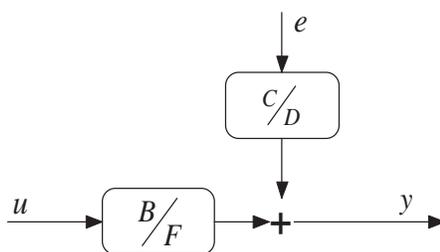


FIG. A.2 – Modèle Box-Jenkins.

Fonctions Polynomiales Utilisées	Nom de Modèle
B	FIR (Réponse Impulsionnelle Finie)
AB	ARX
ABC	ARMAX
AC	ARMA
ABD	ARARX
ABCD	ARARMAX
BF	OE (Erreur de Sortie)
BFCD	BJ (Box-Jenkins)

TAB. A.1 – Modèles boîte noire classiques.

Les structures qu'on a présenté dans cette section peut produire 32 modèles différents en fonction de quelles fonctions polynomiales A , B , C , D et F sont utilisées. Le tableau A résume un certain nombre de cas particuliers des modèles A.0.8 et A.0.10.

Annexe B

Représentation géométrique de la sémantique d'identification

On considère, sans perte de généralité, que le modèle est un modèle réponse impulsionnelle d'ordre deux. Dans un premier temps et pour faciliter la représentation graphique, on suppose que l'erreur additive est nulle. Donc, l'équation entre l'entrée et la sortie est comme suit :

$$y_k = b_{1,k}u_{k-1} + b_{2,k}u_{k-2} \quad (\text{B.0.1})$$

Les paramètres $b_{1,k}$ et $b_{2,k}$ sont supposés indépendants et par conséquent, le domaine déterminé par $[\theta] = [[b_1] [b_2]]^T$ dans l'espace de paramètres est une boîte. Cette boîte est dessinée sur la figure B.1. On a déjà vu dans l'équation 3.2.8 qu'un vecteur intervalle peut être décrit sous la forme normalisée. On a donc :

$$[\theta] = \theta_c + \lambda_\theta \times [v]$$
$$\theta_c = \begin{bmatrix} \text{centre}([b_1]) \\ \text{centre}([b_2]) \end{bmatrix}, \quad \lambda_\theta = \begin{bmatrix} \text{rayon}([b_1]) & 0 \\ 0 & \text{rayon}([b_2]) \end{bmatrix} \quad (\text{B.0.2})$$

où le vecteur θ_c détermine le centre de la boîte et la matrice diagonale λ_θ définit la largeur de cette boîte sur chaque axe.

Les valeurs y_k , u_{k-1} et u_{k-2} sont des observations sur le système et sont donc connues. Si on résout l'équation B.0.1 par rapport aux paramètres du modèle $b_{1,k}$ et $b_{2,k}$ qui doivent être identifiés, on obtient l'expression suivante :

$$l_k : b_{2,k} = -\frac{u_{k-1}}{u_{k-2}}b_{1,k} + \frac{1}{u_{k-2}}y_k \quad (\text{B.0.3})$$

Annexe B. Représentation géométrique de la sémantique d'identification

Pour $u_{k-2} = 0$, l'expression de la droite l_k est sous la forme suivante :

$$l_k : b_{1,k} = \frac{y_k}{u_{k-1}}$$

Cette expression détermine une droite l_k dans l'espace des paramètres qui est en deux dimensions¹. Toutes les valeurs sur la droite l_k sont des solutions des paramètres $b_{1,k}$ et $b_{2,k}$ pour le triplet $\{y_k, u_{k-1}, u_{k-2}\}$.

Pour différentes valeurs de k , les triplets $\{y_k, u_{k-1}, u_{k-2}\}$ conduisent à un ensemble de droites dans l'espace $\{b_1, b_2\}$ et chaque droite détermine les solutions possibles de $b_{1,k}$ et $b_{2,k}$ pour le triplet respectif. La sémantique d'identification dit que les bornes des intervalles $[b_1]$ et $[b_2]$ doivent être identifiées telles que :

$$\forall k, \exists b_{1,k} \in [b_1], \exists b_{2,k} \in [b_2] \mid y_k = b_{1,k}u_{k-1} + b_{2,k}u_{k-2}$$

Autrement dit, la boîte $[\theta] = [[b_1] [b_2]]^T$ doit contenir au moins un point de toutes les droites l_k pour différentes valeurs de k . Si les droites se croisent en un point, ce point est la solution du problème d'identification et le modèle sera un modèle parfait sans incertitude paramétrique. Sinon, aucune solution identique existe pour tous les instants k . Dans ce cas, on définit la boîte par un centre θ_c et un espace autour de ce centre où à chaque instant, au moins une valeur θ_k soit solution. Une condition complémentaire sur $[\theta]$ est que l'image de $[\theta]$ dans l'espace de sortie -enveloppe englobante- doit être l'enveloppe la plus petite possible. Cette condition est décrite sous la forme du critère d'optimisation qui est défini dans la section 3.3.4 3.3.24.

Dans le cas où l'erreur additive est présente, au lieu de l'équation B.0.1 on aura :

$$y_k = b_{1,k}u_{k-1} + b_{2,k}u_{k-2} + e_k \quad (\text{B.0.4})$$

Si on résout l'équation B.0.4 par rapport aux paramètres $b_{1,k}$ et $b_{2,k}$, on obtient :

$$l_k : b_{2,k} = -\frac{u_{k-1}}{u_{k-2}}b_{1,k} + \frac{1}{u_{k-2}}y_k - \frac{1}{u_{k-2}}e_k \quad (\text{B.0.5})$$

Pour $u_{k-2} = 0$, on a :

$$l_k : b_{1,k} = \frac{1}{u_{k-1}}y_k - \frac{1}{u_{k-1}}e_k$$

Le domaine de variation de l'erreur additive est caractérisé par l'intervalle $[e]$. On écrit cet intervalle sous la forme normalisée :

¹De manière plus général, cela sera un hyperplan dans l'espace des paramètres en plusieurs dimensions.

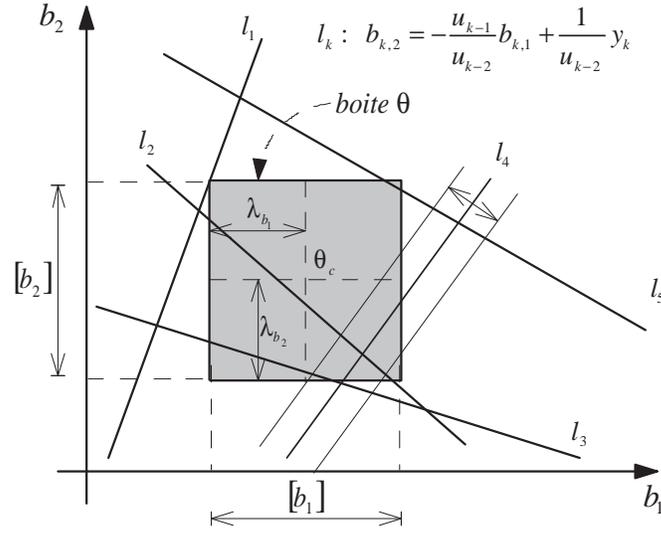


FIG. B.1 – Espace des paramètres et domaine de solution pour les paramètres du modèle.

$$[e] = e_c + \lambda_e \times [-1 \ 1]$$

Si on substitue la forme normalisée de l'erreur dans l'équation B.0.6, pour chaque k , on obtient une bande qui est encadrée par les deux bornes suivantes :

$$\begin{aligned} \bar{l}_k & : b_{2,k} = -\frac{u_{k-1}}{u_{k-2}} b_{1,k} + \frac{1}{u_{k-2}} y_k - \frac{1}{u_{k-2}} e_c + \frac{\lambda_e}{|u_{k-2}|} \\ \underline{l}_k & : b_{2,k} = -\frac{u_{k-1}}{u_{k-2}} b_{1,k} + \frac{1}{u_{k-2}} y_k - \frac{1}{u_{k-2}} e_c - \frac{\lambda_e}{|u_{k-2}|} \end{aligned}$$

Ces deux bornes sont déterminées par

$$\begin{aligned} \bar{l}_k & : b_{1,k} = \frac{1}{u_{k-1}} y_k - \frac{1}{u_{k-1}} e_c + \frac{\lambda_e}{|u_{k-1}|} \\ \underline{l}_k & : b_{1,k} = \frac{1}{u_{k-1}} y_k - \frac{1}{u_{k-1}} e_c - \frac{\lambda_e}{|u_{k-1}|} \end{aligned}$$

dans le cas où $u_{k-2} = 0$. Cette bande est représentée pour $k = 4$ (l_4) sur la figure B.1. En considérant l'équation 3.3.22, la sémantique d'identification décrit :

$$\forall k, \exists b_{1,k} \in [b_1], \exists b_{2,k} \in [b_2] \exists e_k \in [e] \mid y_k = b_{1,k} u_{k-1} + b_{2,k} u_{k-2} + e_k$$

Cette expression peut être interprétée comme suit :

Annexe B. Représentation géométrique de la sémantique d'identification

La boîte $[\theta]$ et l'intervalle $[e]$ doivent être déterminés tels que la boîte a au moins un point en commun avec toutes les bandes pour différentes valeurs de k .

Remarque. Pour un intervalle $[e]$ de taille suffisante, la boîte $[\theta]$ peut se réduire à un vecteur intervalle de taille nulle (un point dans l'espace de paramètres). Cela signifie que l'incertitude sur le terme additif e_k est suffisamment grande pour expliquer toutes les observations. Il est donc inutile d'en ajouter sur les paramètres qui seront alors certains. Pourtant, cette solution n'en est qu'une parmi d'autres et elle est très souvent non-optimale. \diamond

En considérant la représentation graphique du domaine θ sur la figure B.1, on constate que dans $[\theta]$, plusieurs points n'expliquent aucun triplet $\{y_k, u_{k-1}, u_{k-2}\}$ -l'espace vide dans l'enveloppe englobante- et pour un triplet $\{y_k, u_{k-1}, u_{k-2}\}$, il peut exister plusieurs solutions -un segment de la ligne l_k est incluse dans $[\theta]$ -. Prendre $[\theta]$ sous la forme d'une boîte n'est donc pas optimal, et son image dans l'espace de sortie ne produira pas l'enveloppe la plus petite possible. En déformant la boîte, il est parfois possible de trouver les domaines alternatifs pour θ_k tels que l'image de ce domaine produise une enveloppe englobante plus petite. Une approche consiste à prendre en compte le couplage potentiel à un instant k entre les paramètres du modèle par l'intermédiaire d'une matrice T .

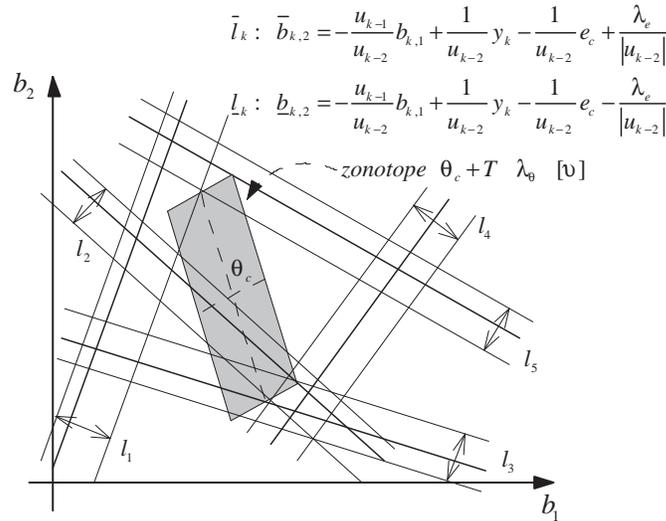


FIG. B.2 – Espace des paramètres et domaine de solution pour les paramètres couplés par une matrice de couplage.

Le nouveau domaine $S(\theta)$ obtenu s'écrit alors de manière analogique sous la forme suivante :

$$S(\theta) = \theta_c + T \times \lambda_\theta \times [v]$$

Le domaine $S(\theta)$ s'appelle un *zonothe* [Ziegler, 1998] et [Kaibel et al., 2003]. Une manière pour déterminer un *zonothe* est de le représenter par son centre θ_c , sa matrice

Annexe B. Représentation géométrique de la sémantique d'identification

de couplage T et la matrice diagonale λ_θ qui déterminent respectivement la forme et les largeurs des bandes du zonotope. Ce domaine est dessiné sur la figure B.2 pour une matrice de couplage $T_{2 \times 2}$. Un zonotope peut également être représenté par l'ensemble de ses sommets (puisque c'est un domaine convexe) ou par un ensemble des contraintes inégalités (on parle alors des bandes de contraintes). Dans ce contexte, plusieurs travaux ont été déjà effectués. Par exemple, [Adrot et al., 2004] a traité le problème d'identification des paramètres d'un modèle dynamique MIMO² intervalle en considérant le couplage entre les paramètres du modèle à l'intermédiaire d'une matrice T . Dans ce document, on supposera que T est la matrice unité et donc, le domaine de θ est une boîte.

²Multi-Input Multi-Output

Bibliographie

- Adrot, O. [2000], Diagnostic à base des Modèles Incertains Utilisant l'Analyse par Intervalles, PhD thesis, Institut National Polytechnique de Lorraine, Nancy, France.
- Adrot, O. & Flaus, J. M. [2004], 'Fault detection of uncertain dynamic systems', *Complex System Intelligence and Modern Technological Applications* .
- Adrot, O., Shahriari, K. & Flaus, J. [2004], 'Estimation of bounded model uncertainties', *11th IFAC Symposium on Automation in Mining and Metal Processing* .
- Alefeld, G. & Herzberger, J. [1983], *Introduction to Interval Computations*, Academic Press, N.Y.
- Anderson, T. W. [1994], *The Statistical Analysis of Time Series*, Wiley.
- Armengol, J. [1999], Application of Modal Interval Analysis to the simulation of the behaviour of dynamic systems with uncertain parameters, PhD thesis, Universitat de Girona.
- Armengol, J., Vehi, J., Sainz, M. A. & Herrero, P. [2003], 'Fault detection in a pilot plant using interval models and multiple sliding time windows', *5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 2003)* .
- Armengol, J., Vehi, J., Sainz, M. A. & Herrero, P. [2004], 'Application of interval models to the detection of faults in industrial processes', *World Automation Congress WAC* .
- Armengol, J., Vehi, J., Travé-Massuyès, L. & Sainz, M. A. [2001], 'Application of modal intervals to the generation of error-bounded envelopes', *Reliable Computing* **7**(2), 171–185.
- Babaali, M. & Egerstedt, M. [2005], 'Asymptotic observers for discrete-time switched linear systems', *Proceedings of 16th IFAC World Congress* .
- Balluchi, A., Benvenuti, L., Benedetto, M. D. D. & Vincentelli, A. L. S. [2001], 'A hybrid observer for the driveline dynamics', *Proceedings of the European Control Conference* .
- Balluchi, A., Benvenuti, L., Benedetto, M. D. D. & Vincentelli, A. L. S. [2002], 'Design of observers for hybrid systems', *Proceedings of Hybrid Systems : Computation and Control* pp. 76–89.
- Berleant, D. & Kuipers, B. [1992], Combined qualitative and numerical simulation with q3, in B. Faltings & P. Struss, eds, 'Recent Advances in Qualitative Physics', MIT Press.

Bibliographie

- Bliet, C., Spellucci, P., Vicente, L. N., Neumaier, A., Granvilliers, L., Monfroy, E., Benhamou, F., Huens, E., Hentenryck, P. V., Sam-Haroud, D. & Flatings, B. [2001], *Algorithms for Solving Nonlinear Constrained and Optimization Problem, COCONUT project*, <http://www.mat.univie.ac.at/~neum/glopt/coconut/>.
- Boillereaux, L., Fibrianto, H. & Flaus, J. M. [2002], ‘Switched linear modelling approach for high pressure thawing process analysis’, *International Journal of Control* **75**(16-17), 1260–1268.
- Bonarini, A. & Bontempi, G. [1994], ‘A qualitative simulation approach for fuzzy dynamical models’, *ACM Transactions on Modeling and Computer Simulation* **4**(4), 285–313.
- Boukhris, A., Giuliani, S. & Mourot, G. [2001], ‘Rainfall-runoff multi-modeling for sensor fault diagnosis’, *Control Engineering Practice* **9**(6), 659–671.
- Box, G. E. P. & Jenkins, G. M. [1970], *Time Series Analysis, Forecasting and Control*, Holand Day, San Francisco.
- Brillinger, D. R. [1981], *Time Series : Data Analysis and Theory*, Holden-Day, San Francisco.
- Calm, R., Sainz, M., Herrero, P., Vehi, J. & Armengol, J. [2006], ‘Parameter identification with quantifiers’, *5th IFAC Symposium on Robust Control Design*.
- Cellier, F. E. & Kofman, E. [2006], *Continuous System Simulation*, Springer.
- Close, C. M., Frederick, D. K. & Newell, J. C. [2001], *Modeling and Analysis of Dynamic Systems*, 3rd edn, Wiley.
- Coleman, T. & Li, Y. [1994], ‘On the convergence of reflective newton methods for large scale nonlinear minimization subject to bounds’, *Mathematical Programming* (2), 189–224.
- Coleman, T. & Li, Y. [1996], ‘An interior, trust region approach for nonlinear minimization subject to bounds’, *SIAM Journal on Optimization* **6**, 418–445.
- Combastel, C. [2003], ‘A state bounding observer based on zonotopes’, *Proc. of European Control Conference*.
- Combastel, C. [2005], ‘A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes’, *44th IEEE Conference on Decision and Control, European Control Conference. CDC-ECC*.
- Dang, T. X. T. [2000], *Vérification et Synthèse des Systèmes Hybrides*, PhD thesis, Institut National Polytechnique de Grenoble.
- Davis, M. [1985], *Stochastic Modelling and Control*, Springer.
- Domlan, E. A. [2006], *Diagnostic des Systèmes à Changement de Régime de Fonctionnement*, PhD thesis, Institut National Polytechnique de Lorraine.
- Dudley, D. G. [1983], ‘Parametric identification of transient electro-magnetic systems’, *Wave Motion* **5**, 369–384.
- Dwyer, P. S. [1951], *Linear Computation*, J. Wiley, N.Y.
- Fiedler, M., Nedoma, J., Ramik, J., Rohn, J. & Zimmermann, K. [2006], *Linear Optimization Problem with Inexact Data*, Springer-Verlag.

Bibliographie

- Figueiredo, L. H. & Stolfi, J. [1997], *Self-Validation Numerical Methods and Applications*, (Brazilian Mathematics Colloquium Monograph), IMPA, Rio de Janeiro, Brazil.
- Filev, D. [1991], ‘Fuzzy modeling of complex systems’, *International Journal of Approximate Reasoning* **5**(3), 281–290.
- Flaus, J. [1998], ‘Modeling and analysis of hybrid dynamical systems : a beginner’s guide’, *JESA* **32**(7-8), 797–830.
- Flaus, J. M. & Adrot, O. [2003], ‘Estimation d’état sûre pour procédés non-linéaires par méthodes ensemblistes - application à un procédé biotechnologique’, *Journal Européen des Systèmes Automatisés* **37**(9), 1145–1162.
- Fletcher, R. & Powell, M. [1963], ‘A rapidly convergent descent method for minimization’, *Computer Journal* **6**, 163–168.
- Gasso, K. [2000], Identification des systèmes dynamiques non-linéaires : Application Multi Modèle, PhD thesis, Centre de Recherche en Automatique de Nancy.
- Godfrey, K. R. [1983], *Comportmental Models and their Applications*, Academic Press, New York.
- Goldfarb, D. [1970], ‘A family of variable metric update derived by variational means’, *Mathematics of Computing* **24**, 23–25.
- Hadj-Sadok, M. & Gouze, J. [1998], ‘Bounds estimations for uncertain models of wastewater treatment’, *Proceedins of the 38th IEEE International Conference on Control Applications* .
- Hanebeck, U. D., Briechle, K. & Rauh, A. [2003], ‘Progressive bayes : A new framework for nonlinear state estimation’, *Proceedings of SPIE AeroSense Symposium* .
- Hannan, E. J. [1970], *Multiple Time Series*, John Wiley & Sons.
- Hansen, E. R. & Walster, G. W. [2004], *Global Optimization Using Interval Analysis*, MIT Press, Cambridge, Massachusette.
- Heath, T. L., ed. [1897], *The Works of Archimedes*, Cambridge University Press, Cambridge. Dover edition, 1953.
- Henrion, D. [2001], *Graduate Course on Polynomial Methods for Robust Control*.
- Herzberger, J., ed. [1994], *Topics in Validated Computations : Proceedinds of IMACS-GAMM International Workshop on Validation Computation*, Elsevier.
- Jacobs, R. A. & Jordan, M. I. [1991], ‘Adaptive mixtures of local experts’, *Neural Computation* **6**, 181–214.
- Jaulin, L., Kieffer, M., Didrit, O. & Walter, E. [2001a], *Applied Interval Analysis*, Springer-Verlag.
- Jaulin, L., Kieffer, M., Didrit, O. & Walter, E. [2001b], *Applied Interval Analysis*, Springer-Verlag, London.
- Johansen, T. A. & Foss, A. B. [1993], ‘Constructing narmax using armax’, *International Journal of Control* **58**(5), 1125–1153.
- Johansen, T. A. & Foss, A. B. [1997], ‘Operating regime based process modeling and identification’, *Computers & Chemical Engineering* **21**(2), 159–176.

Bibliographie

- Johansen, T. A. & Foss, B. A. [1992], ‘Nonlinear local model representation for adaptive systems’, *Singapore International Conference on Intelligent Control and Instrumentation, SICICI* **2**, 677–682.
- Johansen, T. A. & Foss, B. A. [1995], ‘Identification of non-linear system structure and parameters using regime decomposition’, *Automatica* **31**(2), 321–326.
- Kaibel, V., Klee, V. & Ziegler, G. [2003], *Convex Polytopes*, Graduate Texts in Mathematics, 2nd edn, Springer Verlag.
- Kashyap, R. L. & Rao, A. R. [1976], *Dynamic Stochastic Models from Empirical Data*, Academic Press, New York.
- Kay, H. [1996], Refining Imprecise Models and Their Behaviors, Dissertation, The University of Texas at Austin.
- Kearfott, R. B. [1996], *Rigorous Global Search : Continuous Problems*, Springer.
- Kletting, M., Rauh, A., Aschemann, H. & Hofer, E. P., eds [2005], *Consistency Techniques for Simulation of Wastewater Treatment Process with Uncertainties*, DVD-Proceeding of IFAC World Congress, Prague, Czech Republic.
- Kolev, L. [1993], *Interval Methods for Circuit Analysis*, World Scientific.
- Kraemer, W. & Gudenberg, J. W. V., eds [2001], *Scientific Computing, Validated Numerics, Interval Methods*, Kluwer Academic Publisher, Boston/Dordrecht/London.
- Kreinovich, V., Lakeyev, A. V., Rohn, J. & Kahl, P. T. [1998], *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Springer.
- Lemmon, M. D., He, K. X. & Markovskiy, I. [1999], ‘Supervisory hybrid systems’, *IEEE Control System Magazine* **19**(4), 42–55.
- Ling, C. & Edgar, T. F. [1997], ‘Real-time control of a water-gas shift reactor by a model-based fuzzy gain scheduling technique’, *Journal of Process Control* **7**(4), 239–253.
- Ljung, L. [1999], *System Identification*, Prentice Hall, New Jersey.
- Ljung, L. & Soderstrom, T. [1983], *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, Mass.
- Malinvaud, E. [1980], *Statistical Methods of Econometrics*, North-Holland, Amsterdam.
- Markel, J. D. & Gray, A. H. [1976], *Linear Prediction of Speech*, Springer-Verlag, New York.
- Milanese, M., Norton, J., Piet-Lahanier, H. & Walter, E. [1996], *Bounding Approaches to System Identification*, Plenum Press, New York, London.
- Moore, R. [1962], Interval Arithmetic and Automatic Error Analysis in Digital Computing, Phd thesis, Applied Mathematics and Statistics Laboratories, Report 25, Stanford University.
- Moore, R. [1966], *Interval Analysis*, Printice Hall, Englewood Cliffs, NJ.
- Murray-Smith, R. & Johansen, T. A. [1997a], *Multiple Model Approaches to Modelling and Control*, CRC.
- Murray-Smith, R. & Johansen, T., eds [1997b], *Multiple Model Approaches to Modelling and Control*, Taylor & Francis.
- Neumaier, A. [1990], *Interval Methods for Systems of Equations*, Cambridge University Press.

Bibliographie

- Neumaier, A. [2001], *Introduction to Numerical Analysis*, Cambridge Univ. Press, Cambridge.
- Neumaier, A. [2002], ‘Grand challenges and scientific standards in interval analysis’, *Reliable Computing* **8**(4), 313–320.
- Nicholson, H. [1981], *Modelling of Dynamical Systems (IEE Control Engineering Series)*, Inspec/Iee.
- Norgaard, M., Ravn, O., Poulsen, N. K. & Hansen, L. K. [2006], *Neural Networks for Modeling and Control of Dynamic Systems*, Springer.
- Petras, I., Chen, Y., Vinagre, B. B. & Podlubny, I. [2003], Stability of linear time invariant systems with interval fractional orders and interval coefficients. Preprint for IEEE CDC 2003 Conference.
- Petras, I., Chen, Y. & Vinagre, B. M. [2003], Robust stability test for interval fractional order linear systems, in ‘Sixty open problems in the mathematics of systems and control’, Princeton University Press.
- Ploix, S. [1998], Diagnostic des Systèmes Incertains : l’Approche Bornante, PhD thesis, l’Institut National Polytechnique de Lorraine, France.
- Ploix, S., Adrot, O. & Ragot, J. [1999], ‘Parameter uncertainty computation in static linear models’, *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on* **2**, 1916–1921.
- Popper, K. R. [1934], *The Logic of Scientific Discovery*, Basic Books.
- Ragot, J., Maquin, D. & Domlan, E. A. [2003], ‘Switching time estimation of piecewise linear systems. application to diagnosis’, *Proceedings of the 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* pp. 669–704.
- Raissi, T., Ramdani, N. & Candau, Y. [2006], ‘Robust nonlinear continuous-time state estimation using interval Taylor models’, *5th IFAC Symposium on Robust Control Design* .
- Ratschek, H. & Rokne, J. [1988], *New Computer Methods for Global Optimization*, Horwood Publishing Limited.
- Ratz, D. [1998], *Automatic Slope Computation and its Application in Nonsmooth Global Optimization*, Shaker-Verlag, Aachen.
- Rauh, A. & Hofer, E. P. [2005], ‘Interval arithmetic optimization techniques for uncertain discrete-time systems’, *13th International Workshop on Dynamic and Control* .
- Rauh, A., Kletting, M., Aschemann, H. & Hofer, E. P. [2005], ‘Robust controller design for bounded state and control variables and uncertain parameters using interval methods’, *International Conference on Control and Automation, ICCA* .
- Shahriari, K., Adrot, O. & Flaus, J. [2004], ‘Analyse de sûreté de procédés multi modes par des méthodes à intervalles’, *Conférence Internationale Francophone d’Automatique CIFA* .
- Shahriari, K., Adrot, O. & Flaus, J. [2006], ‘Multi mode modeling with linear dynamic models using set-membership tools’, *The 11th Information Processing and Management of Uncertainty in Knowledge-Based Systems International Conference* .
- Soderstrom, T. & Stoica, P. [1987], *System Identification*, Printice Hall, Englewood Cliffs, N.J.

Bibliographie

- Spriet, J. A. & Vansteenkiste, G. C. [1982], *Computer Aided Modelling and Simulation*, Academic Press, New York.
- Sugeno, M. & Kang, G. T. [1988], 'Structure identification of fuzzy model', *Fuzzy Sets and Systems* **28**(1), 15–33.
- Takagi, T. & Sugeno, M. [1985], 'Fuzzy identification of systems and its applications to modeling and control', *IEEE Transactions on Systems, Man and Cybernetics* **15**, 116–132.
- Valdes-Gonzalez, H.-M. [2002], Contribution à l'Estimation d'Etat à Horizon Glissant par Méthodes Ensemblistes, PhD thesis, Laboratoire d'Automatique de Grenoble, Université Joseph Fourier.
- Valdes-Gonzalez, H.-M., Flaus, J.-M. & Acuna, G. [2003], 'Moving horizon state estimation with global convergence using interval techniques : Application to biological processes', *Journal of Process Control* **13**(4), 325–336.
- Venkat, A. N., Vijaysai, P. & Gudi, R. D. [2003], 'Identification of complex nonlinear processes based on fuzzy decomposition of the steady state space', *Journal of Process Control* **13**(6), 473–488.
- Walter, E. & Pronzato, L. [1997], *Identification of Parametric Models : From Experimental Data*, Springer-Verlag Berlin and Heidelberg, GmbH & Co. K.
- Warmus, M. [1956], 'Calculus of approximations', *Bulletin de l'Academie Polonaise de Sciences* **4**(5), 253–257.
- Warmus, M. [1961], 'Approximations and inequalities in the calculus of approximations. classification of approximate numbers', *Bull. Acad. Polon. Sci. Ser. Sci. Math. Astronom. Phys.* **9**, 241–245.
- Wellstead, P. E. [1979], *Introduction to Physical System Modelling*, Academic Press, New York.
- Yager, R. R. & Filev, D. P. [1993], 'Unified structure and parameter identification of fuzzy models', *IEEE Transactions on Systems, Man and Cybernetics* **23**(4), 1198–1205.
- Young, P. C. [1984], *Recursive Estimation and Time-Series Analysis*, Springer-Verlag, New York.
- Young, R. C. [1931], 'The algebra of multi-valued quantities', *Mathematische Annalen* **104**, 260–290.
- Young, W. H. [1908], 'Sull due funzioni a piu valori costituite dai limiti d'una funzione di variable reale a destra ed a sinistra di ciascun punto', *Rendiconti Accademia di Lincei, Classes di Scienza Fifiche* **17**(5), 582–587.
- Ziegler, G. [1998], *Lectures on Polytopes*, Graduate Texts in Mathematics, Springer Verlag, New York, USA.

Résumé

Lorsqu'un système physique est en fonctionnement, un des objectifs d'une analyse de sûreté est d'être sûr que les variables caractérisant l'état du système suivent une certaine trajectoire ou restent dans une zone donnée compte tenu de toutes les perturbations et les incertitudes sur les paramètres et les mesures du système. Une solution originale pour ce problème est proposée dans ce document à l'aide d'une approche ensembliste qui consiste à calculer l'ensemble des trajectoires possibles générées en présence des perturbations dues à l'environnement du système et à la méconnaissance des modèles du système se traduisant par des incertitudes sur la valeur de certains de ses paramètres. Dans le cas de notre problème, les incertitudes peuvent porter sur les valeurs des paramètres, l'état initial ou les mesures du système. Pour garantir le résultat de l'analyse de sûreté, toutes les trajectoires doivent être envisagées, pour toutes les perturbations et les incertitudes considérées. Pour représenter le comportement d'un système complexe (un système multi mode ou un système non-linéaire), au lieu d'utiliser un modèle avec une structure complexe, une approche standard est de diviser le régime opératoire du système en zones locales est de représenter chaque zone par un modèle linéaire avec une structure simple. On essaie dans ce travail de thèse d'apporter une solution au problème d'identification de systèmes incertains dans un premier temps par un modèle mono-mode et ensuite par un modèle multi-mode pour un objectif d'analyse de sûreté de système. Ce document est constitué de cinq chapitres. Dans le premier chapitre, la problématique du travail est détaillée. Dans le deuxième chapitre, un état de l'art sur la problématique étudiée est donné. On traite ensuite du problème d'identification et de simulation d'un modèle ensembliste (intervalle) dans le chapitre trois. Pour améliorer la qualité de la modélisation et augmenter la précision de la prédiction, une approche méthodologique est proposée dans le chapitre quatre pour identifier et pour simuler ce modèle intervalle. Dans le dernier chapitre, une conclusion est apportée sur l'ensemble de ce travail de thèse.

Mots Clés : Identification des Systèmes Dynamiques Incertains, Simulation des Systèmes Dynamiques Incertains, Analyse par Intervalle, Analyse de Sûreté, Modèle Multi-Mode

Abstract

While a system is in operation, one of the aims of the process safety analysis is to guaranty that the physical variables of the system follow a certain trajectory and stay in a given zone considering all the perturbations and the uncertainty on the real values of the parameters of the model and the measurements. An original solution for this problem which has been proposed in this document is to consider the environmental perturbations and the ignorance of the exact values of the parameters of the model as uncertainty on the parameters of the model, characterize the support of the parameters of the model by intervals and then to use the interval arithmetic to calculate all the possible trajectories of the system. In the other hand, to represent the behaviour of a complex system (a multi-mode system or a non-linear system), instead of using a complex single-mode model, a classical approach is to split the operating regime of the system into local zones and to represent every zone by a linear model which has a simple structure. In this work, we try to propose a solution to the identification problem of uncertain dynamic systems in the first time by a single-mode model and then by a multi-mode model in the process safety analysis context. This document consists of five chapters. In the first chapter, the objective of this work and the problem to be solved is explained. In the second chapter, the study of the literature of the problem is given. We then deal with the problem of the identification and the simulation of interval model in chapter three. To improve modelling quality and to increase the prediction precision, a methodological approach is proposed in chapter four to identify and to simulate multi-mode interval models. In the last chapter, the conclusion of this work is given.

Keyword : Uncertained Dynamic System Identification, Uncertained Dynamic System Simulation, Interval Analysis, Process Safety Analysis, Multi-Mode Model